☐ int saisie\_utilisateur();

# Éléments d'informatique $\underline{-\operatorname{contrôle}}$ continue

D /	NT.
Prénom:	Nom:
N° INE ·	
1, 11,12,	

a continue   continu			
wins awaz déjà fait un for duns le même price gramme principal   wins n'avez pas dédaré de fonction   finctime principal   wins n'avez pas dédaré de fonction   finctime de la variable de boucle n'est pas 1   wins ne cumaissez pas le nombre d'itérations de la boucle à l'avance   un document de carrier propriet à la compilation :   erreur : conflècting types for 'max' , que doit- on cherche dans le programme?   une directive préprocesseur #include manquante   une fonction apélée wunt sa déclaration   une fonction apélée wunt sa déclaration   une fonction apélée wunt sa déclaration   un document de référence du système   un document de référence du système   un deciment sur l'oritie sur l'oritie   x = racine(x);	Barème : 1 points par réponse juste (unique) ; $-0, 5$ points par réponse fausse. Durée : 20 minutes.		
wins new past déclaré de fonction     finitude (statio.h.b.)     sinclude (statio.h.b.)     main   min   m	1. Vous utilisez une boucle while quand:	☐ #include <studio.h></studio.h>	□ begin
wous n'awez pas déclaré de fonction   fincrément de la variable de bouche n'est pas 1   winclude statio.h>   finclude statio.h>   winclude statio.h>   finclude statio.h   finclude statio in tradiut en instructions principal.   finclude station en instruction deliciter protégé en tradiut en instruction dure definice to initialisée, il est incirce d'éverire:   sa racine(x);   lissou mix (on linux), la commande cd permet de station en returne station en ferier en descript en ferieure du source	v	☐ #appart <stdlib.h></stdlib.h>	□ include
Timerément de la variable de boucle n'est pas     vous ne counsisses pas le vous ne counsisses pas le vous ne combre d'iterations de la boucle à l'avance  2. Si cette erreur apparaît à la compliation :   un document qui doit être protégé     un document de référence du système     un fichier source cst :   desaccord entre la déclaration et la définition d'une fonction et la définition d'une fonction d'une fonction et la définition d'une fonction d'une fonction della finition d'une fonction d'une fonction della finition d'une fonction d'une fon		☐ #include <studlib.h></studlib.h>	$\square$ main
Ose racine est un contains et al abouche à l'avance   Os un decument qui doit être protégé   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur = une directive préprocesseur = #include manquante   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur   Un document qui doit être protégé   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier texte qui sera traduit en instructions processeur   Un fichier que l'oxidit sur l'ordinateur   Un fichier que	- I	☐ #include <stdio.h></stdio.h>	□ init
un document illisible pour les humains   un document un control prenare et removant la racine carrée de cet réel, et que x est une variable réelle définite et mitialisée, il est incorrect dévirre:   un document qui doit être protégé   un fichier texte qui sera traduit en instructions processeur #include manquante   une fonction appelée avant sa déclaration   un document de référence du système   un décancer mais non définite   un désaccord entre la déclaration et la définition d'une fonction déclarée mais non définite   un foitier que l'out doit citer dans les documents produits sur l'ordinateur   7. Soit la fonction g définie par : int g(int a \ n');   f( 1 > 0)   frintf("Mineur\n");   f	- I	6. Un fichier source est:	
2. Si cette erreur apparaît à la compilation : erreur: conflicting types for "max" , que doit on chercher dans le programme?   une directive próprocesseur #include manquante   une fonction appelée avant sa déclaration   une fonction déclarée mais non définie   une dissecord entre la déclaration et la définition   d'une fonction   une fonction déclarée mais non définie   une dissecord entre la déclaration et la définition   d'une fonction   Le code suivant : int age = 20; if (age < 18) {     printf("Majeur\n"); } else {     printf("Majeur\n"); } Alors l'expression g(0) prendra la valeur :   mineur   majeur   rien   4. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur (char co);   class equel vous avez déclaré ces fonction   dans lequel vous avez déclaré ces fonctions oat sie_utilisateur (char co);   main   une doument qui doit être protégé   une directive profeceseur #include manquante     une directive profece du veystème     une document qui doit être protégé   une directive profece du verie en instructions processeur     x = racine(2/3);   x = racine(x x ) - racine(x);   x = r			
erreur: conflicting types for "sax", que doit- on chercher dans le programme?    une directive préprocesseur #include manquante   une fonction appelée avant sa déclaration   une fonction appelée avant sa déclaration et la définition   d'une fonction déclarée mais non définie   un désaccord entre la déclaration et la définition   d'une fonction déclarée mais non définie   un désaccord entre la déclaration et la définition   d'une fonction declarée mais non définie   un fichier que l'ont doit citer dans les documents   produits sur l'ordinateur   Soit la fonction g définie par:  int gge = 20;  if (age < 18) {     printf("Mineur\n"); } else {     printf("Majeur\n"); } }  Alors l'expression g(0) prendra la valeur:   demande à l'utilisateur d'entrer un entier au clavier et reuvoie cet entier on derit:   void saisie_utilisateur (char c);   un ordre quelconque   un fichier texte qui sera traduit en instructions processeur   un document de référence du système   un fichier que l'ont doit citer dans les documents produits sur l'ordinateur   x = racine(2/3);   x = racine(x)* x - 1 = racine(x)			· · ·
me directive préprocesseur #include manquante   me directive préprocesseur #include manquante   me fonction appelée avant sa déclaration   mu desaccord entre la déclaration et la définition d'une fonction déclarée mais non définie   mu désaccord entre la déclaration et la définition d'une fonction et la définition d'une fonction age = 20;  if (age < 18) {     printf("Mineur\n"); } else {     printf("Majeur\n"); } Alors l'expression g(0) prendra la valeur :       Mineur		• • •	d'écrire :
une fonction appelée avant sa déclaration   une fonction déclarée mais non définie   un désaccord entre la déclaration et la définition d'une fonction g définie par :  int g(int a)  { printf("Mineur\n"); } else { printf("Majeur\n"); } Alors l'expression g(0) prendra la valeur :    Mineur			$\square$ x = racine(2/3);
une fonction déclarée mais non définie   un désaccord entre la déclaration et la définition d'une fonction g définie par :  int age = 20; if (age < 18) {     printf("Mineur\n"); } else {     printf("Majeur\n"); }  affichera :   0     mineur     Majeur     Majeur     Majeur     Majeur     Majeur     Majeur     Majeur     Majeur     demande à l'utilisateur d'entrer une entier au clavier et renvoie cet entier on écrit :   dans lequel vous avez déclaré ces fonction est l'ordre :   dans lequel cos fonctions sont appelées dans le main   un ordre quelconque   printf("Valeur den 7 %%\n", n);       uni ordre quelconque   un ordre quelconque   printf("valeur den 7 %%\n", n);       printf("Valeur den 7 %%\n", n);	I	$\hfill \square$ un document de référence du système	$\square$ x = racine(x * x) - racine(x);
d'une fonction  3. Le code suivant :     int age = 20;     if (age < 18)     {         printf("Mineur\n");     }     else     {             printf("Majeur\n");     } }  Alors l'expression g(0) prendra la valeur :		$\hfill\Box$ un fichier que l'ont doit citer dans les documents	$\square$ x = racine(racine(x)*racine(x));
3. Le code suivant:  int age = 20;  if (age < 18)  {  printf("Mineur\n"); } else {  printf("Majeur\n"); }  Alors l'expression g(0) prendra la valeur:    Mineur	$\square$ un désaccord entre la déclaration et la définition	produits sur l'ordinateur	$\square x - 1 = racine(x);$
int age = 20; if (age < 18) {     printf("Mineur\n"); } else {     printf("Majeur\n"); } Alors l'expression g(0) prendra la valeur:       Mineur       Majeur       Majeur       Majeur       rein  4. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur (char c);       Maiseur (dans lequel vous avez déclaré ces fonction est l'ordre :       dans lequel vous avez déclaré ces fonction soisie_utilisateur (char c);       Saisie_utilisateur (sanf (%d));       void saisie_utilisateur (int n);  int g(int a) {     printf("a = \n", %d);     if (1 > 0)     {         return 5;     }     return 7; } Alors l'expression g(0) prendra la valeur:       0       7       5       S. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :       dans lequel vous avez déclaré ces fonction       dans lequel ces fonctions sont appelées dans le main       void saisie_utilisateur(sanf (%d));       un ordre quelconque		7. Soit la fonction g définie par :	11 Sous unix (ou linux) la commande cd permet de :
changer de répertoire courant   jouer de la musique   détruire un fichier   jouer de la musique   détruire un fichier   detruire un programme arrêté avec la commande ab   return 7;   }   Alors l'expression g(0) prendra la valeur :   demande à l'utilisateur deur demande à l'utilisateur deur et renvoie cet entier on écrit :   dans lequel vous avez déclaré ces fonction   dans lequel ces fonctions sont appelées dans le main   un débogueur   scanf ("d", kn);   printf("Valeur de n ? %g\n", n);		int g(int a)	, , , , , , , , , , , , , , , , , , , ,
frit(1 > 0)	•	{	
printf("Mineur\n"); } else {     return 5; } printf("Majeur\n"); }  Alors l'expression g(0) prendra la valeur :    Mineur	-	=	<u> </u>
return 5; } return 7; } Alors l'expression g(0) prendra la valeur :  affichera :    Mineur     Majeur     Majeur     rien  4. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :    void saisie_utilisateur(scanf(%d));   void saisie_utilisateur(int n);    void saisie_utilisateur(int n);    void saisie_utilisateur(int n);    return 5;   return 7;   Alors l'expression g(0) prendra la valeur :   0		{	_
return 7;	•	return 5;	
printf("Majeur\n"); } Alors l'expression g(0) prendra la valeur :  afflichera :    Mineur   Majeur	else f	}	
Alors l'expression g(0) prendra la valeur :  affichera :    Mineur     Mineur     Majeur     Majeur     rien    4. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de x <sup>n</sup> on écrit :    void exposant(double x^n);     exposant(double x, int n, int r);     dans lequel vous devez maintenant définir ces fonctions est l'ordre :   dans lequel vous avez déclaré ces fonction renvoie cet entier on écrit :   void saisie_utilisateur (char c);     saisie_utilisateur (scanf (%d));     void saisie_utilisateur (int n);     un ordre quelconque      Alors l'expression g(0) prendra la valeur :   0	<pre>printf("Majeur\n");</pre>	return 7;	
affichera:    Mineur	- "	J	
affichera:    Mineur			
<ul> <li>Mineur Majeur  Majeur  Majeur  Majeur  Indicated the problem of the pro</li></ul>			
Majeur  ☐ Majeur ☐ rien  8. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :  ☐ dans lequel vous avez déclaré ces fonction est l'ordre :  ☐ dans lequel vous avez déclaré ces fonction cest l'ordre :  ☐ dans lequel vous avez déclaré ces fonction cest l'ordre :  ☐ dans lequel ces fonctions sont appelées dans le main  ☐ un débogueur ☐ un débogueur ☐ un débogueur ☐ scanf("%d", &n); ☐ un ordre quelconque ☐ printf("Valeur de n ? %g\n", n); ☐ printf("Valeur de n ? %g\n", n);			-
<ul> <li>Majeur</li></ul>			-
□ rien  4. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit : □ void saisie_utilisateur(char c); □ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(int n); □ un ordre quelconque    tions utinsées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ dans lequel vous avez déclaré ces fonction □ dans lequel vous avez déclaré ces fonction □ un débogueur □ scanf("%d", &n); □ un ordre quelconque □ printf("Valeur de n ? %g\n", n);	-		-
4. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :  □ void saisie_utilisateur(char c); □ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(int n); □ un ordre quelconque  13. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt : □ un débogueur □ un débogueur □ scanf("%d", &n); □ un ordre quelconque □ printf("Valeur de n ? %g\n", n);	- I		-
renvoie cet entier on écrit :  □ void saisie_utilisateur(char c); □ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(int n); □ un ordre quelconque □ un débogueur □ scanf("%d", &n); □ printf("Valeur de n ? %g\n", n);	•	est l'ordre :	
□ void saisie_utilisateur(char c); □ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(int n); □ un ordre quelconque □ printf("Valeur de n ? %g\n", n); □ void saisie_utilisateur(int n);		-	□ un débogueur
□ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(int n); □ un ordre quelconque □ printf("Valeur de n ? %g\n", n);			☐ scanf("%d", &n);
U void saisie_utilisateur(int n);			
	<pre>□ void saisie_utilisateur(int n);</pre> □ int saisie utilisateur():	□ alphabétique	☐ printf("Valeur de n ? %d\n", n);

### Éléments d'informatique – contrôle continue

Prénom : Nom : No INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
   for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 2; j = j + 1)
            printf("%d ", i);
        }
    }
   printf("\n");
  qu'est ce qui sera affiché?
    \Box 1 2 3 1 2
    \Box 0 1 0 1 0 1
    \Box 0 0 1 1 2 2
    \Box 0 1 2 0 1 2
2. Le code suivant :
    int i;
    for (i = 0; i < 5; i = i + 1)
        printf("%d ", i);
   }
   printf("\n");
  affichera:
    \Box \ 4\ 3\ 2\ 1\ 0
    \square 0 1 2 3
    \Box 01234
    \square 4 3 2 1
3. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
  Alors l'affichage sera:
```

```
□ ccccc
     \square A
    ☐ ABCDEF
    \Box i
4. L'ordonnancement par tourniquet permet :
    □ d'afficher des ronds colorés à l'écran
    \square de doubler la mémoire disponible
     \square de ne pas perdre de temps avec la commutation
       de contexte
    ☐ d'entretenir l'illusion que les processus tournent
       en parallèle
5. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
  Undefined symbols : "_prinft" ou
  référence indéfinie vers « prinft »
    □ l'analyse sémantique
    ☐ l'analyse des entrées clavier
    □ l'édition de liens
    ☐ l'analyse harmonique
6. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
  {
     printf("coucou\n");
     \square il ne compile pas
    ☐ il risque d'afficher bonjour à la place de coucou
    □ il n'affiche rien
    □ il comporte une boucle infinie
7. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle(double n);
    □ struct int factorielle(int n);
    ☐ int factorielle(int x);
```

☐ int factorielle():

```
8. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
      □ un point-virgule en trop
      ☐ un point-virgule manquant
      \square une accolade en trop
      ☐ une accolade manquante
 9. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
      return 7;
   Alors l'expression g(0) prendra la valeur :
      \Box 0
      \square 7
      \Box 5
10. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
            int x = 5;
 11
 12
            int y = 3;
 13
 14
            x = y;
 15
 16
 17
      \square la variable x vaut 3
      \square la variable y vaut 5
      ☐ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 3
```

11. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :	15. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :	19. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
<pre>□ void afficher_menu();</pre> □ int afficher_menu(int char);	□ 7 	☐ int char;
☐ int afficher_menu();	□ 111 □ 2	□ char 'c';
☐ double afficher_menu();	□ 8 	
☐ char afficher_menu(printf("menu"));		$\Box$ char c;
12. Lorsqu'un programme utilise printf ou scanf il faut	16. Pour l'extrait de programme suivant :	☐ char "c";
qu'il contienne l'instruction préprocesseur :	int somme = 0;	
☐ #include <studlib.h></studlib.h>	for (i = 0; i < 5; i = i + 1)	20. Soit le programme principal suivant :
☐ #include <stdio.h></stdio.h>	<pre>somme = somme + i;</pre>	<pre>int main()</pre>
☐ #include <studio.h></studio.h>	i = i + 1; /* attention ! */	int main()
☐ #appart <stdlib.h></stdlib.h>	}	int a = 3;
13. Si factorielle est une fonction prenant en entrée un	<pre>printf("somme = %d",somme);</pre>	int b = 5;
entier et renvoyant un entier, il est correct d'écrire :  □ n = factorielle();	La valeur de somme affichée est :	printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
int factorielle(int 2);	□ 10	return EXIT_SUCCESS;
□ n = factorielle(p, q);	□ 6	J
□ printf("%d", factorielle(n));	□ 0	appelant la fonction f ainsi définie :
14. Soit la fonction f définie par :	□ 15	
int f(int a)	17. Le type des réels en C est :	<pre>int f(int a, int b) {</pre>
{	☐ double	a = a + b;
printf("a = \n", %d);	□ int	return a;
if (a > 0)	□ char	}
return 3;	□ real	L'affichage dans le main est le suivant :
}	18. Un enregistrement permet de grouper plusieurs valeurs	L'amenage dans le main est le survant.
return 4;	dans :	$\Box$ f(a,b)=8, a=3, b=5
}	□ ses champs	$\Box$ f(a,b)=13, a=8, b=5
Alors l'expression $f(0)$ prendra la valeur : $\square$ 3	□ ses chants	☐ f(3,5)=8, a=3, b=5
	□ ses cases	
	□ ses blocs	$\Box$ f(a,b)=8, a=8, b=5
	•	

### Éléments d'informatique – contrôle continue

Prénom : Nom :		
	Prénom:	Nom:
$N^{\circ}$ INE :	N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si a et b sont deux variables de type :
  struct toto_s
  {
     int n;
     double x;
  };
  Alors pour tester l'égalité de a et de b on utilise la
  condition:
    \square a\{n, x\} == b\{n, x\}
    \square (a.n == b.n) && (a.x == b.x)
    \square a == b
    \Box a = b
2. Si cette erreur apparaît à la compilation :
  erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
     □ une fonction déclarée mais non définie
    □ une directive préprocesseur #include manquante
    □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
3. Quel est le problème d'un programme comportant les
  lignes suivantes?
  while (1)
     printf("coucou\n");
    \square il risque d'afficher bonjour à la place de coucou
     \square il ne compile pas
     \square il comporte une boucle infinie
     □ il n'affiche rien
4. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
     ☐ #appart <stdlib.h>
```

☐ #include <studlib.h>

☐ #include <stdio.h>

☐ #include <studio.h>

```
5. Le code suivant :
   int age = 15;
   if (age < 18)
   {
        printf("Mineur\n");
   else
   {
        printf("Majeur\n");
  affichera:
    □ Mineur
       Majeur
    \square rien
    □ Majeur
    □ Mineur
6. Pour déclarer une fonction factorielle qui prend en
  argument un entier et renvoie sa factorielle on écrit :
    ☐ int factorielle(double n):
    ☐ int factorielle();
    □ struct int factorielle(int n):
    \square int factorielle(int x);
7. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
    □ void afficher_menu();
    ☐ int afficher_menu();
    □ double afficher_menu();
    ☐ char afficher_menu(printf("menu"));
    ☐ int afficher_menu(int char);
8. Si cette erreur apparaît à la compilation :
  Undefined symbols : "_prinft" ou
  référence indéfinie vers « prinft » que doit-
  on chercher dans le programme?
    □ une variable non déclarée
    □ un caractère interdit en C
    \square une faute de frappe dans un appel de fonction
    ☐ une directive préprocesseur #include manquante
```

```
9. Soit le programme principal suivant :
   int main()
    int a = 3:
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
10. Après exécution jusqu'à la ligne 14 du programme C :
      int main() {
 11
           int x = 5;
 12
 13
           printf(" x = %d\n", 2);
 14
 15
 16
     \square le terminal affiche x = 5
     □ le terminal affiche "Faux"
     □ le terminal affiche 5
     \square le terminal affiche x = 2
11. Vous utilisez une boucle while quand :
     □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ l'incrément de la variable de boucle n'est pas 1
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
```

```
12. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
    on chercher dans le programme?
      \square une accolade en trop
      \square un point-virgule en trop
      ☐ une accolade manquante
      □ un point-virgule manquant
13. Le code suivant :
     int i;
     for (i = 4; i \ge 0; i = i - 1)
         printf("%d ", i);
     printf("\n");
    affichera:
      \Box 01234
      \Box \ 4\ 3\ 2\ 1\ 0
      \square 4 3 2 1
      \Box 1 2 3 4
14. Sur unix (ou linux), la commande mkdir permet de :
      \square créer un fichier texte
      □ changer de répertoire courant
      □ créer un répertoire
      \square ouvrir un fichier texte
15. Avant de faire appel à une fonction il est nécessaire
    de:
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      □ l'avoir déclarée
      □ l'avoir déclarée et définie
      ☐ l'avoir définie
```

```
16. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
     printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box j = 0
     \Box j = 4
      \Box j = %d
      \Box i = 5
17. Sous unix (ou linux), la commande cd permet de :
      □ changer de répertoire courant
      □ détruire un fichier
     □ ouvir un bureau partagé (common desktop)
      \square jouer de la musique
      □ récupérer un programme arrêté avec la commande
18. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
```

return 3;

```
}
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \Box 0
      \Box 4
      \square 3
19. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1:
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \Box 1
      \Box 5
      \Box 4
      \Box 0
20. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
     \square int pgcd(int x, y);
     \square int pgcd(int y, int x);
     \square void pgcd(int x, int y);
      \square int pgcd(int x, int x);
```

### Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si le code : struct toto s

```
int n;
  double x:
};
précède la fonction main(), alors on peut écrire en
début de main() :
 \square int toto.n = 3;
  \square toto_s struct z = {3, 0.5};
  □ struct toto_s toto;
```

 $\square$  int struct toto\_s = {3, -1e10}; 2. Les lignes

```
int i;
int x=0;
for(i=0,i<5,i=i+1)
{
  x=x+1;
```

 $\square$  toto\_s n, x;

□ comportent une erreur qui sera détectée au cours de l'édition de lien

 $\square$  ne comportent aucune erreur

□ comportent une erreur qui ne sera pas détectée

□ comportent une erreur qui sera détectée au cours de l'analyse syntaxique

3. Le code suivant :

```
int somme = 0;
 int i;
 for (i = 1; i < 4; i = i + 1)
   somme = somme + i;
 printf("%d", somme);
affichera:
```

```
\square 42
     \Box 6
    \square 0
    \Box 1
4. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera:
    \Box C
    \square B
    \square A
     ПЪ
   warning: implicit declaration of function 'max'
  , que doit-on chercher dans le programme?
```

5. Si cet avertissement apparaît à la compilation : □ une directive préprocesseur **#include** manquante

□ une fonction déclarée mais non définie

 $\square$  un désaccord entre la déclaration et la définition d'une fonction

☐ une fonction appelée avant sa déclaration

6. Soit le programme principal suivant :

```
int main()
int a = 3;
int b = 5;
printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
return EXIT_SUCCESS;
appelant la fonction f ainsi définie :
int f(int a, int b)
  a = a + b;
  return a;
```

L'affichage dans le main est le suivant :

f(a,b)=8,	a=3,	b=5
f(a,b)=8,	a=8,	b=5
f(a,b)=13,	a=8	, b=5
f(3.5)=8.	a=3.	b=5

7. Avant de faire appel à une fonction il est nécessaire de:

□ l'avoir déclarée et définie

☐ l'avoir définie

☐ l'avoir déclarée

□ avoir défini une constante symbolique de la taille de cette fonction

8. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 2; i = i + 1)
    for (j = 0; j < 3; j = j + 1)
        printf("%d ", i);
   }
}
```

qu'est ce qui sera affiché?

 $\Box$  0 1 2 0 1 2  $\Box$  0 0 0 1 1 1  $\Box$  0 1 0 1 0 1 0 1

 $\Box$  1 2 1 2 3

9. L'écriture 111 en binaire correspond au nombre naturel:

 $\square$  3  $\square$  111

 $\square$  7

 $\square$  8

10. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?

□ un point-virgule en trop

 $\square$  une accolade en trop

□ un point-virgule manquant

 $\square$  une accolade manquante

11. Si pgcd est une fonction prenant en entrée deux entiers	Alors l'expression g(0) prendra la valeur :	$\Box$ le terminal affiche x = 2
et renvoyant un entier, il est correct d'écrire :	$\square$ 5	$\Box$ le terminal affiche x = 5
$\square n = pgcd(n, 3);$	□ 7	
$\square$ n = pgcd(int p, int q);	$\square$ 0	□ le terminal affiche "Faux"
$\square$ int n = pgcd();	15. Après exécution jusqu'à la ligne 15 du programme C:	$\square$ le terminal affiche $5$
☐ int pgcd(2);	10 int main() {	
12. Une de ces manière de composer les blocs de pro-	11 int x = 5;	18. Un enregistrement permet de grouper plusieurs valeurs
grammes ne fait pas partie des opérations de la pro-	12 int y;	dans:
grammation structurée :	13	$\square$ ses chants
□ répéter un bloc tant qu'une condition est vérifée	14   y = x;	□ ses blocs
□ sélectionner entre deux blocs à l'aide d'une condi-	15 16	
tion $\Box$ retourner un bloc	16 17 }	$\square$ ses champs
		$\square$ ses cases
☐ mettre les blocs en séquence les uns à la suite des autres	☐ la variable y vaut 5	40.7
13. Pour afficher à l'aide de printf("%d\n",tab[i]);	☐ la variable x vaut 0	19. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
le contenu d'un tableau de 5 entiers initialisé au	☐ la variable x vaut 5 et la variable y vaut 0	d une variable de type caractere en langage C:
préalable, on utilise plutôt :	☐ le programme affiche "Faux"	□ char c;
☐ for(i=1;i<5;i=i+1)	16. Si $n$ est une variable entière pour demander sa valeur	☐ char 'c';
☐ for(i=0;i<5;i=i+1)	à l'utilisateur, on utilise plutôt :	□ char "c";
☐ for(i=1;i<=5;i=i+1)	□ un débogueur	,
☐ for(i=0;i<=5;i=i+1)	☐ printf("Valeur de n ? %g\n", n);	$\square$ int char;
14. Soit la fonction g définie par :	☐ scanf("%d", &n);	20. Pour déclarer une procédure afficher_menu sans ar-
int g(int a)	☐ printf("Valeur de n ? %d\n", n);	gument et qui ne renvoie rien on utilise :
{	17. Après exécution jusqu'à la ligne 14 du programme C :	•
<pre>printf("a = \n", %d);</pre>	10 int main() {	☐ int afficher_menu(int char);
if (1 > 0)	11 int x = 5;	☐ int afficher_menu();
{	12	☐ char afficher_menu(printf("menu"));
return 5;	13 printf(" x = %d\n", 2);	<u>-</u>
return 7;	14 15	☐ double afficher_menu();
}	15 16 }	□ void afficher_menu();
·	' '	

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

 ${\it 1. \ Quel \ est \ le \ problème \ d'un \ programme \ comportant \ les \ lignes \ suivantes ?}$ 

```
while (1)
{
   printf("coucou\n");
}

   il ne compile pas
```

- $\square$  il comporte une boucle infinie
- □ il n'affiche rien
- $\Box\,$ il risque d'afficher bonjour à la place de coucou
- 2. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
  - $\square$  un ordre quelconque
  - □ dans lequel ces fonctions sont appelées dans le main
  - □ alphabétique
  - $\Box\,$  dans lequel vous avez déclaré ces fonction
- 3. Sous unix (ou linux), la commande cd permet de :
  - $\hfill \square$ récupérer un programme arrêté avec la commande  $\verb"ab"$
  - $\Box\,$ jouer de la musique
  - $\Box$  changer de répertoire courant
  - $\Box\,$  détruire un fichier
  - $\square$  ouvir un bureau partagé (common desktop)
- 4. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\[ x = racine(2/3);
\[ x = racine(x * x) - racine(x);
\[ x - 1 = racine(x);
\[ x = racine(racine(x)*racine(x));
\]
```

```
5. Soit le programme principal suivant :
  int main()
  {
   int a = 3;
   int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  }
  appelant la fonction f ainsi définie :
  int f(int a int b)
```

int f(int a, int b)
{
 a = a + b;
 return a;
}

L'affichage dans le main est le suivant :

- $\Box$  f(a,b)=8, a=3, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=13, a=8, b=5
- $\Box$  f(a,b)=8, a=8, b=5
- 6. L'écriture  $\underline{101}$  en binaire correspond au nombre naturel :
  - $\Box$  4
  - $\Box$  5
  - $\square$  101
  - $\square$  3
- 7. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
  - $\square$  n = carre(n);
  - $\square$  int carre(2);
  - $\square$  int n = carre();
  - $\square$  n = carre(int n);
- 8. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.

Ш	#define	N	3

- $\square$  #define taille = N
- $\square$  #define N = 3
- $\square$  #define taille = 3

- 9. Dans la commande gcc, l'option -Wall signifie :
  - $\Box\,$  que l'on veut voir tous les avertissements
  - $\Box\,$ qu'il faut lancer un déboggueur
  - $\square$  qu'il faut indenter le fichier source
  - $\Box\,$  qu'on veut changer alétoirement de fond d'écran
- 10. Si cette erreur apparaît à la compilation :

```
Undefined symbols :"_prinft" ou
```

référence indéfinie vers « prinft » que doiton chercher dans le programme?

- $\Box\,$  une faute de frappe dans un appel de fonction
- $\Box\,$  une directive préprocesseur  $\# {\tt include}$  man quante
- □ une variable non déclarée
- $\square$  un caractère interdit en C
- 11. Quels calculs peut-on programmer en programmation structurée?
  - ☐ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
  - $\square$  certains programmes sont de vrais plats de spaghetti
  - □ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
- 12. Au début de la fonction main() on place le code :

```
char i;
for (i = 'A'; i <= 'F'; i = i + 1)
{
   printf("%c", i);
}</pre>
```

printf("\n");
Alors l'affichage sera :

- □ сссссс
- $\Box$  A
- $\square$  ABCDEF
- $\square$  i

```
13. Pour déclarer une procédure afficher_date qui prend
                                                           16. Si cette erreur apparaît à la compilation :
                                                                                                                            }
                                                               error: expected ';' before '}' token que doit-
   en argument un struct date_s et affiche le contenu
                                                                                                                            return 7;
                                                               on chercher dans le programme?
   du struct, on écrit :
     □ struct date_s afficher_date(struct date_s | d);
                                                                 ☐ un point-virgule manquant
     □ void afficher_date(date_s d);
                                                                 \square une accolade en trop
                                                                                                                            \Box 5
     □ void afficher_date(struct date_s d);
                                                                 ☐ un point-virgule en trop
                                                                                                                            \Box 7
     ☐ int afficher_date(date_s d);
                                                                 \square une accolade manquante
                                                                                                                            \Box 0
14. Pour l'extrait de programme suivant :
                                                           17. Le code suivant :
     int i = 0;
                                                                int age = 15;
     int j = 0;
                                                                if (age < 18)
     for (i = 0; i < 3; i = i + 1)
                                                                {
                                                                     printf("Mineur\n");
         for (j = 0; j < 2; j = j + 1)
                                                                }
                                                                else
              printf("%d ", i);
                                                                {
         }
                                                                    printf("Majeur\n");
                                                                }
    printf("\n");
                                                                                                                      20. Le code suivant :
   qu'est ce qui sera affiché?
                                                               affichera:
                                                                                                                           int age = 20;
     \Box 0 1 0 1 0 1
                                                                                                                           if (age < 18)
                                                                 □ Majeur
     \Box 1 2 3 1 2
                                                                 \square rien
     \Box 0 0 1 1 2 2
                                                                 □ Mineur
     \Box 0 1 2 0 1 2
                                                                    Majeur
15. Un fichier source est:
                                                                 □ Mineur
     \square un document de référence du système
                                                                                                                          affichera:
                                                           18. Soit la fonction g définie par :
     □ un document qui doit être protégé
                                                                                                                            ☐ Mineur
                                                               int g(int a)
     \square un document illisible pour les humains
                                                                                                                               Majeur
     \square un fichier texte qui sera traduit en instructions
                                                                                                                            \square rien
                                                                 printf("a = \n", %d);
        processeur
                                                                 if (1 > 0)
                                                                                                                            ☐ Mineur
     \square un fichier que l'ont doit citer dans les documents
                                                                 {
                                                                                                                            □ Majeur
        produits sur l'ordinateur
                                                                   return 5;
```

```
Alors l'expression g(0) prendra la valeur :
19. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \square void pgcd(int x, int y);
     \Box int pgcd(int y, int x);
     \Box int pgcd(int x, int x);
     \Box int pgcd(int x, y);
          printf("Mineur\n");
    printf("Majeur\n");
```

### Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt : □ un débogueur □ printf("Valeur de n ? %d\n", n); □ scanf("%d", &n); □ printf("Valeur de n ? %g\n", n); 2. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <stdio.h> ☐ #include <studlib.h> ☐ #appart <stdlib.h> ☐ #include <studio.h> 3. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme? □ un désaccord entre la déclaration et la définition d'une fonction ☐ une directive préprocesseur #include manquante ☐ une fonction appelée avant sa déclaration □ une fonction déclarée mais non définie 4. Sous unix (ou linux), la commande cd permet de : □ détruire un fichier ☐ récupérer un programme arrêté avec la commande ab □ changer de répertoire courant □ ouvir un bureau partagé (common desktop) □ jouer de la musique 5. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  int carre(2);  $\square$  n = carre(n);  $\square$  n = carre(int n);

 $\square$  int n = carre();

```
6. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
  , que doit-on chercher dans le programme?
     \square une directive préprocesseur \#include manquante
    \square une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
    □ un désaccord entre la déclaration et la définition
        d'une fonction
7. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
  La valeur de somme affichée est :
     \square 0
     \square 10
     \square 15
     \Box 6
8. Pour l'extrait de programme suivant :
     int produit = 1;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
     {
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 4
    \Box 0
     \square 8
     \Box 16
9. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     \square analyse harmonique
    □ analyse sémantique
     \square analyse lexicale
     \square analyse syntaxique
```

10.	Un bit est:
	$\Box$ un battement d'horloge processeur
	$\Box$ la longueur d'un mot mémoire
	$\square$ un chiffre binaire (0 ou 1)
	$\hfill\Box$ l'instruction qui met fin à un programme
11.	Soit le programme principal suivant :
	<pre>int main() {</pre>
	int a = 3;
	<pre>int b = 5; printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b) return EXIT_SUCCESS; }</pre>
	appelant la fonction f ainsi définie :
	<pre>int f(int a, int b) {</pre>
	a = a + b;
	<pre>return a; }</pre>
	L'affichage dans le main est le suivant :
	$\Box$ f(a,b)=8, a=8, b=5
	$\Box$ f(a,b)=8, a=3, b=5
	$\Box$ f(a,b)=13, a=8, b=5
	$\Box$ f(3,5)=8, a=3, b=5
12.	L'écriture $\underline{101}$ en binaire correspond au nombre naturel :
	$\square$ 3
	$\Box$ 4
	□ 5
	□ 101

```
15. Après exécution jusqu'à la ligne 15 du programme C:
13. Le code suivant :
                                                                                                                           18. Pour l'extrait de programme suivant :
     int somme = 0;
                                                               10
                                                                     int main() {
                                                                                                                                  int produit = 0;
                                                                                                                                  int serie[4] = \{2, 2, 2, 2\};
     int i;
                                                               11
                                                                          int x = 5;
                                                                                                                                  for (i = 0; i < 4; i = i + 1)
     for (i = 1; i < 4; i = i + 1)
                                                               12
                                                                          int y;
                                                               13
                                                                                                                                    produit = produit * serie[i];
                                                               14
       somme = somme + i;
                                                                          y = x;
                                                                15
     printf("%d", somme);
                                                                16
                                                                                                                                  printf("produit = %d", produit);
                                                               17
                                                                                                                               La valeur affichée est :
   affichera:
                                                                   \square le programme affiche "Faux"
                                                                                                                                 \square 8
      \Box 0
                                                                   \square la variable y vaut 5
                                                                                                                                 \Box 0
      \Box 6
                                                                   \Box la variable x vaut 0
                                                                                                                                  \Box 16
      \square 42
                                                                                                                                  \Box 4
                                                                   \square la variable x vaut 5 et la variable y vaut 0
      \Box 1
                                                                                                                           19. Vous utilisez une boucle while quand :
                                                             16. Un programme en langage C doit comporter une et une
                                                                                                                                 \square vous ne connaissez pas le nombre d'itérations de
14. Au début de la fonction main() on place le code :
                                                                  seule définition de la fonction :
                                                                                                                                     la boucle à l'avance
                                                                   \square init
     char i;
                                                                                                                                  □ vous n'avez pas déclaré de fonction
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                                   □ begin
                                                                                                                                 □ vous avez déjà fait un for dans le même pro-
                                                                   \square main
                                                                                                                                    gramme principal
       printf("%c", i);
                                                                   \square include
                                                                                                                                 \Box l'incrément de la variable de boucle n'est pas 1
     printf("\n");
                                                             17. Si x est une variable réelle (de type double) alors
                                                                                                                           20. Laquelle de ces écritures correspond à la déclaration
   Alors l'affichage sera :
                                                                 x = 3/2 lui affecte la valeur :
                                                                                                                               d'une variable de type caractère en langage C?
      \square i
                                                                    \Box 1
                                                                                                                                 ☐ char c;
      □ A
                                                                   \Box 0
                                                                                                                                 ☐ int char:
                                                                                                                                 ☐ char "c";
      ☐ ABCDEF
                                                                   \square 0.5
                                                                                                                                  □ char 'c';
                                                                   \square 1.5
     □ cccccc
```

### Éléments d'informatique – contrôle continue

 $\Box$  0

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Quel est le problème d'un programme comportant les
  lignes suivantes?
  while (1)
     printf("coucou\n");
    □ il n'affiche rien
     ☐ il risque d'afficher bonjour à la place de coucou
    \square il ne compile pas
     \square il comporte une boucle infinie
2. Le code suivant :
    int i:
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
   printf("\n");
  affichera:
    \Box 4 3 2 1 0
    \square 4 3 2 1
    \Box 01234
     \square 1 2 3 4
3. Si cette erreur apparaît à la compilation :
  error: expected ';' before '}' token que doit-
  on chercher dans le programme?
     ☐ un point-virgule manquant
     □ un point-virgule en trop
     \square une accolade en trop
     \square une accolade manquante
4. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
     \square printf("Valeur de n ? %g\n", n);
     \square un débogueur
     □ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
```

```
5. Le bus système sert à :
     □ transporter les processus du tourniquet au pro-
        cesseur
    ☐ Écrire des données sur le dique dur
     ☐ Arriver à l'heure en cours
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
6. Pour l'extrait de programme suivant :
     int somme = 0;
     int serie[4] = \{2, 4, 10, 4\};
     for (i = 0; i < 4; i = i + 1)
     {
       somme = somme + serie[i];
     printf("somme = %d",somme);
   La valeur de somme affichée est :
     \square 16
     \square 3
    \square 20
     \Box 6
7. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3:
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
```

```
8. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 16
      \Box 4
      \Box 0
      \square 8
 9. Sous unix (ou linux), la commande 1s permet de :
      \square afficher la liste de fichiers contenus dans un
         répertoire
      \square voir des clips musicaux
      □ afficher le contenu d'un fichier texte
      □ compiler un programme
10. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      \square int tab[] = 5;
      \square int toto[5];
      ☐ int toto[taille=5];
      ☐ int[] new tableau(5);
      ☐ char tableau[5]:
11. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x = racine(2/3):
     \square x = racine(racine(x)*racine(x));
      \square x - 1 = racine(x):
      \Box x = racine(x * x) - racine(x);
12. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square int n = pgcd();
      \square int pgcd(2);
      \square n = pgcd(int p, int q);
      \square n = pgcd(n, 3);
```

```
13. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
         }
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 3 0 1 2
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 0 1 2
14. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
     if (a > 0)
        return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 5
     \Box 0
```

□ 1	$\hfill\Box$ dans lequel vous avez déclaré ces fonction
$\Box$ 4	$\Box$ dans lequel ces fonctions sont appelées dans le
15. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :	main □ alphabétique
<ul><li>□ void afficher_date(struct date_s d);</li><li>□ struct date_s afficher_date(struct date_s</li></ul>	18. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?
<pre>□ void afficher_date(date_s d);</pre> □ int afficher_date(date_s d);	□ A && B
16. Après exécution jusqu'à la ligne 14 du programme C : 10 int main() {	☐ !(!A    B) == (A && !B) ☐ (A == TRUE) && (B == TRUE) ☐ (!A    B)
11 int x = 5; 12 13 printf(" x = %d\n", 2);	19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
14 15 16 }	□ analyse syntaxique □ analyse sémantique
□ le terminal affiche $x = 2$ □ le terminal affiche $x = 5$ □ le terminal affiche $5$	□ analyse harmonique □ analyse lexicale  20. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :
$\Box$ le terminal affiche "Faux"	☐ int afficher_menu(int char);
17. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :	☐ char afficher_menu(printf("menu")); ☐ int afficher_menu(); ☐ void afficher_menu();
$\Box$ un ordre quel conque	$\square$ double afficher_menu();

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Sur unix (ou linux), la commande mkdir permet de : □ créer un fichier texte □ changer de répertoire courant □ ouvrir un fichier texte □ créer un répertoire 2. Vous utilisez une boucle while quand :  $\square$  vous n'avez pas déclaré de fonction □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ vous avez déjà fait un for dans le même programme principal  $\square$  l'incrément de la variable de boucle n'est pas 13. Le code suivant : int age = 15; if (age < 18) { printf("Mineur\n"); } else printf("Majeur\n"); } affichera: □ rien ☐ Mineur □ Majeur ☐ Mineur Majeur 4. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme? □ un caractère interdit en C ☐ une directive préprocesseur #include manquante  $\square$  une variable non déclarée  $\square$  une faute de frappe dans un appel de fonction

```
5. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
       return f(a - 1) + 1;
     return 4;
  }
   Alors l'expression f(1) prendra la valeur :
     \square 0
     \square 5
     \Box 4
     \Box 1
6. Une de ces manière de composer les blocs de pro-
  grammes ne fait pas partie des opérations de la pro-
   grammation structurée:
     □ retourner un bloc
     □ sélectionner entre deux blocs à l'aide d'une condi-
        tion
     □ répéter un bloc tant qu'une condition est vérifée
     \square mettre les blocs en séquence les uns à la suite des
        autres
7. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     □ begin
     \square include
     \square init
     \square main
8. Les lignes
  int i;
  int x=0;
  for(i=0,i<5,i=i+1)
  {
     x=x+1;
     □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
```

```
□ comportent une erreur qui ne sera pas détectée
      \square ne comportent aucune erreur
      □ comportent une erreur qui sera détectée au cours
         de l'édition de lien
 9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square exposant(double x, int n, int r);
      \square double exposant(double x, int n);
      \square int exposant(double n, int x);
      \square void exposant(double x^n);
10. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
      \square void pgcd(int x, int y);
     \Box int pgcd(int x, int x);
      \Box int pgcd(int x, y);
      \square int pgcd(int y, int x);
11. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n", f(a,b), a,b);
     return EXIT_SUCCESS;
    appelant la fonction f ainsi définie :
    int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
      \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
      \Box f(a,b)=13, a=8, b=5
      \Box f(a,b)=8, a=8, b=5
```

```
12. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
      □ saisie_utilisateur(scanf(%d));
     ☐ int saisie_utilisateur();
     □ void saisie_utilisateur(char c);
      □ void saisie_utilisateur(int n);
13. Le bus système sert à :
      \square Arriver à l'heure en cours
     □ transporter les processus du tourniquet au pro-
        cesseur
     \Box Transférer des données et intructions entre pro-
        cesseur et mémoire
      ☐ Écrire des données sur le dique dur
14. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
      \square #define taille = N
     \square #define taille = 3
      \square #define N = 3
      □ #define N 3
15. Le code suivant :
     int i:
     for (i = 4; i \ge 0; i = i - 1)
     {
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \Box \ 4\ 3\ 2\ 1\ 0
     \square 4 3 2 1
     \Box 1 2 3 4
```

```
16. Au début de la fonction main() on place le code :
                                                          18. Une variable booléenne est un variable :
    char i;
                                                                □ NaN (not a number, qui n'est pas un nombre)
    for (i = 'A'; i \le 'F'; i = i + 1)
                                                                □ à laquelle une valeur vient d'être affectée
                                                                ☐ qui est vraie ou fausse
       printf("%c", i);
                                                                ☐ jamais nulle
    printf("\n");
                                                                ☐ réelle positive
   Alors l'affichage sera:
                                                          19. Soit la fonction f définie par :
     □ ccccc
                                                              int f(int a)
     ☐ ABCDEF
     □ A
                                                                printf("a = \n", %d);
                                                                if (a > 0)
     \Box i
17. Soit un programme contenant les lignes suivantes :
                                                                  return 3;
    int i = 0;
                                                                return 4;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
                                                              Alors l'expression f(0) prendra la valeur :
         for (j = 0; j < 5; j = j + 1)
                                                                \square 3
                                                                \Box 4
                . . .
         }
                                                                \Box 0
    printf("j = %d\n", j);
                                                          20. Pour déclarer une procédure afficher_date qui prend
                                                              en argument un struct date_s et affiche le contenu
                                                              du struct, on écrit :
   qu'est ce qui sera affiché par ce printf?
                                                                ☐ int afficher_date(date_s d);
     \Box j = %d
                                                                ☐ struct date_s afficher_date(struct date_s d);
     \Box j = 5
     \Box j = 0
                                                                □ void afficher_date(struct date_s d);
     \Box j = 4
                                                                □ void afficher_date(date_s d);
```

### Éléments d'informatique – contrôle continue

 $\Box$  4

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par

· r	réponse fausse. Durée : 20 minutes.
1.	Pour compiler un programme $prog.c$ , on utilise la ligne de commande :
	$\square$ gcc prog.exe -Wall -o prog.c
	☐ gcc prog.c -o -Wall prog.exe
	☐ gcc -Wall prog.c -o prog.exe
	$\square$ gcc -Wall prog.exe -o prog.c
2.	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	$\square$ analyse syntaxique
	$\Box$ analyse lexicale
	$\hfill\Box$ analyse harmonique
	$\Box$ analyse sémantique
3.	Au début de la fonction main() on place le code :
	<pre>char i; for (i = 'A'; i &lt;= 'F'; i = i + 1) {    printf("%c", i); } printf("\n");</pre>
	Alors l'affichage sera :
	□ A
	□ ABCDEF
	□ сссссс
	□i
4.	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	$\Box$ une accolade en trop
	$\Box$ une accolade man quante
	$\square$ un point-virgule manquant

□ un point-virgule en trop

```
5. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
 11
           int x = 5;
 12
           int y = 3;
 13
 14
           x = y;
 15
 16
 17
      }
     \square la variable y vaut 5
     \square la variable x vaut 3
     \square la variable x vaut 5 et la variable y vaut 3
     ☐ le programme affiche "Faux"
6. La virtualisation de la mémoire permet notamment de
   stocker des portions inactives de la mémoire de travail
   sur le disque dur. Mais on perd :
     □ certaines données de la mémoire de travail
     \square les fichiers du disque
     □ en temps d'accès
     ☐ des processus
7. Si a et b sont deux variables de type:
   struct toto_s
     int n;
     double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \square (a.n == b.n) \&\& (a.x == b.x)
     \square a{n, x} == b{n, x}
     \Box a = b
     □ a == b
8. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \square 101
     \square 3
     \Box 5
```

```
9. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     □ void saisie_utilisateur(int n);
     ☐ int saisie_utilisateur();
     ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
10. Vous utilisez une boucle while quand:
     □ vous n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
      □ l'incrément de la variable de boucle n'est pas 1
     \square vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
11. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
     □ une directive préprocesseur #include manquante
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     □ une fonction déclarée mais non définie
      ☐ une fonction appelée avant sa déclaration
12. Avant de faire appel à une fonction il est nécessaire
   de:
     □ l'avoir déclarée
     □ l'avoir déclarée et définie
     \square avoir défini une constante symbolique de la taille
        de cette fonction
     □ l'avoir définie
13. Soit le programme principal suivant :
   int main()
   ₹
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
```

```
appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=13, a=8, b=5
14. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \Box for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
15. Le langage C est un langage
     □ lu, écrit, parlé
     □ composé
     □ compilé
     □ interprété
16. Le code suivant :
     int age = 18;
     if (age < 18)
     {
```

```
printf("Mineur\n");
    }
     else
    {
         printf("Majeur\n");
    }
    affichera:
     \square rien
     □ Majeur
     ☐ Mineur
        Majeur
     ☐ Mineur
17. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 8
     \Box 7
     □ 111
     \square 3
18. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
```

```
Alors l'expression f(1) prendra la valeur :
      \Box 4
      \Box 5
      \Box 1
      \Box 0
19. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
         produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 0
      \Box 4
      \square 8
      \Box 16
20. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
    d'écrire :
     \square int n = carre();
     \square n = carre(n);
```

 $\square$  n = carre(int n);

 $\square$  int carre(2);

 $\square$  7

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Un fichier source est:
     □ un document illisible pour les humains
     □ un document de référence du système
     □ un fichier texte qui sera traduit en instructions
        processeur
     ☐ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
     □ un document qui doit être protégé
2. Pour l'extrait de programme suivant :
     int somme = 0;
     int serie[4] = \{2, 4, 10, 4\};
     for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
     printf("somme = %d",somme);
   La valeur de somme affichée est :
     \square 20
     \square 16
     \Box 6
     \square 3
3. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
        return 5;
     }
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \square 5
     \square 0
```

```
4. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
    }
    printf("\n");
   affichera:
     \square 1 2 3 4
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 01234
     \square 4 3 2 1
5. Quel est l'opérateur de différence en C:
     □ !=
     \Box \neq
     \Box !
     □ <>
6. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre :
     □ dans lequel ces fonctions sont appelées dans le
        main
     □ alphabétique
     □ un ordre quelconque
     □ dans lequel vous avez déclaré ces fonction
7. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     □ créer un fichier texte
     □ ouvrir un fichier texte
     □ changer de répertoire courant
8. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(racine(x)*racine(x));
     \Box x = racine(x * x) - racine(x);
     \square x - 1 = racine(x);
     \square x = racine(2/3);
```

```
9. Le code suivant :
     int i:
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
     printf("\n");
    affichera:
     \Box 02468
     \Box 0123456
     \Box 0246
     \Box 01234567
10. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \square int mccarthy(int 2);
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
     \square n = mccarthy();
11. Un registre du processeur est :
      un composant qui contient la liste des fichiers du
        système
      □ une gamme de fréquence de fonctionnement du
        processeur
      □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
      □ une unité de calcul spécialisée de l'ordinateur
12. Le langage C est un langage
      □ composé
     □ interprété
     □ compilé
      □ lu, écrit, parlé
```

```
13. Soit le programme principal suivant :
                                                                  \square (n<=a) && (n<=b)
                                                                                                                               ☐ Mineur
                                                                  \square (a<n) || (n>b)
                                                                                                                               ☐ Mineur
   int main()
                                                                                                                                  Majeur
   {
                                                                  □ a<=n<=b
     int a = 3;
                                                                                                                               \square rien
                                                                  \square (a<=n) && (n<=b)
     int b = 5;
                                                                                                                               □ Majeur
                                                            15. Si n est une variable entière pour demander sa valeur
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                         18. Si x est une variable réelle (de type double) alors
                                                                à l'utilisateur, on utilise plutôt :
     return EXIT_SUCCESS;
                                                                                                                             x = 3/2 lui affecte la valeur :
                                                                  □ un débogueur
                                                                                                                               \Box 0
   appelant la fonction f ainsi définie :
                                                                  ☐ printf("Valeur de n ? %d\n", n);
                                                                                                                               \square 0.5
   int f(int a, int b)
                                                                  □ scanf("%d", &n);
                                                                                                                               \Box 1
                                                                  ☐ printf("Valeur de n ? %g\n", n);
      a = a + b;
                                                                                                                               \square 1.5
                                                            16. Sous unix (ou linux), la commande cd permet de :
      return a;
                                                                                                                         19. Au début de la fonction main() on place le code :
                                                                  □ changer de répertoire courant
                                                                                                                               char b = 'A';
   L'affichage dans le main est le suivant :
                                                                  \square jouer de la musique
                                                                                                                              b = b + 2;
     \Box f(3,5)=8, a=3, b=5
                                                                  □ récupérer un programme arrêté avec la commande
                                                                                                                              printf("%c\n", b);
                                                                     ab
     \Box f(a,b)=8, a=3, b=5
                                                                                                                             Alors l'affichage sera:
                                                                  \square détruire un fichier
     \Box f(a,b)=13, a=8, b=5
                                                                                                                               □ b
                                                                  □ ouvir un bureau partagé (common desktop)
     \Box f(a,b)=8, a=8, b=5
                                                                                                                               \square B
                                                            17. Le code suivant :
14. On souhaite faire une boucle de contrôle de saisie : tant
                                                                                                                               \square A
   que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                 int age = 15;
                                                                                                                               \Box C
   recommence la saisie de n. Soit le programme suivant :
                                                                 if (age < 18)
                                                                                                                         20. Si cette erreur apparaît à la compilation :
                                                                 {
     int a = 0;
                                                                      printf("Mineur\n");
                                                                                                                             erreur: conflicting types for 'max', que doit-
     int b = 20;
                                                                                                                             on chercher dans le programme?
                                                                 }
     int n;
     scanf("%d", &n);
                                                                 else
                                                                                                                               \square un désaccord entre la déclaration et la définition
                                                                 {
     while(cond)
                                                                                                                                  d'une fonction
                                                                      printf("Majeur\n");
                                                                                                                               \Box une fonction déclarée mais non définie
                                                                 }
       scanf("%d", &n);
                                                                                                                               \square une fonction appelée avant sa déclaration
                                                                                                                               □ une directive préprocesseur #include manquante
   Quelle est la condition cond :
                                                                affichera:
```

т		-1
	100000	- 1
	acence	

int a = 0;

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  ${\tt n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  ${\tt n}$ . Soit le programme suivant :

2. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
    x = racine(2/3);
    x = racine(racine(x)*racine(x));
    x = racine(x * x) - racine(x);

    x - 1 = racine(x);
```

3. Au début de la fonction main() on place le code :

 $\Box$  C

□В

```
4. Dans la commande gcc, l'option -Wall signifie :
    □ qu'on veut changer alétoirement de fond d'écran
    ☐ qu'il faut indenter le fichier source
    ☐ que l'on veut voir tous les avertissements
     □ qu'il faut lancer un déboggueur
5. Pour déclarer une variable qui sera utilisée comme va-
  riable de boucle on peut utiliser l'instruction
    □ loop i;
    \square int loop n;
    \square int k;
    □ int %d:
6. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit:
    □ void saisie_utilisateur(char c);
    ☐ int saisie_utilisateur();
    ☐ saisie_utilisateur(scanf(%d));
    □ void saisie_utilisateur(int n);
7. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int i = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
        }
    }
  qu'est ce qui sera affiché?
    \Box 0 1 0 1 0 1 0 1
    \Box 0 0 0 1 1 1
```

 $\Box$  0 1 2 0 1 2

 $\Box$  1 2 1 2 3

```
8. Une de ces manière de composer les blocs de pro-
    grammes ne fait pas partie des opérations de la pro-
    grammation structurée :
      □ mettre les blocs en séquence les uns à la suite des
         autres
      □ répéter un bloc tant qu'une condition est vérifée
      ☐ sélectionner entre deux blocs à l'aide d'une condi-
         tion
      □ retourner un bloc
 9. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      □ l'avoir déclarée
      □ l'avoir déclarée et définie
10. L'ordonnancement par tourniquet permet :
      ☐ d'entretenir l'illusion que les processus tournent
         en parallèle
      □ de doubler la mémoire disponible
      \square de ne pas perdre de temps avec la commutation
         de contexte
      □ d'afficher des ronds colorés à l'écran
11. Le type des réels en C est :
      ☐ double
      □ real
      □ char
      \square int
12. Si le code :
    struct toto_s
      int n:
      double x;
   };
    précède la fonction main(), alors on peut écrire en
   début de main() :
      \square toto_s n, x;
```

☐ int toto.n = 3;	15. Si factorielle est une fonction prenant en entrée un	$\hfill\Box$ un ordre quel conque
☐ struct toto_s toto;	entier et renvoyant un entier, il est correct d'écrire :	☐ dans lequel vous avez déclaré ces fonction
$\square$ int struct toto_s = {3, -1e10};	$\Box$ int factorielle(int 2);	$\Box$ dans lequel ces fonctions sont appelées dans le
$\Box$ toto_s struct z = {3, 0.5};	☐ n = factorielle();	main
13. Quels calculs peut-on programmer en programmation	$\Box$ n = factorielle(p, q);	□ alphabétique
structurée?  □ il y a des calculs programmables en langage ma-	☐ printf("%d", factorielle(n));	□ arpnabetique
chine et qui ne sont pas programmables en pro-	16. Un bit est:	19. Sous unix (ou linux), pour créer un répertoire TP4
grammation structurée	□ la longueur d'un mot mémoire	dans le répertoire courant on peut utiliser la com-
$\Box$ il y a des calculs programmables en programma-	☐ l'instruction qui met fin à un programme	mande:
tion structurée qui ne sont pas programmables en langage machine	□ un battement d'horloge processeur	$\square$ yppasswd
□ en programmation structurée on peut program-	$\square$ un chiffre binaire (0 ou 1)	$\square$ kwrite TP4
mer tous les calculs programmables en langage	17. Laquelle de ces écritures correspond à la déclaration	☐ mkdir TP4
machine	d'une variable de type caractère en langage C?	□ new TP4
□ certains programmes sont de vrais plats de spa-	☐ char 'c';	
ghetti	$\square$ char c;	20. Un programme en langage C doit comporter une et une
14. Le bus système sert à :  ☐ transporter les processus du tourniquet au pro-	☐ char "c";	seule définition de la fonction :
cesseur	$\square$ int char;	$\Box$ include
$\Box$ Écrire des données sur le dique dur	18. Vous avez déclaré préalablement un ensemble de fonc-	$\Box$ begin
$\square$ Arriver à l'heure en cours	tions utilisées par votre programme principal. L'ordre	$\Box$ init
☐ Transférer des données et intructions entre processeur et mémoire	dans lequel vous devez maintenant définir ces fonctions est l'ordre :	$\square$ main

### Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

T Émina des dennées sur la dieux dun

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution jusqu'à la ligne 15 du programme C:
```

```
int main() {
10
11
         int x = 5;
12
         int y;
13
14
         y = x;
15
16
17
```

□ le programme affiche "Faux"

```
\square la variable y vaut 5
```

 $\Box$ la variable x vaut 5 et la variable y vaut 0

 $\square$  la variable x vaut 0

2. Le code suivant :

```
int i;
 for (i = 4; i \ge 0; i = i - 1)
     printf("%d ", i);
printf("\n");
affichera:
 \Box 1234
 \square 4 3 2 1
```

3. Une variable booléenne est un variable :

```
□ à laquelle une valeur vient d'être affectée
```

□ NaN (not a number, qui n'est pas un nombre)

☐ réelle positive

 $\Box 01234$ 

 $\Box \ 4\ 3\ 2\ 1\ 0$ 

□ qui est vraie ou fausse

□ jamais nulle

4. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
  scanf("%d", &n);
```

Quelle est la condition cond :

```
□ a<=n<=b
```

```
\square (a<n) || (n>b)
```

```
□ (n<=a) && (n<=b)
```

 $\square$  (a<=n) && (n<=b)

5. Pour l'extrait de programme suivant :

```
int i = 0;
 int i = 0;
 for (i = 0; i < 3; i = i + 1)
     for (j = 0; j < 2; j = j + 1)
         printf("%d ", i);
     }
printf("\n");
qu'est ce qui sera affiché?
```

```
\Box 1 2 3 1 2
```

 $\Box$  0 0 1 1 2 2

 $\Box$  0 1 0 1 0 1

 $\Box$  0 1 2 0 1 2

6. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

char	"c"	;
,		

char	с;

$\square$ char	'c'
----------------	-----

_	т	1			`
7	Lе	bus	système	sert	a.
		CLO	D, DOCTITO	DOLU	C.

cesseur

Ш	Ecrire des d	.ОПП	iees sur ie	aiqt	ie dur		
	transporter	les	processus	$\mathrm{d}\mathrm{u}$	tourniquet	au	pro-

☐ Arriver à l'heure en cours

${\it Transf\'erer}$	${\rm des}$	${\rm donn\acute{e}es}$	$\operatorname{et}$	intructions	entre	pro
cesseur et i	mém	noire				

8. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :

] #appart	<pre><stdlib.h></stdlib.h></pre>
-----------	----------------------------------

$\neg$		17 .1	1.
	#include	<stuallo.< td=""><td>. n&gt;</td></stuallo.<>	. n>

	#include	<studio.h></studio.h>
--	----------	-----------------------

☐ #include <stdio.h>

9. Vous utilisez une boucle while quand:

□ vous ne connaissez	pas le	nombre	d'itérations	d
la boucle à l'avance				

	l'incrément	de l	la variable d	de bou	cle n'est	pas 1
_	I IIIOI CIIICIIO	CLC I	ia railani	ac boa	OIC II CDC	PODL

vous	avez	déjà	fait	un	for	dans	le	${\rm m\^{e}me}$	pro-
gram	me pi	rincip	al						

□ vous n'avez pas déclaré de fonction

10. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :

```
☐ int factorielle(int x);
```

□ Int lactorierie(double n)		int	factorielle(double	n)
-----------------------------	--	-----	--------------------	----

☐ int factorielle();

□ struct int factorielle(int n);

11. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction

☐ char tableau[5];

☐ int toto[taille=5];

 $\square$  int[] new tableau(5);

 $\square$  int tab[] = 5;

 $\square$  int toto[5];

```
12. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
    La valeur de somme affichée est :
      \square 20
      \Box 16
      \Box 6
      \square 3
13. Un fichier source est:
      □ un fichier texte qui sera traduit en instructions
         processeur
      \square un document de référence du système
      □ un document illisible pour les humains
      ☐ un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
      \Box un document qui doit être protégé
14. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square n = mccarthy();
      \square x = mccarthy(n);
      \square int mccarthy(int 2);
      \square n = mccarthy(p, q);
15. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
    du struct, on écrit :
      ☐ struct date_s afficher_date(struct date_s d);
      □ void afficher_date(date_s d);
      □ void afficher_date(struct date_s d);
      ☐ int afficher_date(date_s d);
```

```
16. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
    appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
17. Un enregistrement permet de grouper plusieurs valeurs
    dans:
     \square ses blocs
     \square ses chants
     \square ses champs
     \square ses cases
18. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
```

return f(a - 1) + 1;

```
}
      return 4;
    Alors l'expression f(1) prendra la valeur :
     \Box 4
     \Box 1
     \Box 5
      \square 0
19. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
     □ void saisie_utilisateur(char c);
     ☐ int saisie_utilisateur();
20. Après exécution jusqu'à la ligne 15 du programme C:
 10
       int main() {
 11
            int x = 5;
 12
            int y = 3;
 13
 14
            x = y;
 15
 16
 17
       }
      \square la variable y vaut 5
      □ le programme affiche "Faux"
     \square la variable x vaut 5 et la variable y vaut 3
```

 $\square$  la variable x vaut 3

т		-1
	acence	

# Éléments d'informatique $\underline{-\operatorname{contrôle}}$ continue

Prénom:	Nom:
N° INE ·	
11 11111 .	

	1	
arème : 1 points par réponse juste (unique) ; $-0, 5$ points ar réponse fausse. Durée : $20$ minutes.	☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	10. L'écriture <u>111</u> en binaire correspond au nombre naturel :
1. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit-	☐ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur	□ 8 □ 7
on chercher dans le programme?	6. Un bit est:	□ 111
$\Box$ une variable non déclarée	$\Box$ un chiffre binaire (0 ou 1)	$\square$ 3
$\Box$ une faute de frappe dans un appel de fonction	$\hfill\Box$ l'instruction qui met fin à un programme	
$\Box$ un caractère interdit en C	☐ la longueur d'un mot mémoire	11. Le type des réels en C est :
$\Box$ une directive préprocesseur $\verb"#include"$ manquante	un battement d'horloge processeur	$\square$ int
2. Avant de faire appel à une fonction il est nécessaire de :	7. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?	□ real □ char
□ l'avoir déclarée et définie	une fonction appelée avant sa déclaration	
□ avoir défini une constante symbolique de la taille	☐ une fonction déclarée mais non définie	$\square$ double
de cette fonction    1'avoir déclarée	un désaccord entre la déclaration et la définition d'une fonction	12. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu
☐ l'avoir définie	une directive préprocesseur #include manquante	avec la commande :
	8. Pour déclarer une fonction factorielle qui prend en	☐ printf("x=%d et y=%d\n,x,y");
3. Le bus système sert à :	argument un entier et renvoie sa factorielle on écrit :	☐ printf("x=%x et y=%y\n");
☐ Transférer des données et intructions entre pro- cesseur et mémoire	☐ struct int factorielle(int n);	□ printf("x=%d et y=%d\n",x,y);
□ transporter les processus du tourniquet au pro-	$\Box$ int factorielle(int x);	
cesseur	☐ int factorielle();	$\square$ printf("x=%d et y=%d\n",x y);
☐ Écrire des données sur le dique dur	☐ int factorielle(double n);	13. L'écriture <u>101</u> en binaire correspond au nombre natu-
☐ Arriver à l'heure en cours	9. Pour l'extrait de programme suivant :	rel:
4. Laquelle de ces écritures correspond à la déclaration	<pre>int i = 0; int j = 0;</pre>	□ 101
d'une variable de type caractère en langage C?	for $(i = 0; i < 3; i = i + 1)$	$\square$ 4
□ char c;	{	
☐ char "c";	for $(j = 0; j < 2; j = j + 1)$	   □ 5
☐ char 'c';	printf("%d ", i);	
☐ int char;	}	14. Vous utilisez une boucle while quand:
5. Une segmentation fault est une erreur qui survient	}	☐ l'incrément de la variable de boucle n'est pas 1
lorsque :  □ la division du programme en zones homogènes	printf("\n"); qu'est ce qui sera affiché?	□ vous avez déjà fait un for dans le même pro- gramme principal
échoue	□ 0 1 0 1 0 1 □ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	□ vous n'avez pas déclaré de fonction
$\Box$ le programme tente d'afficher des caractères sur		□ vous ne connaissez pas le nombre d'itérations de
une ligne qui va au delà de la largeur de la fenêtre du terminal	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	la boucle à l'avance

```
15. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     □ Majeur
      ☐ Mineur
        Majeur
      ☐ Mineur
      □ rien
16. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int carre(2);
     \square int n = carre();
```

```
\square n = carre(n);
      \square n = carre(int n);
17. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
         somme = somme + serie[i];
      printf("somme = %d",somme);
    La valeur de somme affichée est :
      \square 3
      \Box 16
      \square 20
      \Box 6
18. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      □ dans lequel ces fonctions sont appelées dans le
      □ alphabétique
      \square un ordre quelconque
      \square dans lequel vous avez déclaré ces fonction
```

```
19. Après exécution jusqu'à la ligne 15 du programme C:
 10
 11
       int main() {
 12
            int x = 5;
 13
 14
           x = 3 * x + 1;
 15
 16
 17
      }
      \square la variable x vaut 16
     \square la variable x vaut -\frac{1}{2}
     □ le programme affiche ****
     \Box le programme affiche x
20. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
     □ void saisie_utilisateur(char c);
```

☐ int saisie\_utilisateur();

### Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int age = 15;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     □ rien
     □ Mineur
     □ Majeur
     □ Mineur
        Majeur
2. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     \square n = factorielle(p, q);
     □ printf("%d", factorielle(n));
     ☐ int factorielle(int 2);
     \square n = factorielle();
3. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 111
     \square 8
     \square 3
     \square 7
4. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \Box 1
     \square 0.5
     \Box 0
```

```
5. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
       return 3;
     return 4;
  Alors l'expression f(0) prendra la valeur :
    \square 0
    \Box 4
    \square 3
6. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
  pressions booléennes suivantes, laquelle a pour valeur
  TRUE?
    \square (A == TRUE) && (B == TRUE)
    □ A && B
    \square !(!A \mid | B) == (A \&\& !B)
     \square (!A || B)
7. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 0; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
           . . .
    printf("j = %d\n", j);
    }
  qu'est ce qui sera affiché?
    \Box j = 5
    \Box j = 0
     \Box j = %d
```

 $\Box$  j = 4

```
8. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 5
      \Box 4
      \square 101
      \square 3
 9. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
          for (j = 0; j < 5; j = j + 1)
          }
     printf("j = %d\n", j);
    qu'est ce qui sera affiché par ce printf?
     \Box j = 5
     \Box i = %d
     \Box j = 0
      \Box j = 4
10. Un registre du processeur est :
      \square un composant qui contient la liste des fichiers du
         système
      □ une gamme de fréquence de fonctionnement du
         processeur
      □ une unité de calcul spécialisée de l'ordinateur
      □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
11. Vous utilisez une boucle while quand :
      □ vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      □ vous n'avez pas déclaré de fonction
      □ l'incrément de la variable de boucle n'est pas 1
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
```

12. Quel est le problème d'un programme comportant lignes suivantes?	les
while (1) {	
<pre>printf("coucou\n"); }</pre>	
$\square$ il ne compile pas	
$\square$ il n'affiche rien	
$\Box$ il risque d'afficher bonjour à la place de couco	u
$\Box$ il comporte une boucle infinie	
13. Avant de faire appel à une fonction il est nécessa de :	aire
$\square$ l'avoir déclarée et définie	
☐ l'avoir déclarée	
$\hfill \square$ avoir défini une constante symbolique de la ta de cette fonction	ille
□ l'avoir définie	
14. Laquelle de ces écritures correspond à la déclarat d'une variable de type caractère en langage C?	ion
☐ char 'c';	
☐ int char;	
☐ char "c";	
□ char c;	

```
15. Le bus système sert à :
     ☐ Arriver à l'heure en cours
     ☐ Écrire des données sur le dique dur
     □ transporter les processus du tourniquet au pro-
         cesseur
     □ Transférer des données et intructions entre pro-
         cesseur et mémoire
16. Dans la commande gcc, l'option -Wall signifie :
     \square qu'on veut changer alétoirement de fond d'écran
     \square que l'on veut voir tous les avertissements
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
17. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle();
     ☐ int factorielle(double n);
     ☐ int factorielle(int x);
     □ struct int factorielle(int n);
18. L'ordonnancement par tourniquet permet :
     □ d'entretenir l'illusion que les processus tournent
         en parallèle
     \square de ne pas perdre de temps avec la commutation
         de contexte
     \square de doubler la mémoire disponible
     □ d'afficher des ronds colorés à l'écran
```

```
19. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
     }
     printf("\n");
    affichera:
     \Box 0246
     \Box 0 1 2 3 4 5 6 7
     \Box 02468
     \Box 0 1 2 3 4 5 6
20. Si a et b sont deux variables de type :
   struct toto_s
   {
      int n;
      double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     □ a == b
     \square (a.n == b.n) && (a.x == b.x)
```

 $\square$  a{n, x} == b{n, x}

 $\Box$  a = b

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. L'écriture 111 en binaire correspond au nombre naturel:  $\square$  3  $\Box$  7  $\square$  111  $\square$  8 2. Soit le programme principal suivant : int main() int a = 3; int b = 5; printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b$ ) return EXIT\_SUCCESS; } appelant la fonction f ainsi définie : int f(int a, int b) a = a + b; return a; L'affichage dans le main est le suivant :  $\Box$  f(a,b)=8, a=8, b=5  $\Box$  f(a,b)=13, a=8, b=5  $\Box$  f(a,b)=8, a=3, b=5  $\Box$  f(3,5)=8, a=3, b=5 3. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme? ☐ une fonction appelée avant sa déclaration □ une fonction déclarée mais non définie ☐ une directive préprocesseur #include manquante □ un désaccord entre la déclaration et la définition d'une fonction

```
4. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
     ☐ int %d:
     \square int k;
     \square int loop n;
     □ loop i;
5. Avant de faire appel à une fonction il est nécessaire
   de:
     □ l'avoir déclarée
    \Box avoir défini une constante symbolique de la taille
        de cette fonction
     □ l'avoir déclarée et définie
     □ l'avoir définie
6. Le bus système sert à :
     ☐ Arriver à l'heure en cours
    ☐ Écrire des données sur le dique dur
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     □ transporter les processus du tourniquet au pro-
        cesseur
7. Le code suivant :
    int i:
    for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 4 3 2 1
     \Box 0123
     \Box 01234
```

 $\Box \ 4\ 3\ 2\ 1\ 0$ 

```
8. Après exécution du programme :
       lecture 8 r0
       valeur 3 r1
       mult r1 r0
       valeur 1 r2
       add r2 r0
       ecriture r0 8
       stop
      □ le terminal affiche 8
      □ la case mémoire 8 contiendra 0
      □ la case mémoire 8 contiendra 16
      \square le bus explose
 9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
      □ void exposant(double x^n);
     \square exposant(double x, int n, int r);
      \square double exposant(double x, int n);
      \square int exposant(double n, int x);
10. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     \Box char c;
     □ char 'c';
     ☐ int char;
     □ char "c":
11. Quel est le problème d'un programme comportant les
    lignes suivantes?
   while (1)
      printf("coucou\n");
      \square il comporte une boucle infinie
      \square il ne compile pas
     \square il risque d'afficher bonjour à la place de coucou
      □ il n'affiche rien
```

12. Un bit est:	15. On considère deux
$\Box$ un chiffre binaire (0 ou 1)	lisées à TRUE et FAI
□ un battement d'horloge processeur	pressions booléenne
☐ l'instruction qui met fin à un programme	TRUE?
☐ la longueur d'un mot mémoire	☐ (A == TRUE) 8
13. Si racine est une fonction prenant en entrée un réel	☐ !(!A    B) ==
et renvoyant la racine carrée de cet réel, et que x est	□ A && B
une variable réelle définie et initialisée, il est incorrect	□ (!A    B)
d'écrire :	16. Après exécution jus
$\square$ x = racine(x * x) - racine(x);	
$\square$ x = racine(racine(x)*racine(x));	10
$\square$ x = racine(2/3);	11 int main() { 12 int x = 5;
$\square x - 1 = racine(x);$	13 Int x = 3
14. Soit un programme contenant les lignes suivantes :	14 x = 3 * x
int i = 0;	15
int $j = 0$ ;	16
for (i = 0; i < 2; i = i + 1)	17 }
{	☐ la variable x va
for (j = 0; j < 3; j = j + 1) {	☐ la variable x va
printf("%d ", i);	☐ le programme a
}	$\Box$ le programme :
}	
	17. Pour compiler un
qu'est ce qui sera affiché?	ligne de commande
$\Box$ 0 1 2 0 1 2	☐ gcc prog.exe
□ 1 2 1 2 3	☐ gcc prog.c -
$\Box$ 0 0 0 1 1 1	☐ gcc -Wall pro
$\Box$ 0 1 0 1 0 1 0 1	☐ gcc -Wall pro
	•

```
15. On considère deux variables booléennes A et B initia-
                                                               18. Si cette erreur apparaît à la compilation :
                          LSE respectivement. Parmi les ex-
                                                                   error: expected ';' before '}' token que doit-
                          es suivantes, laquelle a pour valeur
                                                                   on chercher dans le programme?
                                                                     \square un point-virgule en trop
                          && (B == TRUE)
                                                                     \square une accolade en trop
                          = (A && !B)
                                                                     ☐ un point-virgule manquant
                                                                     \Box une accolade manquante
                                                               19. Soit la fonction g définie par :
                          squ'à la ligne 15 du programme C :
                                                                   int g(int a)
                                                                     printf("a = \n", %d);
                                                                     if (1 > 0)
                          + 1;
                                                                        return 5;
                                                                     }
                                                                     return 7;
                          aut -\frac{1}{2}
                                                                   Alors l'expression g(0) prendra la valeur :
                          aut 16
                                                                     \Box 7
                          affiche x
                                                                     \Box 5
                          affiche ****
                                                                     \Box 0
                          programme prog.c, on utilise la
                                                               20. Le langage C est un langage
                                                                     \square compilé
                           -Wall -o prog.c
                                                                     □ interprété
                          o -Wall prog.exe
                                                                     □ composé
                          og.exe -o prog.c
                                                                     □ lu, écrit, parlé
                          og.c -o prog.exe
```

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un bit est:  $\square$  un battement d'horloge processeur  $\square$  la longueur d'un mot mémoire □ l'instruction qui met fin à un programme  $\square$  un chiffre binaire (0 ou 1) 2. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :  $\square$  n = mccarthy();  $\square$  x = mccarthy(n);  $\Box$  int mccarthy(int 2);  $\square$  n = mccarthy(p, q); 3. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\square$  1.5  $\Box$  0  $\Box$  1  $\square$  0.5 4. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme? ☐ un point-virgule manquant  $\square$  une accolade en trop □ un point-virgule en trop □ une accolade manquante 5. Après exécution jusqu'à la ligne 15 du programme C: int main() { 11 int x = 5; 12 int y; 13 14 y = x;15 16 } 17 ☐ le programme affiche "Faux"  $\square$  la variable y vaut 5

 $\square$  la variable x vaut 5 et la variable y vaut 0

 $\square$  la variable x vaut 0

```
6. Pour l'extrait de programme suivant :
     int somme = 0:
     for (i = 0; i < 5; i = i + 1)
       somme = somme + i;
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
  La valeur de somme affichée est :
    \Box 10
    \Box 6
    \Box 0
    \square 15
7. Soit le programme principal suivant :
  int main()
  {
   int a = 3;
   int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
8. Pour déclarer une fonction exposant qui prend en ar-
  gument un réel x et un entier positif n et renvoie la
  valeur de x^n on écrit :
    \square exposant(double x, int n, int r);
    \square int exposant(double n, int x);
```

 $\square$  double exposant(double x, int n);

 $\square$  void exposant(double x^n);

```
9. Si n est une variable entière pour demander sa valeur
    à l'utilisateur, on utilise plutôt :
      ☐ printf("Valeur de n ? %g\n", n);
      □ scanf("%d", &n);
      □ un débogueur
      □ printf("Valeur de n ? %d\n", n);
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      ☐ il risque d'afficher bonjour à la place de coucou
      □ il comporte une boucle infinie
      \square il ne compile pas
      □ il n'affiche rien
11. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
  10
            int x = 5;
 11
 12
            int y = 3;
 13
 14
            x = y;
 15
  16
 17
      }
      \square la variable y vaut 5
      \square la variable x vaut 3
      \square la variable x vaut 5 et la variable y vaut 3
      ☐ le programme affiche "Faux"
```

```
12. Les lignes
   int i;
   int x=0;
   for(i=0,i<5,i=i+1)
      x=x+1;
      □ comportent une erreur qui ne sera pas détectée
      □ comportent une erreur qui sera détectée au cours
         de l'analyse syntaxique
     \square ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
         de l'édition de lien
13. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
      □ main
      \square include
      □ begin
      \square init
14. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
      □ analyse sémantique
      \square analyse syntaxique
     \square analyse harmonique
      \square analyse lexicale
15. Le code suivant :
     int age = 18;
     if (age < 18)
         printf("Mineur\n");
     else
```

```
{
          printf("Majeur\n");
     }
    affichera:
      ☐ Mineur
         Majeur
      □ Mineur
      \square rien
      □ Majeur
16. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
         return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 7
      \Box 0
      \Box 5
17. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      □ alphabétique
      \square dans lequel ces fonctions sont appelées dans le
         main
      \square un ordre quelconque
      \square dans lequel vous avez déclaré ces fonction
```

```
18. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 101
      \square 3
      \Box 4
      \Box 5
19. Au début de la fonction main() on place le code :
     char i;
     for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
     printf("\n");
   Alors l'affichage sera:
      □ ccccc
      \Box i
      □ A
      ☐ ABCDEF
20. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 8
      \Box 4
      \Box 0
      \Box 16
```

### Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si le code :
  struct toto s
  {
     int n;
     double x;
  };
  précède la fonction main(), alors on peut écrire en
  début de main() :
    \square toto_s struct z = {3, 0.5};
    \square toto_s n, x;
    □ struct toto_s toto;
     \Box int struct toto_s = {3, -1e10};
     \square int toto.n = 3;
2. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
     ☐ #include <studlib.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
3. Laquelle de ces écritures correspond à la déclaration
  d'une variable de type caractère en langage C?
     □ char 'c':
    ☐ int char;
    ☐ char "c";
     □ char c;
4. Soit la fonction g définie par :
  int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
```

```
Alors l'expression g(0) prendra la valeur :
     \square 0
     \Box 5
     \square 7
5. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
     \square (!A | | B)
     □ A && B
     \square !(!A \mid | B) == (A \&\& !B)
     \square (A == TRUE) && (B == TRUE)
6. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
  est l'ordre :
     □ dans lequel ces fonctions sont appelées dans le
        main
     □ alphabétique
     □ dans lequel vous avez déclaré ces fonction
     \square un ordre quelconque
7. Soit un programme contenant les lignes suivantes :
    int i = 0:
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
         }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
    \Box j = 4
    \Box j = 5
     \Box j = 0
     \Box j = %d
```

```
8. Le code suivant :
     int i;
     for (i = 4; i >= 0; i = i - 1)
         printf("%d ", i);
     }
     printf("\n");
    affichera:
     \Box \ 4\ 3\ 2\ 1\ 0
     \square 1 2 3 4
     \square 4 3 2 1
     \Box 01234
9. Le langage C est un langage
     □ compilé
      □ lu, écrit, parlé
     □ composé
      □ interprété
10. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
     ☐ #define taille = 3
     \square #define N = 3
     ☐ #define taille = N
     □ #define N 3
11. Au début de la fonction main() on place le code :
     for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
     printf("\n");
   Alors l'affichage sera:
     □ cccccc
     \Box i
     □ A
     ☐ ABCDEF
12. Le type des réels en C est :
     \square int
     \square real
      □ char
```

☐ double

<pre>13. Si racine est une fonction prenant en entrée un réel     et renvoyant la racine carrée de cet réel, et que x est     une variable réelle définie et initialisée, il est incorrect     d'écrire:         □ x = racine(2/3);         □ x - 1 = racine(x);         □ x = racine(racine(x)*racine(x));         □ x = racine(x * x) - racine(x);  14. Soit la fonction f définie par:     int f(int a)     {         printf("a = \n", %d);         if (a &gt; 0)         {             return 3;         }         return 4;     }  Alors l'expression f(0) prendra la valeur:     □ 3</pre>	□ ses chants □ ses blocs □ ses champs  16. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise: □ int afficher_menu(int char); □ int afficher_menu(); □ void afficher_menu(); □ char afficher_menu(printf("menu")); □ double afficher_menu();  17. Au début de la fonction main() on place le code: char b = 'A'; b = b + 2; printf("%c\n", b); Alors l'affichage sera: □ C □ b □ A □ B  18. Après la déclaration: int mccarthy(int n);, il est	<pre>19. Un bit est :</pre>
□ 0 □ 4	correct d'écrire :  □ x = mccarthy(n);	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
15. Un enregistrement permet de grouper plusieurs valeurs dans :  □ ses cases	□ n = mccarthy(p, q); □ n = mccarthy(); □ int mccarthy(int 2);	□ 0 1 2 0 1 2 □ 0 1 0 1 0 1 0 1 □ 1 2 1 2 3

### Éléments d'informatique – contrôle continue

Prénom: Nom:	
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le type des réels en C est :
     □ char
     □ int.
    □ double
    ☐ real
2. Si le code:
  struct toto s
     int n:
     double x;
  };
  précède la fonction main(), alors on peut écrire en
  début de main() :
     \Box int struct toto_s = {3, -1e10};
    \square int toto.n = 3:
     \square toto_s n, x;
     \square toto_s struct z = {3, 0.5};
    □ struct toto_s toto;
3. Le code suivant :
    int somme = 0;
    int i:
    for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
    printf("%d", somme);
  affichera:
     \square 42
    \Box 6
    \Box 0
     \square 1
```

```
4. Vous utilisez une boucle while quand :
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
     ☐ l'incrément de la variable de boucle n'est pas 1
5. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire:
    \square x = mccarthy(n);
    \square n = mccarthy();
    \square n = mccarthy(p, q);
     \Box int mccarthy(int 2);
6. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 0
     \Box 4
     \square 3
7. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
    \square une fonction appelée avant sa déclaration
     □ une fonction déclarée mais non définie
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une directive préprocesseur #include manquante
```

```
8. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
    Alors l'affichage sera:
      \sqcap C
      □ b
      \Box A
      \square B
 9. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      □ main
      \square init
      □ begin
      \square include
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il ne compile pas
      \square il comporte une boucle infinie
      □ il n'affiche rien
      ☐ il risque d'afficher bonjour à la place de coucou
11. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
      \square (!A | | B)
      \square (A == TRUE) && (B == TRUE)
      \square !(!A \mid | B) == (A \&\& !B)
      □ A && B
```

```
12. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
   affichera:
      □ Majeur
      □ Mineur
     \square rien
      ☐ Mineur
        Majeur
13. Un registre du processeur est :
     \Box une unité de calcul spécialisée de l'ordinateur
      \square un composant qui contient la liste des fichiers du
        système
      □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     \square une gamme de fréquence de fonctionnement du
        processeur
14. L'ordonnancement par tourniquet permet :
      □ d'afficher des ronds colorés à l'écran
      ☐ de ne pas perdre de temps avec la commutation
        de contexte
```

```
□ de doubler la mémoire disponible
                                                                   printf("%c", i);
     □ d'entretenir l'illusion que les processus tournent
                                                                printf("\n");
        en parallèle
                                                               Alors l'affichage sera:
15. Pour compiler un programme prog.c, on utilise la
                                                                 ☐ ABCDEF
    ligne de commande :
                                                                 □ A
     ☐ gcc -Wall prog.c -o prog.exe
                                                                 \square i
     ☐ gcc -Wall prog.exe -o prog.c
                                                                 □ ccccc
     ☐ gcc prog.exe -Wall -o prog.c
                                                           18. Laquelle de ces écritures correspond à la déclaration
     ☐ gcc prog.c -o -Wall prog.exe
                                                               d'une variable de type caractère en langage C?
16. Pour l'extrait de programme suivant :
                                                                 ☐ char "c";
     int i = 0;
                                                                 \Box char c;
     int j = 0;
                                                                 □ char 'c';
     for (i = 0; i < 2; i = i + 1)
     {
                                                                 ☐ int char:
         for (j = 0; j < 3; j = j + 1)
                                                           19. Pour déclarer une variable qui sera utilisée comme va-
                                                               riable de boucle on peut utiliser l'instruction
              printf("%d ", j);
                                                                 \square int loop n;
         }
    }
                                                                 \square loop i;
                                                                 \square int %d;
   qu'est ce qui sera affiché?
                                                                 \square int k;
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2 3
                                                           20. Pour déclarer une fonction exposant qui prend en ar-
                                                               gument un réel x et un entier positif n et renvoie la
     \Box 0 1 2 3 0 1 2
                                                               valeur de x^n on écrit :
     \Box 0 1 2 0 1 2 3
                                                                 \square int exposant(double n, int x);
17. Au début de la fonction main() on place le code :
                                                                 \square void exposant(double x^n);
     char i;
                                                                 \square double exposant(double x, int n);
    for (i = 'A'; i \le 'F'; i = i + 1)
                                                                 \square exposant(double x, int n, int r);
```

## Éléments d'informatique – contrôle continue

 $\Box$  4

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

ar i	reponse fausse. Duree : 20 minutes.
1.	Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
	☐ void saisie_utilisateur(int n);
	☐ saisie_utilisateur(scanf(%d));
	☐ void saisie_utilisateur(char c);
	☐ int saisie_utilisateur();
2.	Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
	☐ #include <studio.h></studio.h>
	☐ #include <stdio.h></stdio.h>
	☐ #appart <stdlib.h></stdlib.h>
	$\square$ #include <studlib.h></studlib.h>
3.	La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
	$\hfill \square$ certaines données de la mémoire de travail
	$\square$ des processus
	$\Box$ les fichiers du disque
	$\Box$ en temps d'accès
4.	Le code suivant :
	<pre>int i; for (i = 0; i &lt; 7; i = i + 2) {</pre>
	<pre>printf("%d ", i); }</pre>
	<pre>printf("\n");</pre>
	affichera:
	$\Box \ 0\ 2\ 4\ 6\ 8$
	$\square \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6$
	$\square \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$
	$\Box \ 0\ 2\ 4\ 6$

```
5. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     □ un point-virgule en trop
     ☐ un point-virgule manquant
     \square une accolade en trop
     \square une accolade manquante
6. Si cette erreur apparaît à la compilation :
   Undefined symbols :"_prinft" ou
   référence indéfinie vers « prinft » que doit-
   on chercher dans le programme?
     □ un caractère interdit en C
     \square une faute de frappe dans un appel de fonction
     ☐ une directive préprocesseur #include manquante
     □ une variable non déclarée
7. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(2/3);
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
     \square x = racine(x * x) - racine(x);
8. Pour l'extrait de programme suivant :
     int produit = 1;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 0
     \square 8
     \Box 16
```

```
9. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \square 3
      \square 7
      \square 8
      □ 111
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il ne compile pas
      □ il risque d'afficher bonjour à la place de coucou
      □ il n'affiche rien
      □ il comporte une boucle infinie
11. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
         return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 7
      \Box 0
      \Box 5
12. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
      \square (!A | | B)
      \square !(!A \mid | B) == (A \&\& !B)
      □ A && B
```

 $\square$  (A == TRUE) && (B == TRUE)

```
13. Soit la fonction f définie par :
                                                                  if (a > 0)
                                                                                                                        1
    int f(int a)
                                                                     return f(a - 1) + 1;
      printf("a = \n", %d);
                                                                  return 4;
      if (a > 0)
                                                                Alors l'expression f(1) prendra la valeur :
        return 3;
                                                                  \Box 5
      return 4;
                                                                  \Box 1
                                                                  \Box 0
    Alors l'expression f(0) prendra la valeur :
                                                                  \Box 4
      \Box 0
                                                            16. Pour déclarer une procédure afficher_menu sans ar-
      \square 3
                                                                gument et qui ne renvoie rien on utilise:
      \Box 4
                                                                  ☐ int afficher_menu(int char);
14. Le bus système sert à :
                                                                  ☐ char afficher_menu(printf("menu"));
      □ transporter les processus du tourniquet au pro-
                                                                  ☐ double afficher_menu();
         cesseur
                                                                  ☐ int afficher_menu();
     ☐ Écrire des données sur le dique dur
                                                                  □ void afficher_menu();
      ☐ Arriver à l'heure en cours
                                                            17. Une variable booléenne est un variable :
     □ Transférer des données et intructions entre pro-
                                                                  \square jamais nulle
                                                                                                                              une fonction appelée avant sa déclaration
         cesseur et mémoire
                                                                  □ réelle positive
                                                                                                                              ☐ une directive préprocesseur #include manquante
15. Soit la fonction f définie par :
                                                                  □ NaN (not a number, qui n'est pas un nombre)
                                                                                                                              \Box un désaccord entre la déclaration et la définition
    int f(int a)
                                                                  \Box à la
quelle une valeur vient d'être affectée
                                                                                                                                 d'une fonction
                                                                  \Box qui est vraie ou fausse
                                                                                                                              □ une fonction déclarée mais non définie
      printf("a = \n", %d);
```

8.	Quelle étape de la compilation vient d'échouer lors- qu'on a un message comme celui-ci :
	Undefined symbols :"_prinft" ou
	référence indéfinie vers « prinft »
	$\Box$ l'analyse des entrées clavier
	$\Box$ l'édition de liens
	$\Box$ l'analyse harmonique
	$\Box$ l'analyse sémantique
9.	Un fichier source est :
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\Box$ un document qui doit être protégé
	$\Box$ un document illisible pour les humains
	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\Box$ un document de référence du système
0.	Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max , que doit-on chercher dans le programme?

## Éléments d'informatique – contrôle continue

rénom : ° INE :	Nom:	
affichera:		
□ Mineur Majeur		
□ rien		

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

- 1. Une variable booléenne est un variable : □ NaN (not a number, qui n'est pas un nombre) □ qui est vraie ou fausse ☐ réelle positive □ jamais nulle □ à laquelle une valeur vient d'être affectée 2. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\Box$  0  $\square$  3  $\Box$  4 3. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
  - □ alphabétique  $\square$  un ordre quelconque  $\square$  dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main
- 4. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :

```
\square printf("x=%x et y=%y\n");
\square printf("x=%d et y=%d\n",x,y);
\square printf("x=%d et y=%d\n,x,y");
\square printf("x=%d et y=%d\n",x y);
```

```
5. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
     {
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
    \square 5
    \Box 7
    \Box 0
6. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
        }
   }
   qu'est ce qui sera affiché?
    \Box 0 1 0 1 0 1 0 1
    \Box 0 0 0 1 1 1
    \Box 0 1 2 0 1 2
    \Box 1 2 1 2 3
7. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
```

```
□ Mineur
      □ Majeur
 8. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     ☐ struct date_s afficher_date(struct date_s d);
     □ void afficher_date(struct date_s d);
     ☐ int afficher_date(date_s d);
     □ void afficher_date(date_s d);
 9. Le type des réels en C est :
     □ char
     \square real
     □ int
     □ double
10. Une de ces manière de composer les blocs de pro-
    grammes ne fait pas partie des opérations de la pro-
   grammation structurée :
      □ répéter un bloc tant qu'une condition est vérifée
     □ retourner un bloc
     \square mettre les blocs en séquence les uns à la suite des
      □ sélectionner entre deux blocs à l'aide d'une condi-
        tion
11. Un fichier source est:
      ☐ un document illisible pour les humains
      □ un document de référence du système
     □ un document qui doit être protégé
     □ un fichier texte qui sera traduit en instructions
        processeur
     \square un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
```

12. Un bit est:	15. Pour déclarer une fonction factorielle qui prend en	19. Soit un programme contenant les lignes suivantes :
□ un battement d'horloge processeur	argument un entier et renvoie sa factorielle on écrit :	
☐ l'instruction qui met fin à un programme	$\square$ int factorielle(double n);	int i = 0;
☐ la longueur d'un mot mémoire	☐ int factorielle();	int $j = 0$ ; for $(i = 0; i < 3; i = i + 1)$
$\Box$ un chiffre binaire (0 ou 1)	☐ struct int factorielle(int n);	{
13. Pour l'extrait de programme suivant :	☐ int factorielle(int x);	for $(j = 0; j < 5; j = j + 1)$
<ul> <li>13. Pour l'extrait de programme suivant :         int i = 0;         int j = 0;         for (i = 0; i &lt; 2; i = i + 1)         {              for (j = 0; j &lt; 3; j = j + 1)</li></ul>	☐ int factorielle(int x);  16. Un programme en langage C doit comporter une et une seule définition de la fonction :  ☐ include ☐ begin ☐ main ☐ init  17. L'écriture 111 en binaire correspond au nombre naturel :  ☐ 111 ☐ 7 ☐ 8 ☐ 3  18. Avant de faire appel à une fonction il est nécessaire de :	for (j = 0; j < 5; j = j + 1) { }  printf("j = %d\n", j);  qu'est ce qui sera affiché par ce printf?
valeur de $x^n$ on écrit :	☐ l'avoir déclarée et définie	$\square$ analyse sémantique
$\square$ double exposant(double x, int n);	$\hfill\Box$ avoir défini une constante symbolique de la taille	$\square$ analyse harmonique
$\Box$ int exposant(double n, int x);	de cette fonction	$\square$ analyse lexicale
☐ void exposant(double x^n);	☐ l'avoir définie	v
$\square$ exposant(double x, int n, int r);	☐ l'avoir déclarée	$\square$ analyse syntaxique

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution jusqu'à la ligne 15 du programme C:
```

```
10 int main() {
11 int x = 5;
12 int y;
13
14 y = x;
15
16 ...
17 }
```

- $\square$  la variable x vaut 0
- □ le programme affiche "Faux"
- $\square$  la variable x vaut 5 et la variable y vaut 0
- $\Box$ la variable y vaut 5
- 2. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
\square n = factorielle(p, q);
```

- ☐ printf("%d", factorielle(n));
- $\square$  n = factorielle();
- ☐ int factorielle(int 2);
- 3. L'écriture  $\underline{101}$  en binaire correspond au nombre naturel :
  - $\square$  101
  - $\Box$  5
  - $\Box$  4
  - $\square$  3
- 4. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :

```
\square int pgcd(int x, int x);
```

- $\square$  void pgcd(int x, int y);
- $\Box$  int pgcd(int x, y);
- $\square$  int pgcd(int y, int x);

5. Le code suivant :

```
int age = 18;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");
}</pre>
```

affichera:

- $\square$  rien
- ☐ Mineur
- ☐ Mineur Majeur
- $\square$  Majeur
- 6. Soit la fonction **f** définie par :

```
int f(int a)
{
  printf("a = \n", %d);
  if (a > 0)
  {
    return 3;
  }
  return 4;
```

Alors l'expression  ${\tt f(0)}$  prendra la valeur :

- $\square$  3
- $\Box$  4
- $\Box$  0
- 7. Un bit est:
  - $\square$  un chiffre binaire (0 ou 1)
  - $\Box$  un battement d'horloge processeur
  - $\Box$  la longueur d'un mot mémoire
  - $\square$  l'instruction qui met fin à un programme

 $8.\ \, {\rm Soit}$  un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 3; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);</pre>
```

qu'est ce qui sera affiché par ce printf?

- $\Box$  j = 5
- $\Box$  j = %d
- $\Box$  j = 4  $\Box$  i = 0
- 9. Le code suivant :

```
int i;
for (i = 4; i > 0; i = i - 1)
{
    printf("%d ", i);
}
printf("\n");
affichera:
```

- $\square$  0 1 2 3
- $\Box \ 0\ 1\ 2\ 3\ 4$
- $\Box$  4 3 2 1 0
- $\square$  4 3 2 1
- 10. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
   printf("coucou\n");
}
```

- $\Box\,$ il ne compile pas
- $\square$  il comporte une boucle infinie
- $\square$  il n'affiche rien
- $\Box\,$ il risque d'afficher bonjour à la place de coucou

<ul><li>11. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :</li><li>□ analyse lexicale</li></ul>	16. On souhaite faire une boucle de contrôle de saisie : tant que l'entier ${\tt n}$ n'appartient pas à l'intervalle $[ab]$ , on recommence la saisie de ${\tt n}$ . Soit le programme suivant :	18. Pour déclarer une fonction <b>exposant</b> qui prend en argument un réel $x$ et un entier positif $n$ et renvoie la valeur de $x^n$ on écrit :
$\square$ analyse harmonique	int a = 0;	$\square$ exposant(double x, int n, int r);
$\Box$ analyse sémantique	int b = 20;	$\square$ void exposant(double x^n);
$\square$ analyse syntaxique	<pre>int n; scanf("%d", &amp;n);</pre>	$\square$ double exposant(double x, int n);
12. Un enregistrement permet de grouper plusieurs valeurs	While(cond)	☐ int exposant(double n, int x);
dans:	{	-
□ ses cases	scanf("%d", &n);	19. Soit la fonction f définie par :
□ ses blocs	}	int f(int a)
$\square$ ses champs	Quelle est la condition cond :	{ printf("a = \n", %d);
□ ses chants	□ (n<=a) && (n<=b)	if $(a > 0)$
13. Dans la commande gcc, l'option -Wall signifie :	□ (a<=n) && (n<=b)	{
$\hfill\Box$ que l'on veut voir tous les avertissements	$\Box$ (a <n) (n=""   ="">b)</n)>	return f(a - 1) + 1;
$\Box$ qu'il faut indenter le fichier source	☐ a<=n<=b	}
□ qu'il faut lancer un déboggueur	17. Soit un programme contenant les lignes suivantes :	return 4; }
□ qu'on veut changer alétoirement de fond d'écran	int i = 0;	Alors l'expression f(1) prendra la valeur :
14. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre	int j = 0;	□ 5
dans lequel vous devez maintenant définir ces fonctions	for (i = 0; i < 2; i = i + 1)	□ 1
est l'ordre :	{     for (j = 0; j < 3; j = j + 1)	
$\Box$ dans lequel vous avez déclaré ces fonction	101 (j = 0, j < 3, j = j + 1) {	
$\Box$ un ordre quel conque	printf("%d ", i);	$\Box$ 4
$\Box$ dans lequel ces fonctions sont appelées dans le	}	20. Si cette erreur apparaît à la compilation :
main	}	erreur: conflicting types for 'max', que doit-
$\square$ alphabétique	m 1 10	on chercher dans le programme?
15. Le langage C est un langage	qu'est ce qui sera affiché?	$\Box$ une fonction déclarée mais non définie
$\square$ interprété	$\Box$ 0 1 0 1 0 1 0 1	$\hfill \square$ une directive préprocesseur $\#\mbox{include}$ man quante
□ lu, écrit, parlé	□ 1 2 1 2 3	$\hfill\Box$ une fonction appelée avant sa déclaration
□ compilé	$\Box$ 0 1 2 0 1 2	$\hfill\Box$ un désaccord entre la déclaration et la définition
□ composé	$\Box$ 0 0 0 1 1 1	d'une fonction

# Éléments d'informat

affichera:

 $\square$  8 2

 $\square$  8 6 4 2

 $\Box 02468$ 

 $\Box$  8 6 4 2 0

ique – contrôle continue	$\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N}^{\circ} \text{ INE}: & & . \end{array}$
6. Soit la fonction f définie par :	9. Après exécution jusqu'à la ligne 14 du programme C :
<pre>int f(int a) {    printf("a = \n", %d);    if (a &gt; 0)    {       return 3;    }    return 4; }</pre>	<pre>10 int main() { 11    int x = 5; 12 13    printf(" x = %d\n", 2); 14 15    16 }</pre>
Alors l'expression f(0) prendra la valeur :	□ le terminal affiche "Faux"
□ 3 □ 4 □ 0	□ le terminal affiche x = 5 □ le terminal affiche 5
7. Le code suivant :	10. Sous unix (ou linux), la commande 1s permet de :
<pre>int somme = 0; int i;</pre>	□ afficher la liste de fichiers contenus dans un répertoire
for (i = 1; i < 4; i = i + 1) {	□ voir des clips musicaux
<pre>somme = somme + i; }</pre>	$\square$ afficher le contenu d'un fichier texte
<pre>printf("%d", somme);</pre>	$\Box$ compiler un programme
affichera: □ 6 □ 1	<ul> <li>11. Un fichier source est :</li> <li>□ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur</li> <li>□ un document qui doit être protégé</li> </ul>
$\square$ 0 $\square$ 42	□ un document illisible pour les humains
8. Le code suivant : int i;	□ un fichier texte qui sera traduit en instructions processeur
for $(i = 8; i > 0; i = i - 2)$	□ un document de référence du système
<pre>{      printf("%d ", i); } printf("\n");</pre>	12. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :

1. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?  $\Box$  un désaccord entre la déclaration et la définition d'une fonction  $\Box$  une directive préprocesseur #include manquante  $\square$  une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration 2. Un enregistrement permet de grouper plusieurs valeurs dans:  $\square$  ses cases  $\square$  ses chants  $\square$  ses champs  $\square$  ses blocs 3. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n");  $\square$  il ne compile pas  $\square$  il comporte une boucle infinie □ il risque d'afficher bonjour à la place de coucou □ il n'affiche rien 4. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <stdio.h> ☐ #include <studio.h> ☐ #include <studlib.h> ☐ #appart <stdlib.h> 5. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme?  $\square$  un caractère interdit en C  $\square$  une directive préprocesseur #include manquante  $\square$  une faute de frappe dans un appel de fonction  $\square$  une variable non déclarée

Barème : 1 points par réponse juste (unique) ; -0.5 points

par réponse fausse. Durée : 20 minutes.

```
\square int pgcd(int x, int x);
\Box int pgcd(int x, y);
\square void pgcd(int x, int y);
\square int pgcd(int y, int x);
```

13. Un registre du processeur est :
☐ une case mémoire interne au processeur qui sera manipulée directement lors des calculs
$\hfill \square$ une unité de calcul spécialisée de l'ordinateur
$\Box$ une gamme de fréquence de fonctionnement du processeur
$\hfill\Box$ un composant qui contient la liste des fichiers du système
14. Vous utilisez une boucle while quand :
$\Box$ vous n'avez pas déclaré de fonction
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
□ vous avez déjà fait un <b>for</b> dans le même pro- gramme principal
$\Box$ l'incrément de la variable de boucle n'est pas $1$
15. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :
$\square$ tour à tour, un petit peu à chaque fois
☐ chacun son tour, après que le processus précédent a terminé
$\Box$ en parallèle, chacun dans un registre
$\Box$ tous ensemble

```
16. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
             printf("%d ", i);
         }
    }
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 1 0 1 0 1
     \Box 1 2 3 1 2
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2
17. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise :
     □ void afficher_menu();
     ☐ double afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu(int char);
     ☐ int afficher_menu();
18. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier \mathbf{n} n'appartient pas à l'intervalle [a..b], on
```

recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
     }
    Quelle est la condition cond :
      \Box (a<n) || (n>b)
      \square (n<=a) && (n<=b)
      □ a<=n<=b
      \square (a<=n) && (n<=b)
19. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     □ char 'c';
      ☐ int char;
      \Box char c;
      ☐ char "c";
20. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \square n = mccarthy();
     \square x = mccarthy(n);
     \square n = mccarthy(p, q);
```

 $\Box$  int mccarthy(int 2);

т		-1
	acence	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :  $\square$  n = mccarthy();  $\square$  x = mccarthy(n);  $\square$  int mccarthy(int 2);  $\square$  n = mccarthy(p, q); 2. Un enregistrement permet de grouper plusieurs valeurs dans: □ ses cases  $\square$  ses champs □ ses chants  $\square$  ses blocs 3. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :  $\square$  tous ensemble □ chacun son tour, après que le processus précédent a terminé □ en parallèle, chacun dans un registre  $\square$  tour à tour, un petit peu à chaque fois 4. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : ☐ int factorielle(double n): □ struct int factorielle(int n); ☐ int factorielle();  $\square$  int factorielle(int x); 5. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?  $\square$  (A == TRUE) && (B == TRUE) □ A && B  $\square$  (!A || B)

 $\square !(!A | | B) == (A \&\& !B)$ 

```
6. Après exécution jusqu'à la ligne 14 du programme C:
10
     int main() {
11
          int x = 5;
 12
 13
          printf(" x = %d\n", 2);
 14
 15
 16
     }
    □ le terminal affiche 5
    □ le terminal affiche "Faux"
    \square le terminal affiche x = 5
    \square le terminal affiche x = 2
7. Le langage C est un langage
    □ compilé
    □ lu, écrit, parlé
    □ interprété
     □ composé
8. Le code suivant:
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
    □ rien
    □ Mineur
    □ Majeur
    □ Mineur
       Majeur
9. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %d\n", n);
    □ scanf("%d", &n);
    □ un débogueur
     □ printf("Valeur de n ? %g\n", n);
```

```
10. Au début de la fonction main() on place le code :
     char b = 'A';
    b = b + 2:
     printf("%c\n", b);
   Alors l'affichage sera:
     \Box C
     □ A
     □ b
     □В
11. Le bus système sert à :
     ☐ Écrire des données sur le dique dur
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     ☐ Arriver à l'heure en cours
     □ transporter les processus du tourniquet au pro-
        cesseur
12. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
   , que doit-on chercher dans le programme?
     □ une fonction déclarée mais non définie
     ☐ une directive préprocesseur #include manquante
     \square une fonction appelée avant sa déclaration
     □ un désaccord entre la déclaration et la définition
        d'une fonction
13. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
   Undefined symbols :"_prinft" ou
   référence indéfinie vers « prinft »
     □ l'analyse des entrées clavier
     □ l'analyse sémantique
     □ l'édition de liens
     \square l'analyse harmonique
```

```
14. Le code suivant :
                                                                                                                               \Box voir des clips musicaux
                                                                  \square toto_s n, x;
                                                                  \Box int struct toto_s = {3, -1e10};
                                                                                                                               □ compiler un programme
     int somme = 0;
     int i;
                                                            16. Soit le programme principal suivant :
                                                                                                                               \Box afficher le contenu d'un fichier texte
     for (i = 1; i < 4; i = i + 1)
                                                                int main()
                                                                                                                         18. Un programme en langage C doit comporter une et une
                                                                {
                                                                                                                             seule définition de la fonction :
       somme = somme + i;
                                                                 int a = 3;
     }
                                                                                                                               □ begin
                                                                 int b = 5;
    printf("%d", somme);
                                                                 printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                               \square main
                                                                 return EXIT_SUCCESS;
                                                                                                                               \square init
   affichera:
     \Box 6
                                                                                                                               \square include
                                                                appelant la fonction f ainsi définie :
     \square 42
                                                                                                                         19. L'écriture 111 en binaire correspond au nombre natu-
                                                                int f(int a, int b)
     \Box 0
                                                                                                                             rel:
     \Box 1
                                                                                                                               \Box 7
                                                                   a = a + b;
15. Si le code:
                                                                   return a;
                                                                                                                               □ 111
   struct toto_s
                                                                                                                               \square 3
                                                                L'affichage dans le main est le suivant :
                                                                                                                               \square 8
      int n;
                                                                  \Box f(3,5)=8, a=3, b=5
      double x;
                                                                                                                         20. Laquelle de ces écritures correspond à la déclaration
                                                                  \Box f(a,b)=13, a=8, b=5
   };
                                                                                                                             d'une variable de type caractère en langage C?
                                                                  \Box f(a,b)=8, a=8, b=5
   précède la fonction main(), alors on peut écrire en
                                                                                                                               ☐ char "c";
                                                                  \Box f(a,b)=8, a=3, b=5
   début de main():
                                                                                                                               \Box char c;
     \square int toto.n = 3;
                                                            17. Sous unix (ou linux), la commande 1s permet de :
                                                                                                                               ☐ int char:
     \Box toto_s struct z = {3, 0.5};
                                                                  \square afficher la liste de fichiers contenus dans un
                                                                                                                               □ char 'c';
     □ struct toto_s toto;
                                                                     répertoire
```

т	•	-1
- 1	acence	- 1

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

	eponse lausse. Durce: 20 minutes.
1.	Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :
	☐ int pgcd(int y, int x);
	☐ int pgcd(int x, y);
	□ void pgcd(int x, int y);
	☐ int pgcd(int x, int x);
2	Soit la fonction f définie par :
۵.	-
	<pre>int f(int a) {</pre>
	printf("a = \n", %d);
	if (a > 0)
	{ 
	return 3;
	return 4;
	}
	Alors l'expression $f(0)$ prendra la valeur :
	$\Box$ 0
	$\Box$ 4
	$\square$ 3
3.	Si cette erreur apparaît à la compilation :
	Undefined symbols :"_prinft" ou
	référence indéfinie vers « prinft » que doiton chercher dans le programme?
	$\Box$ une variable non déclarée
	$\square$ un caractère interdit en C
	$\Box$ une faute de frappe dans un appel de fonction
	$\Box$ une directive préprocesseur $\#\mbox{include}$ manquante
4.	Un enregistrement permet de grouper plusieurs valeurs dans :
	$\square$ ses chants
	□ ses blocs
	$\square$ ses champs
	□ ses cases

```
5. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     ☐ int factorielle(int 2);
     \square n = factorielle():
     \square n = factorielle(p, q);
    ☐ printf("%d", factorielle(n));
6. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     □ main
     \square include
     \square init
     □ begin
7. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     □ un point-virgule en trop
    \square une accolade manquante
     \square une accolade en trop
     ☐ un point-virgule manquant
8. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     □ Majeur
     ☐ Mineur
     □ Mineur
        Majeur
     \square rien
9. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
```

```
\square il ne compile pas
      □ il n'affiche rien
      \square il comporte une boucle infinie
      \Box il risque d'afficher bonjour à la place de coucou
10. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse lexicale
      \square analyse harmonique
      □ analyse syntaxique
      □ analyse sémantique
11. Soit la fonction f définie par :
    int f(int a)
    ₹
      printf("a = \n", %d);
      if (a > 0)
         return f(a - 1) + 1;
      return 4:
    Alors l'expression f(1) prendra la valeur :
      \Box 1
      \Box 0
      \square 5
      \Box 4
12. Sous unix (ou linux), la commande 1s permet de :
      □ compiler un programme
      □ afficher la liste de fichiers contenus dans un
         répertoire
      □ voir des clips musicaux
      \square afficher le contenu d'un fichier texte
13. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
      \square (!A || B)
      □ A && B
      \square !(!A \mid | B) == (A \&\& !B)
```

 $\square$  (A == TRUE) && (B == TRUE)

```
14. Après la déclaration : int mccarthy(int n);, il est
                                                                  \square int exposant(double n, int x);
                                                                                                                                \square a{n, x} == b{n, x}
    correct d'écrire :
                                                                  \square double exposant(double x, int n);
                                                                                                                               \Box a = b
      \square int mccarthy(int 2);
                                                            17. Le code suivant :
                                                                                                                         19. Pour l'extrait de programme suivant :
     \square n = mccarthy(p, q);
                                                                  int i:
                                                                                                                               int i = 0;
     \square n = mccarthy();
                                                                  for (i = 1; i < 5; i = i + 1)
                                                                                                                               int j = 0;
      \square x = mccarthy(n);
                                                                                                                               for (i = 0; i < 3; i = i + 1)
                                                                      printf("%d ", i);
15. Si le code :
                                                                                                                                    for (j = 0; j < 2; j = j + 1)
    struct toto_s
                                                                 printf("\n");
    {
                                                                                                                                        printf("%d ", i);
                                                                 affichera:
      int n;
                                                                                                                                   }
                                                                  \square 1 2 3 4
      double x;
                                                                                                                               }
                                                                  \Box 01234
                                                                                                                               printf("\n");
    précède la fonction main(), alors on peut écrire en
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
                                                                                                                             qu'est ce qui sera affiché?
    début de main() :
                                                                   \square 4 3 2 1
                                                                                                                               \Box 0 1 2 0 1 2
      □ struct toto_s toto;
                                                             18. Si a et b sont deux variables de type:
                                                                                                                               \Box 1 2 3 1 2
      \Box int struct toto_s = {3, -1e10};
                                                                 struct toto_s
                                                                                                                               \Box 0 1 0 1 0 1
      \square toto_s n, x;
      \square int toto.n = 3;
                                                                                                                               \Box 0 0 1 1 2 2
                                                                   int n;
     \Box toto_s struct z = {3, 0.5};
                                                                   double x;
                                                                                                                         20. Sur unix (ou linux), la commande mkdir permet de :
                                                                 };
16. Pour déclarer une fonction exposant qui prend en ar-
                                                                                                                                □ créer un répertoire
                                                                Alors pour tester l'égalité de a et de b on utilise la
    gument un réel x et un entier positif n et renvoie la
                                                                                                                               \square changer de répertoire courant
    valeur de x^n on écrit :
                                                                 condition:
                                                                                                                                \square créer un fichier texte
     \square void exposant(double x^n);
                                                                  □ a == b
                                                                                                                                □ ouvrir un fichier texte
     \square exposant(double x, int n, int r);
                                                                   \square (a.n == b.n) && (a.x == b.x)
```

т.	-1
Licence	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de  $x^n$  on écrit :  $\square$  double exposant(double x, int n);  $\square$  int exposant(double n, int x);  $\square$  void exposant(double x^n);  $\square$  exposant(double x, int n, int r); 2. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :  $\square$  n = mccarthy();  $\square$  x = mccarthy(n); ☐ int mccarthy(int 2);  $\square$  n = mccarthy(p, q); 3. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme? □ une directive préprocesseur #include manquante □ un caractère interdit en C □ une variable non déclarée  $\square$  une faute de frappe dans un appel de fonction 4. Un enregistrement permet de grouper plusieurs valeurs dans:  $\square$  ses chants  $\square$  ses blocs  $\square$  ses champs □ ses cases 5. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?  $\square$  un désaccord entre la déclaration et la définition d'une fonction  $\square$  une fonction appelée avant sa déclaration □ une fonction déclarée mais non définie

□ une directive préprocesseur #include manquante

```
6. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
11
           int x = 5;
12
           int y;
13
14
           v = x;
15
16
17
     }
    \square la variable x vaut 5 et la variable y vaut 0
    \square la variable x vaut 0
    ☐ le programme affiche "Faux"
    \square la variable y vaut 5
7. Vous avez déclaré préalablement un ensemble de fonc-
  tions utilisées par votre programme principal. L'ordre
  dans lequel vous devez maintenant définir ces fonctions
  est l'ordre:
    \square dans lequel ces fonctions sont appelées dans le
        main
    □ dans lequel vous avez déclaré ces fonction
    □ alphabétique
     □ un ordre quelconque
8. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
10
11
           int x = 5;
12
           int y = 3;
13
14
           x = y;
15
16
17
    \square la variable x vaut 5 et la variable y vaut 3
    ☐ le programme affiche "Faux"
    \square la variable x vaut 3
     \square la variable y vaut 5
```

```
9. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     □ void afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu(int char);
10. Le langage C est un langage
     □ composé
     □ compilé
     □ lu, écrit, parlé
     □ interprété
11. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \Box int pgcd(int x, int x);
     \Box int pgcd(int x, y);
     \square void pgcd(int x, int y);
     \square int pgcd(int y, int x);
12. Le code suivant :
    int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
         printf("Majeur\n");
   affichera:
     □ rien
     □ Mineur
        Majeur
     □ Majeur
     □ Mineur
```

13.	Si carre est une fonction prenant en entrée un en-
	tier et renvoyant le carré de cet entier, et que n est
	une variable entière définie et initialisée, il est correct d'écrire :
	☐ int carre(2);
	☐ int n = carre();
	$\square$ n = carre(n);
	□ n = carre(int n);
14.	Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
	☐ #define taille = 3
	$\square$ #define taille = N
	□ #define N = 3
	$\square$ #define N 3
15.	Le type des réels en C est :
	$\square$ int
	$\square$ real
	$\square$ double
	□ char
16.	Soit la fonction g définie par :
	<pre>int g(int a) {</pre>
	<pre>printf("a = \n", %d); if (1 &gt; 0) {</pre>

```
return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
     \Box 7
     \Box 5
     \Box 0
17. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
              printf("%d ", i);
         }
    printf("\n");
    qu'est ce qui sera affiché?
     \Box 1 2 3 1 2
     \Box 0 1 0 1 0 1
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2
```

```
18. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
19. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
   Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 4
     \Box 0
20. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     \square ouvrir un fichier texte
     □ changer de répertoire courant
     \square créer un fichier texte
```

Prénom : Nom : No INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  $\mathbf{n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  $\mathbf{n}$ . Soit le programme suivant :

```
recommence la saisie de n. Soit le programme suivant :
    int a = 0;
    int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
      scanf("%d", &n);
  Quelle est la condition cond:
    □ a<=n<=b
    \square (a<=n) && (n<=b)
    \square (n<=a) && (n<=b)
    \square (a<n) || (n>b)
2. Soit le programme principal suivant :
  int main()
  ₹
    int a = 3;
    int b = 5:
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  }
  appelant la fonction f ainsi définie :
  int f(int a, int b)
  {
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
```

```
3. Après exécution jusqu'à la ligne 15 du programme C:
 10
      int main() {
11
           int x = 5;
12
           int y = 3;
13
14
           x = y;
15
 16
           . . .
     }
 17
     \square la variable x vaut 5 et la variable y vaut 3
    \square la variable x vaut 3
    ☐ le programme affiche "Faux"
    \square la variable y vaut 5
4. Si le code :
   struct toto_s
     int n;
     double x;
  }:
  précède la fonction main(), alors on peut écrire en
  début de main() :
    \square toto_s n, x;
    □ struct toto_s toto;
    \square int struct toto_s = {3, -1e10};
    \square toto_s struct z = {3, 0.5};
    \square int toto.n = 3;
5. Le langage C est un langage
    □ compilé
    □ lu, écrit, parlé
    □ composé
     □ interprété
6. Le type des réels en C est :
     □ real
    □ char
     □ int
```

☐ double

```
7. Soit un programme contenant les lignes suivantes :
     int i = 0:
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
          for (j = 0; j < 5; j = j + 1)
     }
     printf("j = %d\n", j);
     }
    qu'est ce qui sera affiché?
     \Box j = 0
     \Box j = 4
     \Box j = 5
      \Box j = %d
 8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \square 7
     \square 8
      \square 3
 9. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n,x,y");
     \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n",x y);
10. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
```

{

```
14. Pour déclarer une fonction saisie_utilisateur qui
                                                                                                                        18. Après exécution jusqu'à la ligne 15 du programme C:
        return 3:
      }
                                                                demande à l'utilisateur d'entrer un entier au clavier et
                                                                                                                                int main() {
                                                                                                                          10
                                                                renvoie cet entier on écrit :
      return 4;
                                                                                                                          11
                                                                                                                                    int x = 5;
    }
                                                                  □ void saisie_utilisateur(int n);
                                                                                                                          12
                                                                                                                                    int y;
                                                                  ☐ saisie_utilisateur(scanf(%d));
    Alors l'expression f(0) prendra la valeur :
                                                                                                                          13
                                                                  ☐ int saisie_utilisateur();
                                                                                                                          14
      \Box 0
                                                                                                                                    y = x;
                                                                  □ void saisie_utilisateur(char c);
                                                                                                                          15
      \square 3
                                                                                                                          16
                                                            15. Le code suivant :
      \Box 4
                                                                                                                          17
                                                                                                                               }
                                                                 int age = 15;
11. Le code suivant :
                                                                 if (age < 18)
                                                                                                                              □ le programme affiche "Faux"
                                                                 {
     int age = 20;
                                                                                                                              \square la variable x vaut 5 et la variable y vaut 0
                                                                      printf("Mineur\n");
     if (age < 18)
                                                                 }
                                                                                                                              \square la variable y vaut 5
                                                                 else
         printf("Mineur\n");
                                                                                                                              \Box la variable x vaut 0
                                                                 ₹
                                                                      printf("Majeur\n");
     printf("Majeur\n");
                                                                                                                        19. Pour l'extrait de programme suivant :
                                                                 }
    affichera:
                                                                                                                              int somme = 0;
                                                                affichera:
                                                                                                                              int serie[4] = \{2, 4, 10, 4\};
      □ Majeur
                                                                                                                              for (i = 0; i < 4; i = i + 1)
                                                                  □ Mineur
      □ rien
                                                                     Majeur
      □ Mineur
                                                                                                                                 somme = somme + serie[i];
                                                                  ☐ Mineur
      □ Mineur
                                                                  □ Majeur
                                                                                                                              printf("somme = %d",somme);
         Majeur
                                                                  □ rien
12. Pour déclarer une procédure afficher_date qui prend
                                                            16. Laquelle des analyses suivantes ne fait pas partie des
                                                                                                                            La valeur de somme affichée est :
    en argument un struct date_s et affiche le contenu
                                                                étapes de la compilation :
                                                                                                                              \Box 6
    du struct, on écrit:
                                                                  □ analyse sémantique
                                                                                                                              \square 20
                                                                  \square analyse harmonique
      □ void afficher_date(date_s d);
                                                                                                                              \Box 16
                                                                  \square analyse syntaxique
      □ void afficher_date(struct date_s d);
                                                                  \square analyse lexicale
                                                                                                                              \square 3
      ☐ int afficher_date(date_s d);
     □ struct date_s afficher_date(struct date_s d); Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                        20. Vous utilisez une boucle while quand :
13. Pour déclarer une variable qui sera utilisée comme va-
                                                                dans lequel vous devez maintenant définir ces fonctions
                                                                                                                              □ vous ne connaissez pas le nombre d'itérations de
    riable de boucle on peut utiliser l'instruction
                                                                est l'ordre :
                                                                                                                                 la boucle à l'avance
                                                                  \square un ordre quelconque
      \square int loop n;
                                                                                                                              □ vous avez déjà fait un for dans le même pro-
                                                                  □ alphabétique
                                                                                                                                 gramme principal
      \square int k;
                                                                  \square dans lequel ces fonctions sont appelées dans le
                                                                                                                              ☐ l'incrément de la variable de boucle n'est pas 1
      ☐ int %d:
                                                                                                                              □ vous n'avez pas déclaré de fonction
      \square loop i;
                                                                  \square dans lequel vous avez déclaré ces fonction
```

т		-1
	acence	

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes. 1. Le code suivant : int age = 15; if (age < 18) printf("Mineur\n"); } else { printf("Majeur\n"); affichera: □ Mineur □ rien ☐ Mineur Majeur □ Majeur 2. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme? □ une fonction déclarée mais non définie □ un désaccord entre la déclaration et la définition d'une fonction ☐ une directive préprocesseur #include manquante  $\square$  une fonction appelée avant sa déclaration 3. Le bus système sert à : □ transporter les processus du tourniquet au processeur ☐ Arriver à l'heure en cours ☐ Écrire des données sur le dique dur □ Transférer des données et intructions entre processeur et mémoire 4. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée: □ répéter un bloc tant qu'une condition est vérifée

```
☐ mettre les blocs en séquence les uns à la suite des
        autres
     □ sélectionner entre deux blocs à l'aide d'une condi-
        tion
     □ retourner un bloc
5. Le code suivant:
    int i:
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
     \square 4 3 2 1
    \Box \ 4\ 3\ 2\ 1\ 0
     \square 1 2 3 4
     \Box 01234
6. Vous utilisez une boucle while quand :
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
7. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
    \square n = carre(n);
     \square n = carre(int n);
     \square int n = carre();
     \square int carre(2);
8. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
```

☐ gcc -Wall prog.exe -o prog.c

```
9. Un fichier source est:
     □ un document de référence du système
     \square un fichier texte qui sera traduit en instructions
        processeur
     □ un document illisible pour les humains
     □ un document qui doit être protégé
      \square un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
10. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
    Quelle est la condition cond :
     □ (n<=a) && (n<=b)
      \square (a<n) || (n>b)
      ☐ (a<=n) && (n<=b)
     □ a<=n<=b
11. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
              printf("%d ", i);
         }
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 0 0 1 1 1
     \Box 1 2 1 2 3
```

 $\Box$  0 1 0 1 0 1 0 1

12. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :	15. Quelle étape de la compilation vient d'échouer lors- qu'on a un message comme celui-ci :	$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction
$\square$ x = mccarthy(n);	Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »	$\hfill\Box$ une fonction déclarée mais non définie
<pre>□ int mccarthy(int 2); □ n = mccarthy(p, q); □ n = mccarthy();</pre>	☐ l'analyse harmonique ☐ l'analyse des entrées clavier	<ul> <li>□ une fonction appelée avant sa déclaration</li> <li>□ une directive préprocesseur #include manquante</li> </ul>
13. Quel est le problème d'un programme comportant les lignes suivantes?	☐ l'analyse sémantique☐ l'édition de liens	19. Un programme en langage C doit comporter une et une seule définition de la fonction :
while (1)	16. Le type des réels en C est :	$\square$ main
<pre>{    printf("coucou\n");</pre>	□ char	$\Box$ begin
printi("coucou\n"); }	$\square$ real	$\Box$ include
☐ il n'affiche rien ☐ il ne compile pas ☐ il comporte une boucle infinie ☐ il risque d'afficher bonjour à la place de coucou  14. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <stdio.h> ☐ #appart <stdlib.h> ☐ #include <studlib.h> ☐ #include <studlib.h></studlib.h></studlib.h></stdlib.h></stdio.h>	☐ int ☐ double  17. Un enregistrement permet de grouper plusieurs valeurs dans : ☐ ses chants ☐ ses champs ☐ ses cases ☐ ses blocs  18. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?	<ul> <li>□ init</li> <li>20. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :</li> <li>□ dans lequel vous avez déclaré ces fonction</li> <li>□ un ordre quelconque</li> <li>□ alphabétique</li> <li>□ dans lequel ces fonctions sont appelées dans le main</li> </ul>

 $\Box$  0

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes. 1. Au début de la fonction main() on place le code : char i; for  $(i = 'A'; i \le 'F'; i = i + 1)$ printf("%c", i); printf("\n"); Alors l'affichage sera: □ ccccc □ A  $\Box$  i ☐ ABCDEF 2. Sous unix (ou linux), la commande 1s permet de :  $\square$  afficher le contenu d'un fichier texte □ voir des clips musicaux  $\square$  afficher la liste de fichiers contenus dans un répertoire □ compiler un programme 3. Le langage C est un langage □ compilé □ lu, écrit, parlé □ interprété □ composé 4. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\Box$  4  $\square$  3

```
5. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studlib.h>
6. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 0; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
    \Box j = %d
     \Box j = 4
     \Box j = 0
     \Box j = 5
7. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera:
     \square B
     □b
     □ C
     \square A
8. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
    □ scanf("%d", &n);
     □ un débogueur
     ☐ printf("Valeur de n ? %g\n", n);
     ☐ printf("Valeur de n ? %d\n", n);
```

INE.
9. Une variable booléenne est un variable :
$\Box$ jamais nulle
$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)
$\square$ réelle positive
$\Box$ qui est vraie ou fausse
$\Box$ à laquelle une valeur vient d'être affectée
10. Le type des réels en C est :
☐ double
$\square$ real
□ int
□ char
11. Soit la fonction g définie par :
<pre>int g(int a) {    printf("a = \n", %d);    if (1 &gt; 0)    {       return 5;    }    return 7; }</pre>
Alors l'expression $g(0)$ prendra la valeur :
□ 7
$\Box$ 5
12. Un enregistrement permet de grouper plusieurs valeurs dans :
$\square$ ses chants
$\square$ ses champs
□ ses cases

 $\square$  ses blocs

<pre>13. On souhaite faire une boucle de contrôle de saisie : tant     que l'entier n n'appartient pas à l'intervalle [ab], on     recommence la saisie de n. Soit le programme suivant :     int a = 0;     int b = 20;     int n;     scanf("%d", &amp;n);     while(cond)     {         scanf("%d", &amp;n);     }     Quelle est la condition cond :         □ (a<n) (n=""   ="">b)         □ (n&lt;=a) &amp;&amp; (n&lt;=b)         □ a&lt;=n&lt;=b</n)></pre>	<pre>□ un battement d'horloge processeur □ un chiffre binaire (0 ou 1)  16. Si cette erreur apparaît à la compilation :     error: expected ';' before '}' token que doit-     on chercher dans le programme? □ une accolade manquante □ une accolade en trop □ un point-virgule manquant □ un point-virgule en trop  17. Le code suivant :     int age = 20;     if (age &lt; 18)     {         printf("Mineur\n");     } </pre>	18. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :  □ yppasswd □ new TP4 □ mkdir TP4 □ kwrite TP4  19. Le bus système sert à : □ Écrire des données sur le dique dur □ transporter les processus du tourniquet au processeur □ Arriver à l'heure en cours
□ (a<=n) && (n<=b)  14. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :  □ struct date_s afficher_date(struct date_s	<pre>} else {     printf("Majeur\n"); }</pre>	☐ Transférer des données et intructions entre processeur et mémoire  20. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit- on chercher dans le programme?
□ void afficher_date(struct date_s d); □ int afficher_date(date_s d); □ void afficher_date(date_s d);  15. Un bit est: □ l'instruction qui met fin à un programme □ la longueur d'un mot mémoire	affichera:    Mineur	<ul> <li>□ une fonction déclarée mais non définie</li> <li>□ une fonction appelée avant sa déclaration</li> <li>□ un désaccord entre la déclaration et la définition d'une fonction</li> <li>□ une directive préprocesseur #include manquante</li> </ul>

т		-1
	acence	

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 0246
     \Box 0123456
     \Box 0 1 2 3 4 5 6 7
     \Box 02468
2. Pour déclarer une fonction pgcd qui calcule et renvoie
  le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit:
     \square void pgcd(int x, int y);
     \square int pgcd(int x, int x);
     \Box int pgcd(int x, y);
     \Box int pgcd(int y, int x);
3. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
4. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
```

```
Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 0
     \square 3
5. Un registre du processeur est :
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     □ une gamme de fréquence de fonctionnement du
     □ un composant qui contient la liste des fichiers du
        système
    \square une unité de calcul spécialisée de l'ordinateur
6. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \square 8
     \Box 4
     \Box 16
     \square 0
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     □ char 'c';
     \Box char c;
     ☐ int char;
     ☐ char "c";
8. Vous utilisez une boucle while quand :
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
```

□ vous n'avez pas déclaré de fonction

```
9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square double exposant(double x, int n);
     \square exposant(double x, int n, int r);
     □ void exposant(double x^n);
     \square int exposant(double n, int x);
10. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
     □ dans lequel ces fonctions sont appelées dans le
        main
     □ un ordre quelconque
     □ alphabétique
     □ dans lequel vous avez déclaré ces fonction
11. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
     □ void afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ double afficher_menu();
12. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses blocs
     \square ses chants
     \square ses cases
13. Le code suivant :
     int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    }
     else
         printf("Majeur\n");
    }
```

affichera:	16. Un bit est:	$\square$ #include <studlib.h></studlib.h>
☐ Mineur	□ la longueur d'un mot mémoire	☐ #include <stdio.h></stdio.h>
Majeur	☐ l'instruction qui met fin à un programme	
□ Majeur	$\square$ un battement d'horloge processeur	19. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
□ rien	$\square$ un chiffre binaire (0 ou 1)	,
☐ Mineur	17. Pour l'extrait de programme suivant :	$\square$ int pgcd(2);
14. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max	, int i = 0;	$\square$ n = pgcd(int p, int q);
, que doit-on chercher dans le programme?	int j = 0;	$\square$ int n = pgcd();
une fonction déclarée mais non définie	for (i = 0; i < 3; i = i + 1)	$\square \ n = pgcd(n, 3);$
□ une directive préprocesseur <b>#include</b> manquante	{	□ n - pgcu(n, 5),
$\Box$ une fonction appelée avant sa déclaration	for (j = 0; j < 2; j = j + 1)	20. Après exécution du programme :
$\square$ un désaccord entre la déclaration et la définition	printf("%d ", i);	1 lecture 8 r0
d'une fonction	}	2 valeur 3 r1
15. Le code suivant :	}	3 mult r1 r0
int i;	<pre>printf("\n");</pre>	4 valeur 1 r2
for $(i = 0; i < 5; i = i + 1)$	qu'est ce qui sera affiché?	5 add r2 r0 6 ecriture r0 8
{	$\Box$ 0 0 1 1 2 2	7 stop
<pre>printf("%d ", i); }</pre>	□ 0 1 0 1 0 1	8 5
<pre>printf("\n");</pre>	□ 0 1 2 0 1 2	
affichera:	□ 1 2 3 1 2	$\square$ la case mémoire 8 contiendra 0
$\square$ 0 1 2 3	18. Lorsqu'un programme utilise printf ou scanf il faut	$\Box$ le terminal affiche 8
$\Box \ 0\ 1\ 2\ 3\ 4$	qu'il contienne l'instruction préprocesseur :	□ la case mémoire 8 contiendra 16
$\Box \ 4\ 3\ 2\ 1\ 0$	☐ #appart <stdlib.h></stdlib.h>	$\square$ le bus explose
$\Box \ 4\ 3\ 2\ 1$	☐ #include <studio.h></studio.h>	□ ic bus explose

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- Un fichier source est :
   un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
  - $\Box$  un document illisible pour les humains
  - □ un document de référence du système
  - □ un document qui doit être protégé
  - $\square$  un fichier texte qui sera traduit en instructions processeur
- 2. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
  - ☐ kwrite TP4
  - □ new TP4
  - □ yppasswd
  - ☐ mkdir TP4
- 3. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
  - $\Box$  char c;
  - ☐ char 'c';
  - ☐ char "c";
  - ☐ int char;
- $4. \ \, \text{Si}$  cet avertissement apparaı̂t à la compilation :

warning: implicit declaration of function 'max', que doit-on chercher dans le programme?

- $\square$  une fonction appelée avant sa déclaration
- $\square$  une directive préprocesseur #include manquante
- $\square$  une fonction déclarée mais non définie
- $\Box$  un désaccord entre la déclaration et la définition d'une fonction
- 5. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)</pre>
```

```
{
         }
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
    \Box j = 0
    \Box j = 4
    \Box j = %d
     \Box j = 5
6. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     ☐ une directive préprocesseur #include manquante
     □ une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
     \square un désaccord entre la déclaration et la définition
        d'une fonction
7. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire:
     \square x = mccarthy(n);
     \square n = mccarthy(p, q);
     \square int mccarthy(int 2);
    \square n = mccarthy();
8. L'ordonnancement par tourniquet permet :
     ☐ d'afficher des ronds colorés à l'écran
     □ d'entretenir l'illusion que les processus tournent
        en parallèle
```

☐ de doubler la mémoire disponible

de contexte

 $\Box$  de ne pas perdre de temps avec la commutation

```
9. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
      ☐ for(i=1:i<=5:i=i+1)
10. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
      □ void afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu();
      ☐ char afficher_menu(printf("menu"));
      ☐ int afficher_menu(int char);
11. Le code suivant :
     int i:
     for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
     printf("\n");
    affichera:
     \square 0 1 2 3
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 01234
     \square 4 3 2 1
12. Les lignes
    int i;
   int x=0;
   for(i=0, i<5, i=i+1)
      x=x+1;
      □ comportent une erreur qui ne sera pas détectée
      □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
      \square ne comportent aucune erreur
      □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
```

```
13. Après exécution jusqu'à la ligne 15 du programme C:
  10
       . . .
       int main() {
  11
            int x = 5;
  12
  13
  14
            x = 3 * x + 1;
  15
  16
       }
  17
      \square la variable x vaut 16
      \Box le programme affiche ****
      \square le programme affiche x
      \square la variable x vaut -\frac{1}{2}
14. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
        return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 0
      \Box 7
      \Box 5
15. Soit le programme principal suivant :
    int main()
    {
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
    }
```

```
appelant la fonction f ainsi définie :
    int f(int a, int b)
      a = a + b;
      return a;
    L'affichage dans le main est le suivant :
      \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
      \Box f(3,5)=8, a=3, b=5
      \Box f(a,b)=8, a=8, b=5
16. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      ☐ il risque d'afficher bonjour à la place de coucou
      \square il ne compile pas
      \square il n'affiche rien
      \square il comporte une boucle infinie
17. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
         produit = produit * serie[i];
      printf("produit = %d", produit);
    La valeur affichée est :
      \Box 0
      \Box 16
      \square 8
      \Box 4
```

18.	Un enregistrement permet de grouper plusieurs valeurs dans :
	$\square$ ses champs
	$\square$ ses chants
	□ ses cases
	$\square$ ses blocs
19.	Le bus système sert à :
	$\Box$ Transférer des données et intructions entre processeur et mémoire
	$\Box$ Arriver à l'heure en cours
	$\Box$ Écrire des données sur le dique dur
	$\Box$ transporter les processus du tourniquet au processeur
20.	Soit la fonction f définie par :
	<pre>int f(int a) {</pre>
	<pre>printf("a = \n", %d); if (a &gt; 0) {</pre>
	return f(a - 1) + 1; }
	return 4; }
	Alors l'expression f(1) prendra la valeur :
	□ 4
	□ 5
	□ 1

 $\Box$  0

т		-1
	100000	- 1
	acence	

 $\Box$  4

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

r r	réponse fausse. Durée : 20 minutes.
1.	Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction
	$\square$ loop i;
	$\square$ int loop n;
	☐ int %d;
	□ int k;
2.	Quel est le problème d'un programme comportant le lignes suivantes?
	while (1) {
	<pre>printf("coucou\n"); }</pre>
	$\Box$ il n'affiche rien
	$\square$ il comporte une boucle infinie
	$\Box$ il risque d'afficher bonjour à la place de coucou
	$\Box$ il ne compile pas
3.	Pour déclarer une procédure afficher_menu sans ar gument et qui ne renvoie rien on utilise :
	☐ double afficher_menu();
	☐ int afficher_menu(int char);
	☐ int afficher_menu();
	☐ char afficher_menu(printf("menu"));
	□ void afficher_menu();
4.	Soit un programme contenant les lignes suivantes :
	<pre>int i = 0; int j = 0; for (i = 0; i &lt; 0; i = i + 1)</pre>
	{
	for (j = 0; j < 5; j = j + 1) {
	}
	<pre>printf("j = %d\n", j);</pre>
	 ኔ

```
qu'est ce qui sera affiché?
     \Box j = 0
     \Box j = 5
     \Box j = %d
     \Box j = 4
5. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 101
     \Box 5
     \square 3
     \Box 4
6. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     \Box un désaccord entre la déclaration et la définition
        d'une fonction
     \square une fonction appelée avant sa déclaration
     □ une directive préprocesseur #include manquante
     □ une fonction déclarée mais non définie
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses chants
     \square ses cases
     \square ses champs
     \square ses blocs
8. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
        return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \square 0
     \Box 5
     \Box 1
```

```
9. Une segmentation fault est une erreur qui survient
   lorsque :
      □ la division du programme en zones homogènes
         échoue
      □ le programme tente d'afficher des caractères sur
        une ligne qui va au delà de la largeur de la fenêtre
        du terminal
      □ le programme tente d'accèder à une partie de la
         mémoire qui ne lui est pas réservée
      ☐ le programme source a été enregistré sur le disque
         dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
10. Soit la fonction f définie par :
   int f(int a)
   ₹
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
     \Box 0
      \Box 4
      \square 3
11. Après exécution jusqu'à la ligne 15 du programme C :
       int main() {
 11
            int x = 5;
 12
            int y;
 13
 14
            y = x;
 15
 16
 17
      ☐ le programme affiche "Faux"
     \square la variable x vaut 5 et la variable y vaut 0
      \square la variable y vaut 5
      \square la variable x vaut 0
```

12. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :	15. Vous utilisez une boucle while quand :  □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	<pre>for (j = 0; j &lt; 5; j = j + 1) {</pre>
<pre>□ printf("x=%d et y=%d\n",x y);</pre> □ printf("x=%d et y=%d\n,x,y");	<ul> <li>□ vous n'avez pas déclaré de fonction</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> </ul>	 } printf("j = %d\n", j);
☐ printf("x=%d et y=%d\n",x,y); ☐ printf("x=%x et y=%y\n");  13. Si factorielle est une fonction prenant en entrée un	☐ l'incrément de la variable de boucle n'est pas 1  16. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la pro-	qu'est ce qui sera affiché par ce printf?
<pre>entier et renvoyant un entier, il est correct d'écrire :</pre>	grammation structurée :  □ répéter un bloc tant qu'une condition est vérifée □ retourner un bloc	<ul><li>□ j = %d</li><li>□ j = 0</li><li>□ j = 5</li></ul>
□ n = factorielle(p, q); □ int factorielle(int 2);  14. Le code suivant :	<ul> <li>□ sélectionner entre deux blocs à l'aide d'une condition</li> <li>□ mettre les blocs en séquence les uns à la suite des autres</li> </ul>	$\hfill \square$ j = 4 19. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :
<pre>int i; for (i = 1; i &lt; 5; i = i + 1) {     printf("%d ", i); } printf("\n"); affichera:</pre>	17. Un registre du processeur est :  □ une gamme de fréquence de fonctionnement du processeur  □ une unité de calcul spécialisée de l'ordinateur  □ un composant qui contient la liste des fichiers du système  □ une case mémoire interne au processeur qui sera	□ 1 □ 0 □ 0.5 □ 1.5  20. Dans la commande gcc, l'option -Wall signifie :
$ \Box 0 1 2 3 4  \Box 4 3 2 1 0  \Box 1 2 3 4 $	manipulée directement lors des calculs  18. Soit un programme contenant les lignes suivantes :  int i = 0;  int j = 0;	<ul> <li>□ qu'il faut lancer un déboggueur</li> <li>□ que l'on veut voir tous les avertissements</li> <li>□ qu'il faut indenter le fichier source</li> <li>□ qu'on veut changer alétoirement de fond d'écran</li> </ul>

## Éléments d'informatique – contrôle continue

};

Prénom: Nom	:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le code suivant : int i: for (i = 0; i < 7; i = i + 2)printf("%d ", i); printf("\n"); affichera:  $\Box$  0 1 2 3 4 5 6  $\Box 02468$  $\Box$  0 1 2 3 4 5 6 7  $\Box 0246$ 2. L'ordonnancement par tourniquet permet : □ d'entretenir l'illusion que les processus tournent en parallèle □ d'afficher des ronds colorés à l'écran  $\square$  de doubler la mémoire disponible  $\square$  de ne pas perdre de temps avec la commutation de contexte 3. Le type des réels en C est : ☐ double □ char  $\square$  int □ real 4. Après exécution jusqu'à la ligne 15 du programme C : 10 11 int main() { 12 int x = 5: 13 14 x = 3 \* x + 1;15 16 17 }  $\square$  le programme affiche x  $\square$  le programme affiche \*\*\*\* □ la variable x vaut 16  $\square$  la variable x vaut  $-\frac{1}{2}$ 

```
5. Vous utilisez une boucle while quand:
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
    □ vous n'avez pas déclaré de fonction
     ☐ l'incrément de la variable de boucle n'est pas 1
     \square vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
6. Après exécution du programme :
      lecture 8 r0
      valeur 3 r1
      mult r1 r0
      valeur 1 r2
      add r2 r0
      ecriture r0 8
      stop
      5
     □ la case mémoire 8 contiendra 16
     □ la case mémoire 8 contiendra 0
     □ le terminal affiche 8
     \square le bus explose
7. Sur unix (ou linux), la commande mkdir permet de :
     \square créer un fichier texte
     □ changer de répertoire courant
     □ ouvrir un fichier texte
     □ créer un répertoire
8. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     □ ses chants
     \square ses blocs
     \square ses champs
9. Si a et b sont deux variables de type:
   struct toto_s
     int n;
     double x;
```

```
Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \Box a = b
     \square (a.n == b.n) && (a.x == b.x)
      \square a\{n, x\} == b\{n, x\}
      □ a == b
10. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
      ☐ #define taille = 3
      □ #define N 3
     ☐ #define taille = N
      \square #define N = 3
11. Le code suivant :
     int i;
     for (i = 4; i > 0; i = i - 1)
          printf("%d ", i);
     printf("\n");
    affichera:
     \square 4 3 2 1
     \Box 01234
     \square 0 1 2 3
      \Box \ 4\ 3\ 2\ 1\ 0
12. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
      \square dans lequel vous avez déclaré ces fonction
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
      □ un ordre quelconque
```

```
13. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 8
     \Box 4
     \Box 0
     \square 16
14. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      }
      return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 0
     \Box 4
     \square 3
```

```
15. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      \square une fonction appelée avant sa déclaration
      □ une fonction déclarée mais non définie
      \square un désaccord entre la déclaration et la définition
         d'une fonction
      □ une directive préprocesseur #include manquante
16. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      ☐ int factorielle();
     ☐ int factorielle(int x);
      ☐ int factorielle(double n);
      □ struct int factorielle(int n);
17. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
      \square une fonction appelée avant sa déclaration
     □ une directive préprocesseur #include manquante
      □ une fonction déclarée mais non définie
     \square un désaccord entre la déclaration et la définition
         d'une fonction
18. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      □ main
      □ begin
      \square init
      \square include
```

```
19. Le code suivant :
     int age = 20;
    if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     \square rien
     □ Majeur
     ☐ Mineur
     □ Mineur
        Majeur
20. Si le code:
   struct toto_s
     int n;
     double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
     ☐ struct toto_s toto;
     \square toto_s n, x;
     \square int toto.n = 3;
     \Box int struct toto_s = {3, -1e10};
     \Box toto_s struct z = {3, 0.5};
```

## Éléments d'informatique – contrôle continue

Prénom: Nom	:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0,5 points par réponse fausse. Durée: 20 minutes.

1. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit:

\[ \subseteq \text{void afficher\_date(date\_s d);} \]

\[ \subseteq \text{void afficher date(struct date s d):} \]

2. Au début de la fonction main() on place le code :

```
char i;
for (i = 'A'; i <= 'F'; i = i + 1)
{
    printf("%c", i);
}
printf("\n");
Alors l'affichage sera :</pre>
```

☐ int afficher date(date s d):

☐ ABCDEF

□ i
□ A

3. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);
...
}</pre>
```

qu'est ce qui sera affiché?

4. Vous utilisez une boucle while quand :

□ 1 increment de la variable de boucle il est pas 1
$\hfill \square$ vous ne connaissez pas le nombre d'itérations de
la boucle à l'avance
□ vous n'avez pas déclaré de fonction

□ vous avez déjà fait un **for** dans le même pro-

struct date\_s afficher\_date(struct date\_s d); gramme principal

5. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?

 $\Box$  un désaccord entre la déclaration et la définition d'une fonction

 $\square\,$ une fonction déclarée mais non définie

 $\Box\,$ une fonction appelée avant sa déclaration

 $\Box\,$  une directive préprocesseur  $\# {\tt include}$  man quante

6. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
  printf("coucou\n");
}
```

 $\Box\,$ il risque d'afficher bonjour à la place de coucou

 $\Box\,$ il ne compile pas

 $\Box\,$ il comporte une boucle infinie

 $\Box\:$ il n'affiche rien

7. L'ordonnancement par tourniquet permet :

 $\hfill \Box$  d'afficher des ronds colorés à l'écran

 $\Box\,$  de doubler la mémoire disponible

 $\Box$  d'entretenir l'illusion que les processus tournent en parallèle

 $\Box$  de ne pas perdre de temps avec la commutation de contexte

8. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie:
  int f(int a, int b)
{
    a = a + b;
    return a;
```

L'affichage dans le main est le suivant :

```
☐ f(3,5)=8, a=3, b=5
☐ f(a,b)=13, a=8, b=5
☐ f(a,b)=8, a=3, b=5
☐ f(a,b)=8, a=8, b=5
```

9. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
    x = racine(2/3);

    x = racine(racine(x)*racine(x));

    x - 1 = racine(x);

    x = racine(x * x) - racine(x);
```

10. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :

```
□ int n = carre();
□ n = carre(n);
□ n = carre(int n);
□ int carre(2);
```

11. Pour compiler un programme prog.c, on utilise la ligne de commande :  ☐ gcc -Wall prog.exe -o prog.c ☐ gcc prog.exe -Wall -o prog.c ☐ gcc prog.c -o -Wall prog.exe ☐ gcc -Wall prog.c -o prog.exe  12. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise : ☐ void afficher_menu(); ☐ int afficher_menu();	14. Après exécution jusqu'à la ligne 15 du programme C:  10 int main() {  11 int x = 5;  12 int y;  13  14 y = x;  15  16  17 }  ☐ le programme affiche "Faux" ☐ la variable y vaut 5 ☐ la variable x vaut 5 et la variable y vaut 0 ☐ la variable x vaut 0	<pre>18. Le code suivant :     int i;     for (i = 8; i &gt; 0; i = i - 2)     {         printf("%d ", i);     }     printf("\n");     affichera :     □ 8 2     □ 8 6 4 2 0     □ 0 2 4 6 8</pre>
<pre>□ double afficher_menu();</pre> □ int afficher_menu(int char); □ char afficher_menu(printf("menu"));	<ul> <li>15. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?</li> <li>□ int char;</li> <li>□ char c;</li> </ul>	<ul> <li>□ 8 6 4 2</li> <li>19. L'écriture <u>101</u> en binaire correspond au nombre nat</li> </ul>
13. Après exécution du programme :  1 lecture 8 r0 2 valeur 3 r1 3 mult r1 r0 4 valeur 1 r2 5 add r2 r0 6 ecriture r0 8 7 stop 8 5  □ la case mémoire 8 contiendra 0 □ la case mémoire 8 contiendra 16 □ le bus explose □ le terminal affiche 8	char 'c'; char 'c'; char "c";  16. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction int toto[taille=5]; int[] new tableau(5); char tableau[5]; int toto[5]; int toto[5]; int tab[] = 5;  17. Une variable booléenne est un variable : NaN (not a number, qui n'est pas un nombre) réelle positive à laquelle une valeur vient d'être affectée qui est vraie ou fausse jamais nulle	rel:  4  5  3  101  20. Le bus système sert à :  Écrire des données sur le dique dur  Arriver à l'heure en cours  Transférer des données et intructions entre pr cesseur et mémoire  transporter les processus du tourniquet au pr cesseur

# Éléments d'informatique – contrôle continue

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1)for (j = 0; j < 5; j = j + 1)}  $printf("j = %d\n", j);$ qu'est ce qui sera affiché par ce printf?  $\Box$  j = 0  $\Box$  j = %d □ j = 5  $\Box$  j = 4 2. L'écriture 101 en binaire correspond au nombre naturel:  $\Box$  4  $\square$  101  $\square$  3  $\square$  5 3. Un fichier source est: □ un document illisible pour les humains □ un document de référence du système  $\square$  un fichier texte qui sera traduit en instructions processeur □ un document qui doit être protégé  $\square$  un fichier que l'ont doit citer dans les documents produits sur l'ordinateur 4. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on

recommence la saisie de n. Soit le programme suivant :

```
int a = 0:
   int b = 20;
   int n;
   scanf("%d", &n);
   while(cond)
      scanf("%d", &n);
   Quelle est la condition cond:
    □ a<=n<=b
    □ (a<=n) && (n<=b)
    \square (n<=a) && (n<=b)
    \Box (a<n) || (n>b)
5. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
       somme = somme + i:
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
  La valeur de somme affichée est :
    \square 15
    \Box 10
    \Box 0
    \Box 6
6. Le code suivant :
   int age = 20;
   if (age < 18)
   {
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Majeur
    \square rien
    ☐ Mineur
       Majeur
    □ Mineur
```

```
N^{\circ} INE:
     7. Un bit est:
          \square un chiffre binaire (0 ou 1)
          ☐ l'instruction qui met fin à un programme
          □ un battement d'horloge processeur
          □ la longueur d'un mot mémoire
    8. Si n est une variable entière pour demander sa valeur
       à l'utilisateur, on utilise plutôt :
          □ un débogueur
          □ scanf("%d", &n);
         ☐ printf("Valeur de n ? %g\n", n);
          ☐ printf("Valeur de n ? %d\n", n);
     9. Si racine est une fonction prenant en entrée un réel
        et renvoyant la racine carrée de cet réel, et que x est
       une variable réelle définie et initialisée, il est incorrect
       d'écrire :
         \square x = racine(2/3):
         \Box x = racine(x * x) - racine(x);
         \square x = racine(racine(x)*racine(x));
         \square x - 1 = racine(x);
   10. Après exécution jusqu'à la ligne 15 du programme C:
     10
           int main() {
     11
                int x = 5;
     12
                int y;
     13
     14
                y = x;
     15
     16
     17
          □ le programme affiche "Faux"
          \square la variable x vaut 5 et la variable y vaut 0
          \square la variable x vaut 0
          \square la variable y vaut 5
```

Nom:

Prénom:

```
18. Le type des réels en C est :
11. Soit la fonction f définie par :
                                                                       {
                                                                            printf("%d ", i);
    int f(int a)
                                                                                                                                 □ double
                                                                       }
                                                                  }
      printf("a = \n", %d);
                                                                                                                                 \square real
                                                                  printf("\n");
      if (a > 0)
                                                                 qu'est ce qui sera affiché?
                                                                                                                                 □ char
                                                                   \Box 0 1 0 1 0 1
        return f(a - 1) + 1;
                                                                                                                                 \square int
                                                                   \Box 0 0 1 1 2 2
      return 4;
                                                                   \Box 1 2 3 1 2
                                                                                                                           19. Si cette erreur apparaît à la compilation :
                                                                   \Box 0 1 2 0 1 2
                                                                                                                               error: expected ';' before '}' token que doit-
    Alors l'expression f(1) prendra la valeur :
                                                             15. Quel est le problème d'un programme comportant les
                                                                                                                               on chercher dans le programme?
                                                                 lignes suivantes?
      \Box 1
                                                                                                                                 \square une accolade en trop
                                                                 while (1)
      \Box 5
                                                                                                                                 ☐ un point-virgule manquant
      \Box 0
                                                                    printf("coucou\n");
      \Box 4
                                                                                                                                 \Box une accolade manquante
12. Un enregistrement permet de grouper plusieurs valeurs
                                                                   □ il n'affiche rien
                                                                                                                                 □ un point-virgule en trop
    dans:
                                                                   \square il ne compile pas
      \square ses blocs
                                                                   □ il risque d'afficher bonjour à la place de coucou
                                                                                                                           20. Après exécution du programme :
      \square ses chants
                                                                   \square il comporte une boucle infinie
      \square ses cases
                                                             16. Si carre est une fonction prenant en entrée un en-
                                                                                                                                  lecture 8 r0
                                                                                                                                   valeur 3 r1
                                                                 tier et renvoyant le carré de cet entier, et que n est
      \square ses champs
                                                                                                                                  mult r1 r0
                                                                 une variable entière définie et initialisée, il est correct
13. Pour déclarer une fonction factorielle qui prend en
                                                                                                                                  valeur 1 r2
                                                                 d'écrire :
   argument un entier et renvoie sa factorielle on écrit :
                                                                                                                                   add r2 r0
                                                                   \square n = carre(int n):
      ☐ int factorielle(double n):
                                                                                                                                   ecriture r0 8
                                                                   \square int carre(2);
                                                                                                                                   stop
      ☐ int factorielle();
                                                                   \square n = carre(n);
                                                                                                                                   5
      \square int factorielle(int x):
                                                                   \square int n = carre();
      □ struct int factorielle(int n);
                                                             17. Vous utilisez une boucle while quand:
                                                                                                                                 □ la case mémoire 8 contiendra 0
                                                                   □ vous n'avez pas déclaré de fonction
14. Pour l'extrait de programme suivant :
                                                                   \Box vous ne connaissez pas le nombre d'itérations de
                                                                                                                                 \square le terminal affiche 8
     int i = 0;
                                                                      la boucle à l'avance
     int j = 0;
                                                                                                                                 □ la case mémoire 8 contiendra 16
                                                                   \square l'incrément de la variable de boucle n'est pas 1
     for (i = 0; i < 3; i = i + 1)
                                                                   □ vous avez déjà fait un for dans le même pro-
                                                                                                                                 \square le bus explose
                                                                      gramme principal
         for (j = 0; j < 2; j = j + 1)
```

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit le programme principal suivant :
  int main()
  {
    int a = 3;
    int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
    return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
2. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
            printf("%d ", j);
  qu'est ce qui sera affiché?
    \Box 0 1 2 0 1 2 3
    \Box 0 1 2 0 1 2
    \Box 0 0 1 1 2 2 3
    \Box 0 1 2 3 0 1 2
```

```
3. Les lignes
   int i:
   int x=0:
   for(i=0,i<5,i=i+1)
     x=x+1;
     □ comportent une erreur qui ne sera pas détectée
     □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
     \square ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
4. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
     \square ses chants
     \square ses cases
     \square ses champs
5. Une segmentation fault est une erreur qui survient
   lorsque:
     □ le programme source a été enregistré sur le disque
        dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
     □ la division du programme en zones homogènes
        échoue
     □ le programme tente d'afficher des caractères sur
        une ligne qui va au delà de la largeur de la fenêtre
        du terminal
     □ le programme tente d'accèder à une partie de la
        mémoire qui ne lui est pas réservée
6. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \Box int mccarthy(int 2);
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
     \square n = mccarthy();
```

7.	Sur unix (ou linux), la commande mkdir permet de :
	$\Box$ créer un fichier texte
	$\Box$ changer de répertoire courant
	□ créer un répertoire
	$\Box$ ouvrir un fichier texte
8.	Un fichier source est :
	$\Box$ un document de référence du système
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\Box$ un document illisible pour les humains
	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\Box$ un document qui doit être protégé
9.	Si $n$ est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :
	$\square$ un débogueur
	$\square$ printf("Valeur de n ? %g\n", n);
	□ scanf("%d", &n);
	☐ printf("Valeur de n ? %d\n", n);
10.	$L'ordonnancement\ par\ tourniquet\ permet:$
	$\Box$ d'afficher des ronds colorés à l'écran
	$\Box$ de doubler la mémoire disponible
	$\Box$ d'entretenir l'illusion que les processus tournent en parallèle
	$\Box$ de ne pas perdre de temps avec la commutation de contexte
11.	Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
	$\Box$ dans lequel ces fonctions sont appelées dans le main
	$\hfill\Box$ alphabétique
	$\Box$ un ordre quel conque
	□ dans lequel vous avez déclaré ces fonction

```
12. Le code suivant :
                                                                 {
                                                                                                                               □ b
                                                                      printf("%d ", i);
                                                                                                                               □В
     int age = 18;
     if (age < 18)
                                                                                                                               \Box C
                                                                 printf("\n");
                                                                                                                               \square A
                                                                affichera:
         printf("Mineur\n");
                                                                                                                         18. On souhaite faire une boucle de contrôle de saisie : tant
                                                                  \square 0 2 4 6
                                                                                                                             que l'entier n n'appartient pas à l'intervalle [a..b], on
     else
                                                                  \Box 0 1 2 3 4 5 6 7
                                                                                                                             recommence la saisie de n. Soit le programme suivant :
                                                                  \Box 02468
         printf("Majeur\n");
                                                                                                                              int a = 0;
                                                                  \Box 0 1 2 3 4 5 6
                                                                                                                              int b = 20;
                                                            15. Sous unix (ou linux), pour créer un répertoire TP4
                                                                                                                              int n;
                                                                dans le répertoire courant on peut utiliser la com-
   affichera:
                                                                                                                              scanf("%d", &n);
                                                                mande:
                                                                                                                              while(cond)
     □ Majeur
                                                                  ☐ kwrite TP4
      ☐ Mineur
                                                                                                                                scanf("%d", &n);
                                                                  ☐ yppasswd
     \square rien
                                                                  ☐ mkdir TP4
      ☐ Mineur
                                                                                                                             Quelle est la condition cond:
                                                                  □ new TP4
        Majeur
                                                                                                                               \square (a<n) || (n>b)
                                                            16. Le code suivant :
13. Soit la fonction f définie par :
                                                                                                                               □ (a<=n) && (n<=b)
                                                                 int i;
   int f(int a)
                                                                                                                               □ (n<=a) && (n<=b)
                                                                 for (i = 4; i > 0; i = i - 1)
                                                                                                                               □ a<=n<=b
      printf("a = \n", %d);
                                                                      printf("%d ", i);
      if (a > 0)
                                                                                                                         19. Si factorielle est une fonction prenant en entrée un
                                                                                                                             entier et renvoyant un entier, il est correct d'écrire :
                                                                 printf("\n");
        return 3;
                                                                                                                               \square n = factorielle();
      }
                                                                affichera:
                                                                                                                              ☐ printf("%d", factorielle(n));
      return 4;
                                                                  \square 0 1 2 3
                                                                                                                               ☐ int factorielle(int 2);
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
                                                                                                                               \square n = factorielle(p, q);
   Alors l'expression f(0) prendra la valeur :
                                                                  \Box 01234
     \square 3
                                                                                                                         20. Si x est une variable réelle (de type double) alors
                                                                  \square 4 3 2 1
                                                                                                                             x = 3/2 lui affecte la valeur :
     \Box 0
                                                            17. Au début de la fonction main() on place le code :
                                                                                                                               \Box 1
     \Box 4
                                                                 char b = 'A';
                                                                                                                               \square 0.5
14. Le code suivant :
                                                                 b = b + 2;
                                                                                                                               \Box 0
                                                                 printf("%c\n", b);
     int i;
                                                                                                                               \square 1.5
                                                                Alors l'affichage sera :
     for (i = 0; i < 7; i = i + 2)
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

int i = 0;

9. Soit un programme contenant les lignes suivantes :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Quels calculs peut-on programmer en programmation structurée?
  - $\Box$  certains programmes sont de vrais plats de spaghetti
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
  - □ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
  - □ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
- 2. Pour compiler un programme prog.c, on utilise la ligne de commande :
  - ☐ gcc -Wall prog.c -o prog.exe ☐ gcc prog.c -o -Wall prog.exe
  - ☐ gcc -Wall prog.exe -o prog.c ☐ gcc prog.exe -Wall -o prog.c
- 3. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
  printf("coucou\n");
}
```

- $\Box\,$ il comporte une boucle infinie
- $\Box\,$ il n'affiche rien
- $\Box\,$ il ne compile pas
- $\Box\,$ il risque d'afficher bonjour à la place de coucou
- 4. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
```

```
appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
   }
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
5. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ un débogueur
    ☐ printf("Valeur de n ? %g\n", n);
    □ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
6. Sur unix (ou linux), la commande mkdir permet de :
    □ changer de répertoire courant
     □ créer un fichier texte
     □ ouvrir un fichier texte
     □ créer un répertoire
7. Une segmentation fault est une erreur qui survient
   lorsque:
     □ le programme tente d'afficher des caractères sur
       une ligne qui va au delà de la largeur de la fenêtre
        du terminal
     □ la division du programme en zones homogènes
        échoue
     □ le programme source a été enregistré sur le disque
        dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
```

 $\square$  le programme tente d'accèder à une partie de la

mémoire qui ne lui est pas réservée

8. Le langage C est un langage

□ lu, écrit, parlé

□ composé

□ compilé

□ interprété

```
int j = 0;
    for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
    }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1 0 1
     \Box 0 0 0 1 1 1
     \Box 1 2 1 2 3
10. Le code suivant :
    int age = 18;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     □ rien
     □ Majeur
     □ Mineur
        Majeur
     ☐ Mineur
```

```
11. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
     {
       scanf("%d", &n);
   Quelle est la condition cond :
      \square (a<=n) && (n<=b)
      □ a<=n<=b
      \square (a<n) || (n>b)
      \square (n<=a) && (n<=b)
12. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
      ☐ char "c";
      ☐ int char;
      □ char 'c';
      □ char c;
13. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
      \square int pgcd(2);
     \square n = pgcd(int p, int q);
      \square n = pgcd(n, 3);
      \square int n = pgcd();
14. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
      \square for(i=1;i<=5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<5;i=i+1)
      \square for(i=0;i<5;i=i+1)
```

```
15. On considère deux variables booléennes A et B initia-
                                                                    return 5;
    lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                 }
    pressions booléennes suivantes, laquelle a pour valeur
                                                                 return 7;
    TRUE?
                                                               }
     \square !(!A || B) == (A && !B)
                                                               Alors l'expression g(0) prendra la valeur :
     \square (!A || B)
                                                                 \Box 7
     \square (A == TRUE) && (B == TRUE)
                                                                 \Box 0
     □ A && B
                                                                 \Box 5
16. Si le code:
                                                           19. Pour l'extrait de programme suivant :
    struct toto_s
                                                                 int produit = 0;
                                                                 int serie[4] = \{2, 2, 2, 2\};
      int n;
                                                                 for (i = 0; i < 4; i = i + 1)
      double x;
   };
                                                                    produit = produit * serie[i];
    précède la fonction main(), alors on peut écrire en
   début de main() :
                                                                 printf("produit = %d", produit);
     □ struct toto_s toto;
                                                               La valeur affichée est :
     \square int toto.n = 3;
                                                                 \Box 0
     \square toto_s n, x;
                                                                 \Box 4
     \Box toto_s struct z = {3, 0.5};
                                                                 \square 8
     \Box int struct toto_s = {3, -1e10};
                                                                 \Box 16
17. Pour déclarer une fonction saisie_utilisateur qui
                                                           20. Le code suivant :
    demande à l'utilisateur d'entrer un entier au clavier et
    renvoie cet entier on écrit :
                                                                int i:
                                                                for (i = 8; i > 0; i = i - 2)
     ☐ int saisie_utilisateur();
     ☐ saisie_utilisateur(scanf(%d));
                                                                     printf("%d ", i);
     □ void saisie_utilisateur(int n);
                                                                printf("\n");
      □ void saisie_utilisateur(char c);
18. Soit la fonction g définie par :
                                                               affichera:
                                                                 \square 8 6 4 2
    int g(int a)
                                                                 \Box 8 6 4 2 0
      printf("a = \n", %d);
                                                                 \Box 02468
      if (1 > 0)
                                                                 \square 8 2
      {
```

 $\square$  7

## Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Au début de la fonction main() on place le code :
    char i:
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
  Alors l'affichage sera:
     ☐ ABCDEF
    □ ccccc
    \Box i
    □ A
2. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Mineur
    □ rien
    □ Majeur
     ☐ Mineur
       Majeur
3. L'écriture 111 en binaire correspond au nombre natu-
  rel:
     \square 3
    \square 111
    \square 8
```

```
4. Soit la fonction f définie par :
   int f(int a)
   {
     printf("a = \n", %d);
     if (a > 0)
     {
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \Box 0
     \square 3
5. L'ordonnancement par tourniquet permet :
     ☐ de doubler la mémoire disponible
     ☐ d'afficher des ronds colorés à l'écran
     \square de ne pas perdre de temps avec la commutation
        de contexte
     ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
6. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
     \square (A == TRUE) && (B == TRUE)
    \square !(!A \mid | B) == (A \&\& !B)
     □ A && B
     \square (!A | | B)
7. Le bus système sert à :
     □ transporter les processus du tourniquet au pro-
        cesseur
     ☐ Écrire des données sur le dique dur
     ☐ Arriver à l'heure en cours
     ☐ Transférer des données et intructions entre pro-
```

cesseur et mémoire

```
8. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
 9. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
     }
     printf("j = %d\n", j);
     }
   qu'est ce qui sera affiché?
     \Box j = 4
     \Box i = 0
     \Box i = %d
     \Box j = 5
10. Vous utilisez une boucle while quand:
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
11. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%x et y=%y\n");
     \Box printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n,x,y");
```

 $\square$  printf("x=%d et y=%d\n",x y);

12. Afin de représenter la taille d'un tableau, définir une	16. Le langage C est un langage
constante symbolique N valant 3.	$\square$ lu, écrit, parlé
☐ #define N = 3	$\square$ compilé
☐ #define taille = 3	$\square$ interprété
□ #define taille = N	$\square$ composé
□ #define N 3	17. Vous avez déclaré préalablement un ensemble de fonc-
13. Quel est l'opérateur de différence en C :	tions utilisées par votre programme principal. L'ordre
□ !	dans lequel vous devez maintenant définir ces fonctions
$\Box \Leftrightarrow$	est l'ordre :
□ ≠	☐ dans lequel ces fonctions sont appelées dans le main
□ !=	
14. Pour déclarer une fonction saisie_utilisateur qui	un ordre quelconque
demande à l'utilisateur d'entrer un entier au clavier et	□ alphabétique
renvoie cet entier on écrit :	$\square$ dans lequel vous avez déclaré ces fonction
□ void saisie_utilisateur(char c);	18. Si pgcd est une fonction prenant en entrée deux entiers
☐ saisie_utilisateur(scanf(%d));	et renvoyant un entier, il est correct d'écrire :
□ void saisie_utilisateur(int n);	$\square \ n = pgcd(n, 3);$
$\square$ int saisie_utilisateur();	$\square$ n = pgcd(int p, int q);
15. Si x est une variable réelle (de type double) alors	$\square$ int pgcd(2);
x = 3/2 lui affecte la valeur :	$\square$ int n = pgcd();
	19. Après exécution jusqu'à la ligne 15 du programme C:
$\square 0.5$	10
$\Box$ 1.5	11 int main() {
	12   int x = 5;

```
13
14
x = 3 * x + 1;
15
16
...
17
}
\Box \text{ le programme affiche *****}
\Box \text{ la variable x vaut } -\frac{1}{2}
\Box \text{ le programme affiche x}
\Box \text{ la variable x vaut } 16
```

20. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  ${\tt n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  ${\tt n}$ . Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
    scanf("%d", &n);
}
Quelle est la condition cond:
    (a<=n) && (n<=b)
    a<=n<=b
    (a<n) || (n>b)
```

□ (n<=a) && (n<=b)</pre>

};

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. Une segmentation fault est une erreur qui survient lorsque : □ la division du programme en zones homogènes □ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée □ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal  $\square$  le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur 2. Le type des réels en C est : ☐ double □ char □ int. □ real 3. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit: □ void afficher\_date(date\_s d); □ struct date\_s afficher\_date(struct date\_s d); □ void afficher\_date(struct date\_s d); ☐ int afficher\_date(date\_s d); 4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :  $\square$  analyse lexicale  $\square$  analyse syntaxique □ analyse sémantique  $\square$  analyse harmonique 5. Si le code : struct toto\_s int n; double x;

```
précède la fonction main(), alors on peut écrire en
   début de main():
     \square toto_s n, x;
     □ struct toto_s toto;
     \Box toto_s struct z = {3, 0.5};
     \Box int toto.n = 3;
     \square int struct toto_s = {3, -1e10};
6. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <studio.h>
     ☐ #include <studlib.h>
     ☐ #appart <stdlib.h>
     ☐ #include <stdio.h>
7. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \Box 7
     □ 111
     \square 8
8. L'ordonnancement par tourniquet permet :
     \square de ne pas perdre de temps avec la commutation
        de contexte
     □ d'afficher des ronds colorés à l'écran
     ☐ de doubler la mémoire disponible
     □ d'entretenir l'illusion que les processus tournent
        en parallèle
9. La virtualisation de la mémoire permet notamment de
  stocker des portions inactives de la mémoire de travail
   sur le disque dur. Mais on perd :
     \square les fichiers du disque
     \square des processus
     □ certaines données de la mémoire de travail
     □ en temps d'accès
```

```
10. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0:
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
   Quelle est la condition cond :
      □ (n<=a) && (n<=b)
      \square (a<n) || (n>b)
      □ a<=n<=b
      ☐ (a<=n) && (n<=b)
11. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre :
     □ alphabétique
      □ un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
      □ dans lequel ces fonctions sont appelées dans le
        main
12. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \Box 0123
      \square 4 3 2 1
```

 $\Box \ 4\ 3\ 2\ 1\ 0$ 

18. Sur unix (ou linux), la commande mkdir permet de : 13. Soit la fonction f définie par :  $\Box$  j = 4 □ changer de répertoire courant  $\Box$  j = 5 int f(int a) {  $\Box$  j = 0 □ ouvrir un fichier texte  $printf("a = \n", %d);$  $\Box$  j = %d □ créer un répertoire if (a > 0)□ créer un fichier texte 15. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doitreturn 3; 19. Sous unix (ou linux), la commande cd permet de : on chercher dans le programme? □ changer de répertoire courant return 4;  $\square$  une accolade en trop □ ouvir un bureau partagé (common desktop)  $\square$  une accolade manquante □ récupérer un programme arrêté avec la commande Alors l'expression f(0) prendra la valeur : □ un point-virgule en trop ab  $\square$  3  $\square$  un point-virgule manquant  $\Box$ jouer de la musique  $\Box$  0  $\square$  détruire un fichier 16. Laquelle de ces écritures correspond à la déclaration  $\Box$  4 d'une variable de type caractère en langage C? 20. Si a et b sont deux variables de type: 14. Soit un programme contenant les lignes suivantes :  $\Box$  char c; struct toto\_s int i = 0; ☐ int char; int j = 0; ☐ char "c"; int n; for (i = 0; i < 3; i = i + 1)double x; ☐ char 'c'; }; for (j = 0; j < 5; j = j + 1)17. Une variable booléenne est un variable : Alors pour tester l'égalité de a et de b on utilise la □ NaN (not a number, qui n'est pas un nombre) condition: □ réelle positive  $\square a\{n, x\} == b\{n, x\}$ □ à laquelle une valeur vient d'être affectée  $\square (a.n == b.n) && (a.x == b.x)$  $printf("j = %d\n", j);$ □ jamais nulle  $\square$  a = b  $\square$  qui est vraie ou fausse qu'est ce qui sera affiché par ce printf? □ a == b

т		-1
	100000	- 1
	acence	

Prénom:  $N^{\circ}$  INE:

Nom:		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Un enregistrement permet de grouper plusieurs valeurs dans:

```
□ ses cases
```

 $\square$  ses blocs

 $\square$  ses champs

 $\square$  ses chants

2. Après exécution du programme :

```
lecture 8 r0
valeur 3 r1
```

mult r1 r0

valeur 1 r2

add r2 r0

ecriture r0 8

stop

5

 $\square$  le bus explose

□ le terminal affiche 8

□ la case mémoire 8 contiendra 16

□ la case mémoire 8 contiendra 0

3. Après exécution jusqu'à la ligne 15 du programme C:

```
10
     int main() {
11
         int x = 5;
12
         int y;
13
14
         y = x;
15
16
17
```

 $\square$  la variable y vaut 5

□ le programme affiche "Faux"

 $\square$  la variable x vaut 5 et la variable y vaut 0

 $\square$  la variable x vaut 0

```
4. Soit la fonction f définie par :
```

```
int f(int a)
  printf("a = \n", %d);
  if (a > 0)
  {
   return f(a - 1) + 1;
  return 4;
```

Alors l'expression f(1) prendra la valeur :

```
\Box 5
```

 $\Box$  1

 $\Box$  4  $\square$  0

5. Pour l'extrait de programme suivant :

```
int produit = 0;
  int serie[4] = \{2, 2, 2, 2\};
  for (i = 0; i < 4; i = i + 1)
    produit = produit * serie[i];
  printf("produit = %d", produit);
La valeur affichée est :
```

```
\Box 0
\Box 16
\square 8
```

 $\Box$  4

6. Pour l'extrait de programme suivant :

```
int i = 0;
int j = 0;
for (i = 0; i < 2; i = i + 1)
    for (j = 0; j < 3; j = j + 1)
        printf("%d ", j);
    }
}
```

qu'est ce qui sera affiché?

```
\Box 0 1 2 0 1 2
\Box 0 1 2 0 1 2 3
\Box 0 1 2 3 0 1 2
\Box 0 0 1 1 2 2 3
```

7. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

```
☐ int char;
□ char 'c';
```

 $\Box$  char c; ☐ char "c";

8. Pour compiler un programme prog.c, on utilise la ligne de commande :

```
☐ gcc -Wall prog.exe -o prog.c
```

$$\hfill\Box$$
 gcc -Wall prog.c -o prog.exe

$$\hfill\Box$$
 gcc prog.exe -Wall -o prog.c

$$\hfill\Box$$
 gcc prog.c -o -Wall prog.exe

9. Pour l'extrait de programme suivant :

```
int i;
int j;
for(i=4;i>0;i=i-1)
  for(j=i;j<6;j=j+1)
  {
    printf("*");
  printf(" ");
```

qu'est ce qui sera affiché?

***	**	**	****	***
** *	**	***	* ***	***
***	* *	***	***	***

10. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :	<pre>13. Si a et b sont deux variables de type :     struct toto_s     {        int n;</pre>	<ul><li>18. Quels calculs peut-on programmer en programmation structurée?</li><li>□ certains programmes sont de vrais plats de spa-</li></ul>
$\square$ dans lequel ces fonctions sont appelées dans le	double x;	ghetti
main  ☐ alphabétique ☐ dans lequel vous avez déclaré ces fonction	}; Alors pour tester l'égalité de a et de b on utilise la condition :	☐ il y a des calculs programmables en programma- tion structurée qui ne sont pas programmables en langage machine
☐ un ordre quelconque	$\square a\{n, x\} == b\{n, x\}$	$\Box$ il y a des calculs programmables en langage ma-
11. Soit la fonction g définie par :	□ a == b	chine et qui ne sont pas programmables en pro-
<pre>int g(int a) {</pre>	□ a = b	grammation structurée
printf("a = \n", %d);	$\square (a.n == b.n) \&\& (a.x == b.x)$	$\Box$ en programmation structurée on peut program-
if (1 > 0) {	14. Quelle étape de la compilation vient d'échouer lors- qu'on a un message comme celui-ci :	mer tous les calculs programmables en langage machine
return 5;	Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »	19. Soit la fonction f définie par :
return 7;	☐ l'analyse sémantique	13. Soft la foliction 1 definie par .
}	☐ l'analyse des entrées clavier	<pre>int f(int a)</pre>
Alors l'expression g(0) prendra la valeur :	☐ l'édition de liens	{
□ 7	☐ l'analyse harmonique	printf("a = \n", %d); if (a > 0)
$\Box$ 5	15. Pour déclarer une fonction saisie_utilisateur qui	{
$\Box$ 0	demande à l'utilisateur d'entrer un entier au clavier et	return 3;
12. Soit le programme principal suivant :	renvoie cet entier on écrit :	}
int main()	☐ int saisie_utilisateur();	return 4;
t int a = 3;	☐ void saisie_utilisateur(int n);	}
int b = 5;	☐ void saisie_utilisateur(char c);	Alors l'expression f(0) prendra la valeur :
printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);	☐ saisie_utilisateur(scanf(%d));	
return EXIT_SUCCESS;	16. Un programme en langage C doit comporter une et une	$\square$ 3
}	seule définition de la fonction :	$\Box$ 0
appelant la fonction f ainsi définie :	$\square$ include	$\Box$ 4
int f(int a, int b)	$\square$ main	
t a = a + b;	$\Box$ init	20. Afin de représenter la taille d'un tableau, définir une
return a;	$\Box$ begin	constante symbolique N valant 3.
}	17. L'écriture $\underline{101}$ en binaire correspond au nombre natu-	☐ #define N 3
L'affichage dans le main est le suivant :	rel:	
$\Box$ f(a,b)=13, a=8, b=5	$\square$ 3	$\square$ #define taille = N
$\Box$ f(3,5)=8, a=3, b=5	$\Box$ 5	☐ #define taille = 3
$\square$ f(a,b)=8, a=3, b=5	□ 101	□ #define N = 3
$\Box$ f(a,b)=8, a=8, b=5	$\Box$ 4	□ #delile M − 2

Prénom:	Nom:
$N^{\circ}$ INE :	

Licence 1 Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\Box$  0  $\square$  3  $\Box$  4 2. Quel est le problème d'un programme comportant les lignes suivantes? while (1) printf("coucou\n"); □ il n'affiche rien  $\square$  il comporte une boucle infinie  $\square$  il risque d'afficher bonjour à la place de coucou  $\square$  il ne compile pas 3. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  int n = carre();  $\square$  n = carre(n);  $\square$  n = carre(int n);  $\square$  int carre(2); 4. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3. ☐ #define taille = N □ #define N 3

☐ #define taille = 3

 $\square$  #define N = 3

5.	Le langage C est un langage
	$\Box$ lu, écrit, parlé
	$\square$ interprété
	□ compilé
	$\square$ composé
6.	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	$\square$ un point-virgule manquant
	$\Box$ un point-virgule en trop
	$\Box$ une accolade manquante
	$\Box$ une accolade en trop
7.	Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?
	$\Box$ une fonction déclarée mais non définie
	$\Box$ une fonction appelée avant sa déclaration
	$\Box$ une directive préprocesseur $\# \mathtt{include}$ man quante
	$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction
8.	On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?
	□ (!A    B)
	☐ (A == TRUE) && (B == TRUE)
	□ A && B
	□ !(!A    B) == (A && !B)
9.	Vous utilisez une boucle while quand :
	$\Box$ l'incrément de la variable de boucle n'est pas $1$
	$\square$ vous n'avez pas déclaré de fonction
	□ vous avez déjà fait un for dans le même programme principal
	□ vous ne connaissez pas le nombre d'itérations de

```
10. Pour déclarer une variable qui sera utilisée comme va-
    riable de boucle on peut utiliser l'instruction
      □ loop i;
      \square int k;
      \square int loop n;
      \square int %d;
11. Si a et b sont deux variables de type:
    struct toto_s
       int n;
       double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
      □ a == b
      \square (a.n == b.n) & (a.x == b.x)
      \square a\{n, x\} == b\{n, x\}
      \Box a = b
12. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
          printf("%d ", i);
     printf("\n");
    affichera:
      \square 0 1 2 3
      \Box 01234
      \Box 43210
      \square 4 3 2 1
13. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 3
      \Box 4
      \square 101
```

 $\Box$  5

<ul> <li>int i; int x=0; for(i=0,i&lt;5,i=i+1) {     x=x+1; }      □ comportent une erreur qui ne sera pas détectée     □ comportent une erreur qui sera détectée au cours     de l'analyse syntaxique     □ comportent une erreur qui sera détectée au cours     de l'édition de lien     □ ne comportent aucune erreur</li> <li>15. Une segmentation fault est une erreur qui survient lorsque:     □ le programme source a été enregistré sur le disque     dur en plusieurs morceaux et l'un d'entre eux ne     peut pas être chargé par le compilateur     □ le programme tente d'accèder à une partie de la     mémoire qui ne lui est pas réservée     □ le programme tente d'afficher des caractères sur     une ligne qui va au delà de la largeur de la fenêtre     du terminal</li> </ul>	□ la division du programme en zones homogènes échoue  16. Avant de faire appel à une fonction il est nécessaire de: □ l'avoir déclarée □ l'avoir déclarée et définie □ l'avoir défini une constante symbolique de la taille de cette fonction  17. Sous unix (ou linux), la commande ls permet de: □ compiler un programme □ afficher la liste de fichiers contenus dans un répertoire □ afficher le contenu d'un fichier texte □ voir des clips musicaux  18. Le code suivant: int i; for (i = 4; i >= 0; i = i - 1) {     printf("%d ", i); } printf("\n");	affichera:    0 1 2 3 4
--	---	-------------------------

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

- 1. Quels calculs peut-on programmer en programmation structurée?
  - $\Box$  certains programmes sont de vrais plats de spaghetti
  - □ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
  - □ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
- 2. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?
  - □ A && B
  - $\square$  !(!A || B) == (A && !B)
  - $\square$  (!A || B)
  - $\square$  (A == TRUE) && (B == TRUE)
- 3. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :
  - $\Box$  1.5
  - $\square$  0.5
  - $\Box$  1
  - $\Box$  0
- 4. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
  - □ char 'c';
  - $\square$  int char;
  - $\square$  char c;
  - □ char "c";

5. Pour l'extrait de programme suivant :

```
int somme = 0;
for (i = 0; i < 5; i = i + 1)
{
   somme = somme + i;
   i = i + 1; /* attention ! */
}</pre>
```

printf("somme = %d",somme);

La valeur de somme affichée est :

- $\Box$  0
- $\Box$  6
- $\square$  15
- $\Box$  10
- 6. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
  - □ void saisie\_utilisateur(int n);
  - ☐ saisie\_utilisateur(scanf(%d));
  - $\hfill\Box$  void saisie\_utilisateur(char c);
  - $\square$  int saisie\_utilisateur();
- 7. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
  - ☐ mkdir TP4
  - □ new TP4
  - $\square$  yppasswd
  - $\square$  kwrite TP4
- 8. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci :

```
Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »
```

- $\hfill \square$  l'analyse des entrées clavier
- $\square$  l'analyse harmonique
- $\Box\,$ l'analyse sémantique
- $\square$  l'édition de liens

9. Le code suivant :

```
int age = 15;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");</pre>
```

#### affichera:

- $\square$  rien
- □ Mineur Majeur
- □ Majeur
- $\square$  Mineur
- 10. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie :
```

int f(int a, int b)
{
 a = a + b;
 return a;

}

L'affichage dans le main est le suivant :

 $\Box$  f(a,b)=8, a=8, b=5

- $\Box$  f(a,b)=13, a=8, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=8, a=3, b=5

11. Un enregistrement permet de grouper plusieurs valeurs	15. Si le code :
dans:	struct tot
□ ses cases	{
$\square$ ses champs	int n;
$\square$ ses chants	double x
$\square$ ses blocs	}; précède la f
12. Le langage C est un langage	début de ma
$\square$ compilé	☐ int to
$\square$ composé	☐ struct
$\square$ lu, écrit, parlé	☐ toto_s
$\square$ interprété	☐ int st
13. Après exécution jusqu'à la ligne 14 du programme C :	☐ toto_s
10 int main() {	16. Le code suiv
11 int $x = 5$ ;	int i;
12	for (i =
13 printf(" x = %d\n", 2);	
14	print }
15 16 }	printf("\
$\Box$ le terminal affiche x = 5	affichera:
□ le terminal affiche "Faux"	
	$\square$ 0 1 2 3
□ le terminal affiche 5	$\square 4321$
$\Box$ le terminal affiche x = 2	$\square 4321$
14. L'écriture $\underline{101}$ en binaire correspond au nombre natu-	17. Si $n$ est une
rel:	à l'utilisateu
$\square$ 5	$\square$ printf
□ 101	□ un déb
$\Box$ 4	□ scanf(
$\square$ 3	$\square$ printf

```
18. Quel est le problème d'un programme comportant les
                                                  lignes suivantes?
co_s
                                                  while (1)
                                                    printf("coucou\n");
fonction main(), alors on peut écrire en
in():
                                                    \square il comporte une boucle infinie
oto.n = 3;
                                                    \square il ne compile pas
 toto_s toto;
 n, x;
                                                    \square il n'affiche rien
truct toto_s = \{3, -1e10\};
                                                    \square il risque d'afficher bonjour à la place de coucou
 struct z = {3, 0.5};
vant:
                                              19. Un programme en langage C doit comporter une et une
                                                  seule définition de la fonction :
0; i < 5; i = i + 1)
                                                    \square main
tf("%d ", i);
                                                    \square init
(n");
                                                    \square include
4
                                                    □ begin
                                              20. Si pgcd est une fonction prenant en entrée deux entiers
                                                  et renvoyant un entier, il est correct d'écrire :
variable entière pour demander sa valeur
                                                    \square n = pgcd(n, 3);
ur, on utilise plutôt :
                                                    \square n = pgcd(int p, int q);
f("Valeur de n ? %g\n", n);
ogueur
                                                    \square int pgcd(2);
("%d", &n);
                                                    \square int n = pgcd();
f("Valeur de n ? %d\n", n);
```

# Éléments d'informatique – contrôle continue

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N}^{\circ} \text{ INE}: \end{array}$ 

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
1. Quelle étape de la compilation vient d'échouer lors-
  qu'on a un message comme celui-ci :
  Undefined symbols : "_prinft" ou
  référence indéfinie vers « prinft »
    □ l'analyse sémantique
    □ l'analyse des entrées clavier
    □ l'édition de liens
    □ l'analyse harmonique
2. Soit la fonction g définie par :
  int g(int a)
  {
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
    \Box 5
    \Box 0
    \square 7
3. Le code suivant :
    int age = 15;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
  affichera:
    □ Majeur
    ☐ Mineur
    □ rien
    □ Mineur
       Majeur
```

```
4. Si x est une variable réelle (de type double) alors
  x = 3/2 lui affecte la valeur :
     \square 1.5
     \square 0.5
     \square 1
     \square 0
5. Un registre du processeur est :
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     \square une unité de calcul spécialisée de l'ordinateur
     □ une gamme de fréquence de fonctionnement du
        processeur
     \square un composant qui contient la liste des fichiers du
        système
6. L'écriture 101 en binaire correspond au nombre natu-
  rel:
     \Box 5
     \square 3
     \square 101
     \Box 4
7. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
```

```
8. Si le code:
    struct toto_s
      int n;
      double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
     \square int toto.n = 3:
     □ struct toto_s toto;
     \square toto_s n, x;
     \square int struct toto_s = {3, -1e10};
     \Box toto_s struct z = {3, 0.5};
 9. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
     \Box 5
     \Box 4
     \Box 1
     \Box 0
10. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<5;i=i+1)
     \square for(i=0;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
```

 $\square$  for(i=1;i<=5;i=i+1)

<ul> <li>11. Un programme en langage C doit comporter une et une seule définition de la fonction :  init  include  main  begin</li> <li>12. Quel est l'opérateur de différence en C : </li></ul>	15. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?  □ (A == TRUE) && (B == TRUE) □ !(!A    B) == (A && !B) □ (!A    B) □ A && B	18. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :  ☐ int pgcd(int x, y); ☐ int pgcd(int x, int x); ☐ void pgcd(int x, int y); ☐ int pgcd(int y, int x); ☐ Vous avez déclaré préalablement un argemble de fonce.
□ <> □ ! □ ≠	<ul><li>16. Une variable booléenne est un variable :</li><li>☐ qui est vraie ou fausse</li><li>☐ à laquelle une valeur vient d'être affectée</li></ul>	19. Vous avez déclaré préalablement un ensemble de fonc- tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre:
□ !=	□ NaN (not a number, qui n'est pas un nombre)	$\Box$ un ordre quel conque
13. Le langage C est un langage  □ interprété	□ jamais nulle □ réelle positive	$\Box$ dans lequel ces fonctions sont appelées dans le main
<ul> <li>□ composé</li> <li>□ compilé</li> <li>□ lu, écrit, parlé</li> <li>14. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :</li> <li>□ double afficher_menu();</li> <li>□ char afficher_menu(printf("menu"));</li> <li>□ int afficher_menu();</li> <li>□ void afficher_menu();</li> <li>□ int afficher_menu(int char);</li> </ul>	17. Une segmentation fault est une erreur qui survient lorsque :  □ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur  □ la division du programme en zones homogènes échoue  □ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal  □ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	<ul> <li>□ alphabétique</li> <li>□ dans lequel vous avez déclaré ces fonction</li> <li>20. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :</li> <li>□ sélectionner entre deux blocs à l'aide d'une condition</li> <li>□ répéter un bloc tant qu'une condition est vérifée</li> <li>□ retourner un bloc</li> <li>□ mettre les blocs en séquence les uns à la suite des autres</li> </ul>

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Sur unix (ou linux), la commande mkdir permet de :  $\square$  créer un fichier texte □ changer de répertoire courant □ ouvrir un fichier texte □ créer un répertoire 2. Le code suivant : int i: for  $(i = 4; i \ge 0; i = i - 1)$ printf("%d ", i); printf("\n"); affichera:  $\square$  1 2 3 4  $\Box \ 4\ 3\ 2\ 1\ 0$  $\square$  4 3 2 1  $\Box 01234$ 3. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n"); printf("Majeur\n"); affichera: □ Mineur □ Majeur □ Mineur Majeur □ rien 4. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme?  $\square$  une fonction appelée avant sa déclaration □ un désaccord entre la déclaration et la définition d'une fonction

 $\square$  une fonction déclarée mais non définie

☐ une directive préprocesseur #include manquante

```
5. Le code suivant :
    int somme = 0:
    int i:
    for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
    printf("%d", somme);
   affichera:
     \Box 1
     \square 0
     \Box 6
     \square 42
6. Le type des réels en C est :
     \square real
     \square int
     □ char
     □ double
7. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ une directive préprocesseur #include manquante
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
     □ une fonction déclarée mais non définie
8. Le code suivant :
    for (i = 1; i < 5; i = i + 1)
    {
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \square 1 2 3 4
     \Box \ 4\ 3\ 2\ 1\ 0
```

 $\square$  4 3 2 1

```
9. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
      \square une accolade manquante
      ☐ un point-virgule en trop
      ☐ un point-virgule manquant
      \square une accolade en trop
10. Après exécution du programme :
       lecture 8 r0
       valeur 3 r1
       mult r1 r0
       valeur 1 r2
       add r2 r0
       ecriture r0 8
       stop
       5
      \square le bus explose
      □ la case mémoire 8 contiendra 0
      □ la case mémoire 8 contiendra 16
     □ le terminal affiche 8
11. Le code suivant :
     int i:
     for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
     printf("\n");
    affichera:
     \Box 01234
     \square 0 1 2 3
     \Box 43210
     \square 4 3 2 1
```

<pre>12. Si racine est une fonction prenant en entrée un réel   et renvoyant la racine carrée de cet réel, et que x est   une variable réelle définie et initialisée, il est incorrect   d'écrire :</pre>	<ul> <li>□ un ordre quelconque</li> <li>□ alphabétique</li> <li>15. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :</li> <li>□ printf("Valeur de n ? %d\n", n);</li> <li>□ scanf("%d", &amp;n);</li> </ul>	<pre>if (a &gt; 0)     {       return f(a - 1) + 1;     }     return 4; } Alors l'expression f(1) prendra la valeur :</pre>
<pre>□ x = racine(2/3); 13. Au début de la fonction main() on place le code :     char i;     for (i = 'A'; i &lt;= 'F'; i = i + 1)     {</pre>	☐ printf("Valeur de n ? %g\n", n); ☐ un débogueur  16. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :	<ul><li>□ 5</li><li>□ 0</li><li>□ 1</li><li>□ 4</li></ul>
<pre>printf("%c", i); } printf("\n");</pre>	<ul><li>□ void afficher_date(struct date_s d);</li><li>□ int afficher_date(date_s d);</li><li>□ void afficher_date(date_s d);</li></ul>	<ul><li>19. Un enregistrement permet de grouper plusieurs valeurs dans :</li><li>□ ses cases</li></ul>
Alors l'affichage sera :    A  ABCDEF  cccccc	<ul> <li>□ struct date_s afficher_date(struct date_s</li> <li>17. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :</li> <li>□ analyse sémantique</li> </ul>	
14. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :  □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main	□ analyse harmonique □ analyse syntaxique □ analyse lexicale  18. Soit la fonction f définie par : int f(int a) {     printf("a = \n", %d);	<ul> <li>20. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?</li> <li>□ int char;</li> <li>□ char 'c';</li> <li>□ char c;</li> <li>□ char "c";</li> </ul>

# Éléments d'informatique – contrôle continue

Pránam ·	Nom .
Prenom:	NOIII:
ATO TATES	
N° INE ·	
11 11/12.	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int i:
    for (i = 4; i \ge 0; i = i - 1)
        printf("%d ", i);
    printf("\n");
  affichera:
    \Box 43210
    \Box 01234
    \Box 1234
    \square 4 3 2 1
2. Dans la commande gcc, l'option -Wall signifie :
     ☐ que l'on veut voir tous les avertissements
     □ qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
3. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
     □ une fonction déclarée mais non définie
     □ une directive préprocesseur #include manquante
     ☐ une fonction appelée avant sa déclaration
     □ un désaccord entre la déclaration et la définition
       d'une fonction
4. Si cette erreur apparaît à la compilation :
  error: expected ';' before '}' token que doit-
  on chercher dans le programme?
     \square une accolade en trop
    ☐ un point-virgule manquant
     \square une accolade manquante
     □ un point-virgule en trop
```

```
5. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
     \square en parallèle, chacun dans un registre
     \square tous ensemble
     □ tour à tour, un petit peu à chaque fois
     □ chacun son tour, après que le processus précédent
        a terminé
6. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera:
     □b
     \sqcap C
     \square A
     □ B
7. Si le code :
   struct toto s
   {
     int n;
     double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
    \Box int struct toto_s = {3, -1e10};
     \square int toto.n = 3;
     \square toto_s n, x;
     □ struct toto_s toto;
     \square toto_s struct z = {3, 0.5};
8. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
    printf("\n");
```

```
affichera:
     \square 0 2 4 6
     \Box 0123456
     \Box 02468
     \Box 0 1 2 3 4 5 6 7
 9. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ char "c";
     ☐ int char:
     □ char c;
     □ char 'c';
10. Un bit est:
     □ un battement d'horloge processeur
      ☐ l'instruction qui met fin à un programme
     \square un chiffre binaire (0 ou 1)
      □ la longueur d'un mot mémoire
11. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
    Quelle est la condition cond:

    a <= n <= b
</p>
      □ (a<=n) && (n<=b)
      □ (n<=a) && (n<=b)
      □ (a<n) || (n>b)
```

```
15. Pour déclarer un tableau d'entiers de taille 5, on peut
12. Après exécution du programme :
                                                                                                                            19. Soit la fonction f définie par :
                                                                  utiliser l'instruction
       lecture 8 r0
                                                                                                                                int f(int a)
                                                                    \Box int tab[] = 5;
       valeur 3 r1
                                                                   \Box int toto[5];
       mult r1 r0
                                                                                                                                  printf("a = \n", %d);
                                                                   ☐ int toto[taille=5];
                                                                                                                                  if (a > 0)
       valeur 1 r2
                                                                   ☐ int[] new tableau(5);
                                                                                                                                  {
  5
       add r2 r0
                                                                    □ char tableau[5];
                                                                                                                                    return 3;
       ecriture r0 8
                                                             16. Un enregistrement permet de grouper plusieurs valeurs
       stop
                                                                                                                                  return 4;
                                                                  dans:
                                                                    \square ses chants
      \square la case mémoire 8 contiendra 0
                                                                    \square ses blocs
                                                                                                                                Alors l'expression f(0) prendra la valeur :
                                                                   \square ses champs
      \Box le terminal affiche 8
                                                                                                                                  \Box 4
                                                                    \square ses cases
      \square le bus explose
                                                             17. Si a et b sont deux variables de type:
                                                                                                                                 \Box 0
      □ la case mémoire 8 contiendra 16
                                                                  struct toto_s
                                                                                                                                  \square 3
13. Quel est l'opérateur de différence en C :
                                                                                                                            20. Les lignes
                                                                    int n;
                                                                    double x;
      □ !=
                                                                                                                                int i;
                                                                 };
                                                                                                                                int x=0;
      □!
                                                                  Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                                for(i=0,i<5,i=i+1)
      □ <>
                                                                  condition:
                                                                                                                               {
                                                                   \Box a = b
                                                                                                                                  x=x+1;
      \square \neq
                                                                   \square a\{n, x\} == b\{n, x\}
                                                                                                                               }
14. Une variable booléenne est un variable :
                                                                   \square (a.n == b.n) && (a.x == b.x)
                                                                                                                                  □ comportent une erreur qui sera détectée au cours
                                                                    □ a == b
      □ NaN (not a number, qui n'est pas un nombre)
                                                                                                                                    de l'analyse syntaxique
                                                             18. Après la déclaration : int mccarthy(int n);, il est
      □ à laquelle une valeur vient d'être affectée
                                                                  correct d'écrire :
                                                                                                                                  \square ne comportent aucune erreur
                                                                   \square n = mccarthy();
      ☐ jamais nulle
                                                                                                                                  □ comportent une erreur qui ne sera pas détectée
                                                                   \square n = mccarthy(p, q);
      □ réelle positive
                                                                                                                                 □ comportent une erreur qui sera détectée au cours
                                                                   \Box int mccarthy(int 2);
                                                                                                                                     de l'édition de lien
      □ qui est vraie ou fausse
                                                                    \square x = mccarthy(n);
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise: ☐ int afficher\_menu(int char); ☐ double afficher\_menu(); ☐ int afficher\_menu(); □ void afficher\_menu(); ☐ char afficher\_menu(printf("menu")); 2. Un bit est: □ la longueur d'un mot mémoire ☐ l'instruction qui met fin à un programme  $\square$  un battement d'horloge processeur  $\square$  un chiffre binaire (0 ou 1) 3. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :  $\square$  printf("x=%d et y=%d\n",x y);  $\square$  printf("x=%d et y=%d\n,x,y");  $\square$  printf("x=%x et y=%y\n");  $\square$  printf("x=%d et y=%d\n",x,y); 4. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; i < 0; i = i + 1)for (j = 0; j < 5; j = j + 1) $printf("j = %d\n", j);$ qu'est ce qui sera affiché?  $\Box$  i = 0  $\Box$  j = %d  $\Box$  j = 5  $\Box$  j = 4

```
5. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
  d'écrire :
    \square x = racine(racine(x)*racine(x)):
    \square x = racine(x * x) - racine(x);
    \square x = racine(2/3):
    \square x - 1 = racine(x);
6. Pour l'extrait de programme suivant :
     int somme = 0;
     int serie[4] = \{2, 4, 10, 4\};
     for (i = 0; i < 4; i = i + 1)
     {
       somme = somme + serie[i];
     printf("somme = %d",somme);
   La valeur de somme affichée est :
    \Box 16
    \Box 6
     \square 20
     \square 3
7. Soit le programme principal suivant :
   int main()
    int a = 3:
    int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
```

 $\Box$  f(a,b)=13, a=8, b=5

```
8. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 4
     \square 101
     \Box 5
     \square 3
 9. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
     \square n = factorielle();
     ☐ int factorielle(int 2);
     ☐ printf("%d", factorielle(n));
     \square n = factorielle(p, q);
10. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
     \square int pgcd(2);
     \square n = pgcd(n, 3);
     \square int n = pgcd();
     \square n = pgcd(int p, int q);
11. Sous unix (ou linux), la commande cd permet de :
      □ jouer de la musique
      □ changer de répertoire courant
     ☐ récupérer un programme arrêté avec la commande
         ab
     □ ouvir un bureau partagé (common desktop)
      □ détruire un fichier
12. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     else
          printf("Majeur\n");
    affichera:
```

□ Majeur	16. Soit la fonction g définie par :	□ il n'affiche rien
□ rien	int g(int a)	☐ il risque d'afficher bonjour à la place de coucou
☐ Mineur	{ printf("a = \n", %d);	☐ il comporte une boucle infinie
Majeur □ Mineur	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	☐ il ne compile pas
13. Laquelle de ces écritures correspond à la déclaration	{	□ if the complie pas
d'une variable de type caractère en langage C?	return 5;	19. Soit la fonction f définie par :
☐ char 'c';	return 7;	int f(int a)
□ char c;	}	<pre>int f(int a) {</pre>
☐ int char; ☐ char "c";	Alors l'expression g(0) prendra la valeur :	printf("a = \n", %d);
□ cnar "c";  14. Quelle étape de la compilation vient d'échouer lors-		if (a > 0)
qu'on a un message comme celui-ci :	$\Box$ 5 $\Box$ 0	treturn 3;
Undefined symbols :"_prinft" ou	17. Après exécution jusqu'à la ligne 15 du programme C:	}
référence indéfinie vers « prinft » □ l'analyse harmonique	10	return 4;
☐ l'édition de liens	11 int main() {	}
☐ l'analyse sémantique	12 int x = 5;	Alors l'expression f(0) prendra la valeur :
☐ l'analyse des entrées clavier	13 14	□ 0
15. Après exécution du programme :	15	
1 lecture 8 r0	16 17 }	$\square$ 4
2 valeur 3 r1 3 mult r1 r0		
4 valeur 1 r2	☐ le programme affiche **** ☐ la variable x vaut 16	20. On considère deux variables booléennes A et B initia-
5 add r2 r0	$\Box$ la variable x vaut $-\frac{1}{2}$	lisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur
6 ecriture r0 8 7 stop	☐ le programme affiche x	TRUE?
8 5	18. Quel est le problème d'un programme comportant les	☐ (A == TRUE) && (B == TRUE)
$\square$ le terminal affiche 8	lignes suivantes?	□ A && B
$\Box$ la case mémoire 8 contiendra 16	while (1)	
$\Box$ la case mémoire 8 contiendra 0	{     printf("coucou\n");	☐ !(!A    B) == (A && !B)
$\Box$ le bus explose	}	□ (!A    B)

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Après exécution jusqu'à la ligne 15 du programme C: int main() { 11 int x = 5; 12 int y = 3; 13 14 x = y;15 16 } 17 ☐ le programme affiche "Faux"  $\square$  la variable y vaut 5  $\square$  la variable x vaut 5 et la variable y vaut 3  $\square$  la variable x vaut 3 2. Un bit est: ☐ l'instruction qui met fin à un programme □ la longueur d'un mot mémoire □ un battement d'horloge processeur  $\square$  un chiffre binaire (0 ou 1) 3. Le type des réels en C est :  $\square$  int  $\square$  real ☐ double □ char 4. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ dans lequel vous avez déclaré ces fonction  $\square$  un ordre quelconque □ alphabétique □ dans lequel ces fonctions sont appelées dans le main

```
5. La virtualisation de la mémoire permet notamment de
   stocker des portions inactives de la mémoire de travail
  sur le disque dur. Mais on perd :
     □ en temps d'accès
     □ certaines données de la mémoire de travail
     □ les fichiers du disque
     \square des processus
6. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
     □ struct int factorielle(int n);
     ☐ int factorielle(double n):
     ☐ int factorielle();
     \Box int factorielle(int x);
7. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \square 0.5
     \Box 1
     \Box 0
8. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     \square ses chants
     \square ses blocs
     \square ses champs
9. Sous unix (ou linux), la commande cd permet de :
     □ détruire un fichier
     □ ouvir un bureau partagé (common desktop)
     ☐ récupérer un programme arrêté avec la commande
        ab
     □ changer de répertoire courant
```

□ jouer de la musique

10. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  $\mathbf{n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant : int a = 0; int b = 20; int n; scanf("%d", &n); while(cond) scanf("%d", &n); Quelle est la condition cond :  $\square$  (a<n) || (n>b) ☐ (a<=n) && (n<=b) □ a<=n<=b □ (n<=a) && (n<=b) 11. Le bus système sert à : □ transporter les processus du tourniquet au processeur ☐ Écrire des données sur le dique dur ☐ Transférer des données et intructions entre processeur et mémoire ☐ Arriver à l'heure en cours 12. Un programme en langage C doit comporter une et une seule définition de la fonction : □ main  $\square$  init □ begin  $\square$  include 13. Pour l'extrait de programme suivant : int somme = 0; int  $serie[4] = \{2, 4, 10, 4\};$ for (i = 0; i < 4; i = i + 1)somme = somme + serie[i];

printf("somme = %d",somme);

La valeur de somme affichée est :

```
\square 20
      \Box 16
      \square 3
      \Box 6
14. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
      □ une directive préprocesseur #include manquante
      \square un désaccord entre la déclaration et la définition
         d'une fonction
     ☐ une fonction appelée avant sa déclaration
      \square une fonction déclarée mais non définie
15. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
   affichera:
      \square rien
      ☐ Mineur
      ☐ Mineur
        Majeur
      □ Majeur
```

```
16. Après exécution jusqu'à la ligne 15 du programme C :
  10
       int main() {
  11
  12
            int x = 5;
  13
  14
            x = 3 * x + 1;
  15
  16
             . . .
       }
  17
      □ la variable x vaut 16
      ☐ le programme affiche ****
      \Box la variable x vaut -\frac{1}{2}
      \Box le programme affiche x
17. Si a et b sont deux variables de type:
    struct toto_s
      int n;
      double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
      □ a == b
      \Box a = b
      \square (a.n == b.n) && (a.x == b.x)
      \square a{n, x} == b{n, x}
18. Sous unix (ou linux), la commande 1s permet de :
      \square afficher le contenu d'un fichier texte
      □ compiler un programme
      □ voir des clips musicaux
      \square afficher la liste de fichiers contenus dans un
         répertoire
```

```
19. Soit la fonction g définie par :
   int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
      return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 0
     \Box 7
     \Box 5
20. Si le code:
   struct toto_s
      int n;
      double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
     \square toto_s struct z = {3, 0.5};
     \square toto_s n, x;
     \square int struct toto_s = {3, -1e10};
     \square int toto.n = 3;
```

□ struct toto\_s toto;

□ A

## Éléments d'informatique – contrôle continue

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

	opense masser 2 aree v 20 mmates.
1.	Un enregistrement permet de grouper plusieurs valeurs dans :
	□ ses chants
	□ ses champs
	□ ses blocs
	□ ses cases
2.	Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
	$\Box$ dans lequel ces fonctions sont appelées dans le main
	$\hfill\Box$ un ordre quel conque
	$\Box$ alphabétique
	$\Box$ dans lequel vous avez déclaré ces fonction
3.	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	$\square$ analyse syntaxique
	$\hfill\Box$ analyse harmonique
	$\square$ analyse lexicale
	$\Box$ analyse sémantique
4.	Au début de la fonction main() on place le code :
	<pre>char b = 'A'; b = b + 2; printf("%c\n", b);</pre>
	Alors l'affichage sera :
	□В
	□ b
	□ С

```
5. Au début de la fonction main() on place le code :
   for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
   printf("\n");
  Alors l'affichage sera :
    Пі
    □ cccccc
    □ A
    ☐ ABCDEF
6. On souhaite faire une boucle de contrôle de saisie : tant
  que l'entier n n'appartient pas à l'intervalle [a..b], on
  recommence la saisie de n. Soit le programme suivant :
   int a = 0;
   int b = 20;
   int n;
   scanf("%d", &n);
   while(cond)
      scanf("%d", &n);
   Quelle est la condition cond :
    □ a<=n<=b
    \square (a<n) || (n>b)
    \square (a<=n) && (n<=b)
    \square (n<=a) && (n<=b)
7. Pour déclarer une procédure afficher_date qui prend
  en argument un struct date_s et affiche le contenu
  du struct, on écrit :
    □ void afficher_date(date_s d);
    □ void afficher_date(struct date_s d);
```

☐ int afficher\_date(date\_s d);

☐ struct date\_s afficher\_date(struct date\_s d);

```
8. Si cette erreur apparaît à la compilation :
    Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft » que doit-
   on chercher dans le programme?
     □ une directive préprocesseur #include manquante
      ☐ une faute de frappe dans un appel de fonction
      \square une variable non déclarée
      \square un caractère interdit en C
 9. Après exécution jusqu'à la ligne 14 du programme C :
      int main() {
 11
            int x = 5;
 12
           printf(" x = %d\n", 2);
 13
 14
 15
      }
 16
     \square le terminal affiche x = 5
     □ le terminal affiche "Faux"
     □ le terminal affiche 5
      \square le terminal affiche x = 2
10. Après exécution jusqu'à la ligne 15 du programme C :
 10
 11
       int main() {
 12
            int x = 5;
 13
 14
           x = 3 * x + 1:
 15
 16
      }
 17
     \square la variable x vaut -\frac{1}{2}
      \square le programme affiche x
      □ la variable x vaut 16
      ☐ le programme affiche ****
11. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
      ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
      □ void afficher_menu();
      ☐ double afficher_menu();
```

☐ char afficher\_menu(printf("menu"));

main	12. Un programme en langage C doit comporter une et une	}	15
main	seule définition de la fonction :	printf("j = %d\n", j);	
init   begin   j = 0   j = 5   la variable x vaut 3   le programme affiche "Faux"   la variable x vaut 3   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme affiche "Faux"   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le variable be obleenne est un variable : la vous avez déjà fait un for dans le même programme principal   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le produit = vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   la variable y vaut 5   le programme suivant : int produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le produit = vous ne connaissez pas le nombre d'itérations de la variable y vaut s' le produit = 0; int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)   le produit = variable y vaut s' le produit = vint p	$\Box$ include		17 }
begin     j = 5	$\Box$ main	qu'est ce qui sera affiché par ce printf?	$\square$ la variable x vaut 5 et la variable y vaut 3
13. Avant de faire appel à une fonction il est nécessaire de :	$\Box$ init	□ j = 0	$\Box$ la variable x vaut 3
de :   j = 4   16. Vous utilisez une boucle while quand :     j = 4   16. Vous utilisez une boucle while quand :	□ begin	□ j = 5	☐ le programme affiche "Faux"
de :	13. Avant de faire appel à une fonction il est nécessaire	□ j = %d	☐ la variable y vaut 5
Tavoir déclarée   avoir défini une constante symbolique de la taille de cette fonction   Tavoir déclarée et définie   vous n'avez pas déclaré de fonction   Tavoir déclarée et définie   I'avoir définie   vous avez déjà fait un for dans le même programme principal   vous avez déjà fait un for dans le même programme principal   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne vous ne varez déjà fait un for dans le même programme principal   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne varez déjà fait un for dans le même programme principal   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'avance   vous ne connaissez pas le nombre d'itérations de la boucle a l'a		□ j = 4	, , , , , , , , , , , , , , , , , , ,
avoir défini une constante symbolique de la taille de cette fonction   l'avoir définie   l'avoir dé	☐ l'avoir déclarée	16. Vous utilisez une boucle while quand:	• •
de cette fonction    l'avoir déclarée et définie   vous avez déjà fait un for dans le même programme principal   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous une connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à l'avance   vous ne connaissez pas le nombre d'itérations de la boucle à	$\square$ avoir défini une constante symbolique de la taille	•	
□ l'avoir déclarée et définie □ l'avoir d'étre affectée □ réelle positive □ l'a vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ l'avoil explositif ("produit = %d", produit); □ La valeur affichée est : □ l'a valeur affichée est : □ l'avaleur affichée est : □ l'a valeur affichée est : □ l'avaleur affichée est : □ l'avale	de cette fonction	•	
l'avoir définie   gramme principal   produit = produit * serie[i]; }   14. Une variable booléenne est un variable :	$\square$ l'avoir déclarée et définie		
à laquelle une valeur vient d'être affectée   la boucle à l'avance   la valeur affichée est :   la vale	☐ l'avoir définie	v -	
à laquelle une valeur vient d'être affectée   la boucle à l'avance   la valeur affichée est :   la vale	14. Une variable booléenne est un variable :	□ vous ne connaissez pas le nombre d'itérations de	}
<ul> <li>□ réelle positive</li> <li>□ jamais nulle</li> <li>□ NaN (not a number, qui n'est pas un nombre)</li> <li>□ qui est vraie ou fausse</li> <li>15. Soit un programme contenant les lignes suivantes :</li> <li>int i = 0;</li> <li>int j = 0;</li> <li>for (i = 0; i &lt; 3; i = i + 1)</li> <li>{</li> <li>for (j = 0; j &lt; 5; j = j + 1)</li> <li>for (j = 0; j &lt; 5; j = j + 1)</li> <li>for (j = 0; j &lt; 5; j = j + 1)</li> <li>int x = 5;</li> <li>int</li></ul>	□ à laquelle une valeur vient d'être affectée	la boucle à l'avance	<pre>printf("produit = %d", produit);</pre>
□ jamais nulle □ NaN (not a number, qui n'est pas un nombre) □ qui est vraie ou fausse  15. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; j < 3; i = i + 1) { for (j = 0; j < 5; j = j + 1) {		17. Si cette erreur apparaît à la compilation:	La valeur affichée est :
<pre>NaN (not a number, qui n'est pas un nombre)</pre>	-		□ 8
□ qui est vraie ou fausse  15. Soit un programme contenant les lignes suivantes :  int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1) {  for (j = 0; j < 5; j = j + 1) {  int x = 5; }  int x = 5; }  int i = 0;  une accolade manquante □ une accolade en trop  18. Après exécution jusqu'à la ligne 15 du programme C :  une accolade en trop  18. Après exécution jusqu'à la ligne 15 du programme C :  une accolade en trop  18. Après exécution jusqu'à la ligne 15 du programme C :  une accolade en trop  19. Pour déclarer une fonction exposant qui prend er gument un réel x et un entier positif n et renvoir valeur de x^n on écrit :  une accolade manquante □ une accolade en trop  18. Après exécution jusqu'à la ligne 15 du programme C :  une accolade manquante □ une accolade en trop  19. Pour déclarer une fonction exposant qui prend er gument un réel x et un entier positif n et renvoir valeur de x^n on écrit : □ void exposant(double x^n); □ int exposant(double n, int x); □ double exposant(double x, int n); □ appraire (double x, int n);	·	on chercher dans le programme?	$\square$ 4
15. Soit un programme contenant les lignes suivantes :  int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1) {  for (j = 0; j < 5; j = j + 1) {  int x = 5; }  int x = 5;  int y = 3;  int point-virgule en trop  une accolade manquante  une accolade en trop  18. Après exécution jusqu'à la ligne 15 du programme C :  une accolade manquante  gument un réel x et un entier positif n et renvoir valeur de x <sup>n</sup> on écrit :  valeur de x <sup>n</sup> on écrit :  valeur de x <sup>n</sup> on écrit :  une accolade manquante  int i = 0; int main() {  int x = 5; int exposant(double x, int n);  double exposant(double x, int n);  une accolade manquante  int i = 0; int int x = 5; int exposant(double x, int n);  une accolade manquante  int i = 0; i	, , ,	$\square$ un point-virgule manquant	
int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1) {  for (j = 0; j < 5; j = j + 1) {  int x = 5; {  int x = 5; }  int x = 5; }  int x = 5;  int x =		$\square$ un point-virgule en trop	□ 16
Int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1) {  for (j = 0; j < 5; j = j + 1) {  int x = 5; {  int y = 3;  int i = 0; int j = 0;  int j = 0;  int j = 0;  int i = 0;  int	15. Soit un programme contenant les lignes suivantes :	$\hfill\Box$ une accolade manquante	
18. Après exécution jusqu'à la ligne 15 du programme C:  {  for (j = 0; j < 5; j = j + 1)  {  for (j = 0; j < 5; j = j + 1)  {  int $x = 5$ ;  int $y = 3$ ; $y$ waleur de $x^n$ on écrit : $y$ valeur de $x^n$ on écrit : $y$ valeur de $x^n$ on écrit : $y$ $y$ $y$ $y$ $y$ $y$ $y$ $y$	· ·	$\Box$ une accolade en trop	
10	<b>5</b> ,	18. Après exécution jusqu'à la ligne 15 du programme C:	
for $(j = 0; j < 5; j = j + 1)$ 11    int $x = 5;$ 12    int $y = 3;$ 13    double exposant(double x, int n);	for (1 = 0; 1 < 3; 1 = 1 + 1)	- v - v - v	□ void exposant(double x^n):
{	for (i = 0: i < 5: i = i + 1)		<u>-</u>
$\Box$	{	· ·	-
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		13	•
· · · · · · · · · · · · · · · · · · ·	}	14   x = y;	$\square$ exposant(double x, int n, int r);

# Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si a et b sont deux variables de type : struct toto\_s { int n; double x; }; Alors pour tester l'égalité de a et de b on utilise la condition: □ a == b  $\square (a.n == b.n) & (a.x == b.x)$  $\Box$  a = b  $\square$  a{n, x} == b{n, x} 2. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :  $\square$  int factorielle(int x); ☐ int factorielle(double n); ☐ int factorielle(); □ struct int factorielle(int n); 3. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n");  $\square$  il ne compile pas □ il risque d'afficher bonjour à la place de coucou  $\square$  il comporte une boucle infinie □ il n'affiche rien 4. Si cette erreur apparaît à la compilation : Undefined symbols : "\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme? □ une variable non déclarée □ un caractère interdit en C  $\square$  une faute de frappe dans un appel de fonction  $\square$  une directive préprocesseur #include manquante

```
5. Le code suivant :
   int i:
   for (i = 0; i < 5; i = i + 1)
        printf("%d ", i);
   printf("\n");
  affichera:
    \Box 0123
    \Box 01234
    \square 4 3 2 1
    \Box 4 3 2 1 0
6. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
    ☐ une directive préprocesseur #include manquante
    □ un désaccord entre la déclaration et la définition
       d'une fonction
    ☐ une fonction appelée avant sa déclaration
    □ une fonction déclarée mais non définie
7. Sur unix (ou linux), la commande mkdir permet de :
    □ changer de répertoire courant
    □ créer un fichier texte
    □ ouvrir un fichier texte
    □ créer un répertoire
8. Après exécution du programme :
     lecture 8 r0
      valeur 3 r1
      mult r1 r0
      valeur 1 r2
      add r2 r0
      ecriture r0 8
      stop
    □ la case mémoire 8 contiendra 0
    \square le bus explose
    □ le terminal affiche 8
    □ la case mémoire 8 contiendra 16
```

```
9. Le type des réels en C est :
     □ char
     □ int
     □ real
     □ double
10. Si le code:
    struct toto_s
      int n;
      double x:
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
     \square int toto.n = 3:
     \Box toto_s struct z = {3, 0.5};
      □ struct toto s toto:
     \square int struct toto_s = {3, -1e10};
     \square toto_s n, x;
11. Pour l'extrait de programme suivant :
     int i = 0:
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
              printf("%d ", i);
         }
     printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 0 1 1 2 2
     \Box 0 1 2 0 1 2
     \Box 1 2 3 1 2
     \Box 0 1 0 1 0 1
```

```
12. Le code suivant :
                                                                 {
                                                                   scanf("%d", &n);
     int i;
     for (i = 8; i > 0; i = i - 2)
                                                                Quelle est la condition cond:
         printf("%d ", i);
                                                                  □ a<=n<=b
                                                                  □ (n<=a) && (n<=b)
     printf("\n");
                                                                  \square (a<n) || (n>b)
   affichera:
                                                                  □ (a<=n) && (n<=b)
     \Box 8 6 4 2 0
                                                            15. Vous utilisez une boucle while quand :
     \Box 8 6 4 2
                                                                  ☐ l'incrément de la variable de boucle n'est pas 1
     \Box 02468
                                                                  \square vous ne connaissez pas le nombre d'itérations de
     \square 8 2
                                                                     la boucle à l'avance
13. Soit la fonction f définie par :
                                                                  □ vous n'avez pas déclaré de fonction
                                                                  □ vous avez déjà fait un for dans le même pro-
   int f(int a)
                                                                     gramme principal
      printf("a = \n", %d);
                                                            16. Au début de la fonction main() on place le code :
      if (a > 0)
                                                                 for (i = 'A'; i \le 'F'; i = i + 1)
        return f(a - 1) + 1;
                                                                   printf("%c", i);
      return 4;
                                                                 printf("\n");
   Alors l'expression f(1) prendra la valeur :
                                                                Alors l'affichage sera:
     \Box 1
                                                                  □ cccccc
     \Box 4
                                                                  □ A
     \Box 0
                                                                  ☐ ABCDEF
      \Box 5
                                                                  \Box i
14. On souhaite faire une boucle de contrôle de saisie : tant
                                                            17. Soient deux variables entières x et y initialisées à 4 et
   que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                5 respectivement. L'affichage x=4 et y=5 est obtenu
   recommence la saisie de n. Soit le programme suivant :
                                                                avec la commande :
                                                                  \Box printf("x=%x et y=%y\n");
     int a = 0;
     int b = 20;
                                                                  \square printf("x=%d et y=%d\n",x y);
     int n:
                                                                  \square printf("x=%d et y=%d\n,x,y");
     scanf("%d", &n);
                                                                  \square printf("x=%d et y=%d\n",x,y);
     while(cond)
```

```
18. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
   Alors l'expression f(0) prendra la valeur :
      \Box 0
      \Box 4
      \square 3
19. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
      \square for(i=1;i<5;i=i+1)
      \square for(i=0;i<5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
20. Après exécution jusqu'à la ligne 15 du programme C:
 10
       int main() {
 11
            int x = 5;
 12
            int y = 3;
 13
 14
            x = y;
 15
 16
 17
      }
      \square la variable x vaut 3
     □ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 3
```

 $\square$  la variable y vaut 5

Prénom:	Nom:
N° INE :	

Licence 1 Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si a et b sont deux variables de type : struct toto\_s { int n; double x; }; Alors pour tester l'égalité de a et de b on utilise la condition:  $\square (a.n == b.n) & (a.x == b.x)$ □ a == b  $\Box$  a = b  $\square$  a{n, x} == b{n, x} 2. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n");  $\square$  il ne compile pas □ il n'affiche rien □ il comporte une boucle infinie  $\square$  il risque d'afficher bonjour à la place de coucou 3. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :  $\square$  des processus □ certaines données de la mémoire de travail  $\square$  les fichiers du disque □ en temps d'accès 4. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :

 $\square$  0.5

 $\square$  1.5

 $\Box$  1

 $\Box$  0

```
5. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
    \square n = pgcd(int p, int q);
     \square n = pgcd(n, 3);
     \square int pgcd(2);
    \square int n = pgcd();
6. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
  d'écrire :
    \square n = carre(int n);
    \square n = carre(n);
     \square int n = carre();
     \square int carre(2);
7. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
  mande:
     ☐ kwrite TP4
     □ new TP4
     □ mkdir TP4
     ☐ yppasswd
8. Soit la fonction f définie par :
   int f(int a)
  ₹
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 0
     \Box 4
```

```
9. Le langage C est un langage
     □ interprété
     □ composé
     □ compilé
     □ lu, écrit, parlé
10. Le code suivant :
     int i;
    for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 8 2
     \Box 8 6 4 2 0
     \square 8 6 4 2
     \Box 02468
11. Le code suivant :
     int age = 15;
     if (age < 18)
         printf("Mineur\n");
    }
     else
    {
         printf("Majeur\n");
   affichera:
     □ Mineur
     □ Majeur
     \square rien
     ☐ Mineur
        Majeur
```

```
12. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box j = %d
     \Box j = 5
     \Box j = 4
     \Box j = 0
13. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
     □ A && B
     \square (A == TRUE) && (B == TRUE)
      \square !(!A || B) == (A && !B)
      \square (!A | | B)
14. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
      ☐ char "c";
     \Box char c;
      \square int char;
      □ char 'c';
```

```
15. Pour afficher à l'aide de printf("%d\n",tab[i]);
                                                           18. Un enregistrement permet de grouper plusieurs valeurs
   le contenu d'un tableau de 5 entiers initialisé au
                                                               dans:
   préalable, on utilise plutôt :
                                                                 \square ses champs
     \square for(i=0;i<=5;i=i+1)
                                                                 \square ses blocs
     \square for(i=1;i<=5;i=i+1)
                                                                 □ ses chants
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
                                                                 \square ses cases
16. Pour l'extrait de programme suivant :
                                                           19. Quels calculs peut-on programmer en programmation
    int i = 0;
                                                               structurée?
    int i = 0;
                                                                 □ en programmation structurée on peut program-
    for (i = 0; i < 3; i = i + 1)
                                                                    mer tous les calculs programmables en langage
                                                                    machine
         for (j = 0; j < 2; j = j + 1)
                                                                 ☐ il y a des calculs programmables en programma-
              printf("%d ", i);
                                                                    tion structurée qui ne sont pas programmables en
         }
                                                                    langage machine
    }
                                                                 \square il y a des calculs programmables en langage ma-
    printf("\n");
                                                                    chine et qui ne sont pas programmables en pro-
   qu'est ce qui sera affiché?
                                                                    grammation structurée
     \Box 0 0 1 1 2 2
                                                                 □ certains programmes sont de vrais plats de spa-
     \Box 1 2 3 1 2
     \Box 0 1 2 0 1 2
                                                           20. Si factorielle est une fonction prenant en entrée un
     \Box 0 1 0 1 0 1
                                                                entier et renvoyant un entier, il est correct d'écrire :
17. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
                                                                 \square n = factorielle();
     □ un débogueur
                                                                 ☐ printf("%d", factorielle(n));
     □ printf("Valeur de n ? %d\n", n);
                                                                 \square n = factorielle(p, q);
     □ printf("Valeur de n ? %g\n", n);
                                                                 ☐ int factorielle(int 2);
     □ scanf("%d", &n);
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
i iciioiii .	Nom .
Nº INF ·	
11 111111111111111111111111111111111111	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le bus système sert à : ☐ Écrire des données sur le dique dur ☐ Arriver à l'heure en cours □ Transférer des données et intructions entre processeur et mémoire □ transporter les processus du tourniquet au processeur 2. Si cette erreur apparaît à la compilation : Undefined symbols : "\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme? □ un caractère interdit en C ☐ une directive préprocesseur #include manquante  $\square$  une faute de frappe dans un appel de fonction  $\square$  une variable non déclarée 3. Le langage C est un langage □ lu, écrit, parlé □ composé □ compilé □ interprété 4. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :  $\square$  int pgcd(2);  $\square$  n = pgcd(n, 3);  $\square$  n = pgcd(int p, int q);  $\square$  int n = pgcd(); 5. Vous utilisez une boucle while quand: □ vous avez déjà fait un for dans le même programme principal ☐ l'incrément de la variable de boucle n'est pas 1 □ vous n'avez pas déclaré de fonction □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance

```
6. Pour déclarer une fonction saisie_utilisateur qui
  demande à l'utilisateur d'entrer un entier au clavier et
  renvoie cet entier on écrit :
    ☐ int saisie_utilisateur();
    □ void saisie_utilisateur(char c);
    □ saisie_utilisateur(scanf(%d));
    □ void saisie_utilisateur(int n);
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
    □ ses chants
    \square ses blocs
    \square ses champs
    \square ses cases
8. Le code suivant :
   int age = 18;
   if (age < 18)
   {
        printf("Mineur\n"):
   }
   else
        printf("Majeur\n");
   }
  affichera:
    □ Mineur
    ☐ Mineur
       Majeur
    □ Majeur
    □ rien
9. Si le code :
  struct toto_s
     int n;
     double x;
  };
  précède la fonction main(), alors on peut écrire en
```

début de main() :

```
\square int toto.n = 3:
      □ struct toto_s toto;
      \square toto_s struct z = {3, 0.5};
      \square int struct toto_s = {3, -1e10};
      \square toto s n. x:
10. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      \square une fonction déclarée mais non définie
      ☐ une directive préprocesseur #include manquante
      □ un désaccord entre la déclaration et la définition
         d'une fonction
      ☐ une fonction appelée avant sa déclaration
11. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     □ char 'c';
     ☐ char c:
      ☐ int char:
      ☐ char "c";
12. Un bit est:
      □ la longueur d'un mot mémoire
     \square un battement d'horloge processeur
      \square un chiffre binaire (0 ou 1)
      ☐ l'instruction qui met fin à un programme
13. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
      \square n = factorielle(p, q);
     \square n = factorielle();
     ☐ printf("%d", factorielle(n));
      ☐ int factorielle(int 2);
14. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
```

```
{
              printf("%d ", i);
         }
     }
   qu'est ce qui sera affiché?
     \Box 1 2 1 2 3
     \Box 0 0 0 1 1 1
     \Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1 0 1
15. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(2/3);
     \square x = racine(racine(x)*racine(x));
     \square x - 1 = racine(x);
     \square x = racine(x * x) - racine(x);
16. Le code suivant :
     int age = 20;
    if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
```

```
affichera:
                                                                   \Box 4
      □ Majeur
                                                                   \Box 0
      \square rien
                                                                   \Box 16
      □ Mineur
                                                             19. Une variable booléenne est un variable :
         Majeur
                                                                   □ à laquelle une valeur vient d'être affectée
      ☐ Mineur
                                                                   □ jamais nulle
17. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
                                                                   □ qui est vraie ou fausse
    une variable entière définie et initialisée, il est correct
                                                                   ☐ réelle positive
    d'écrire :
                                                                   □ NaN (not a number, qui n'est pas un nombre)
      \square int n = carre();
      \square int carre(2);
                                                             20. Le code suivant :
     \square n = carre(int n);
                                                                  int i;
     \square n = carre(n):
                                                                  for (i = 4; i \ge 0; i = i - 1)
18. Pour l'extrait de programme suivant :
                                                                       printf("%d ", i);
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
                                                                  printf("\n");
      for (i = 0; i < 4; i = i + 1)
                                                                 affichera:
        produit = produit * serie[i];
                                                                   \square 1 2 3 4
                                                                   \Box 01234
      printf("produit = %d", produit);
                                                                   \square 4 3 2 1
    La valeur affichée est :
                                                                   \Box 4 3 2 1 0
      \square 8
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Le langage C est un langage □ compilé □ interprété □ composé □ lu, écrit, parlé 2. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :  $\square$  x = racine(2/3):  $\square$  x = racine(racine(x)\*racine(x));  $\square$  x = racine(x \* x) - racine(x);  $\square$  x - 1 = racine(x); 3. Le code suivant : int i: for (i = 8; i > 0; i = i - 2)printf("%d ", i); printf("\n"); affichera:  $\square$  8 2  $\Box$  8 6 4 2 0  $\Box$  8 6 4 2  $\Box 02468$ 4. Pour compiler un programme prog.c, on utilise la ligne de commande : ☐ gcc -Wall prog.c -o prog.exe ☐ gcc prog.exe -Wall -o prog.c
- ☐ gcc -Wall prog.exe -o prog.c
  - ☐ gcc prog.c -o -Wall prog.exe
- 5. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
\square n = factorielle(p, q);
☐ int factorielle(int 2);
□ printf("%d", factorielle(n));
\square n = factorielle();
```

```
6. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 4
     \square 5
    \Box 0
     \Box 1
7. Si a et b sont deux variables de type:
   struct toto_s
   {
     int n:
     double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
    \Box (a.n == b.n) && (a.x == b.x)
    □ a == b
    \square a{n, x} == b{n, x}
     \Box a = b
8. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
```

- , que doit-on chercher dans le programme?
  - ☐ une fonction appelée avant sa déclaration
  - $\square$  un désaccord entre la déclaration et la définition d'une fonction
  - □ une directive préprocesseur #include manquante
  - □ une fonction déclarée mais non définie

```
9. Si le code:
    struct toto_s
      int n:
      double x;
   };
    précède la fonction main(), alors on peut écrire en
    début de main() :
     \Box toto_s struct z = {3, 0.5};
     \Box int struct toto_s = {3, -1e10};
      \square toto_s n, x;
     \square int toto.n = 3;
      □ struct toto_s toto;
10. Une variable booléenne est un variable :
      □ qui est vraie ou fausse
      ☐ jamais nulle
      ☐ réelle positive
      □ NaN (not a number, qui n'est pas un nombre)
      □ à laquelle une valeur vient d'être affectée
11. Un bit est:
      □ un battement d'horloge processeur
      \square un chiffre binaire (0 ou 1)
      □ la longueur d'un mot mémoire
      ☐ l'instruction qui met fin à un programme
12. Le code suivant :
     int somme = 0;
     int i:
     for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
     printf("%d", somme);
    affichera:
      \Box 0
      \square 42
      \Box 1
```

 $\Box$  6

13. Le code suivant :	15. Sous unix (ou linux), la commande cd permet de :	18. Au début de la fonction main() on place le code :
<pre>int i; for (i = 4; i &gt; 0; i = i - 1) {     printf("%d ", i);</pre>	<ul> <li>□ détruire un fichier</li> <li>□ jouer de la musique</li> <li>□ récupérer un programme arrêté avec la commande ab</li> </ul>	<pre>char b = 'A'; b = b + 2; printf("%c\n", b); Alors l'affichage sera :</pre>
<pre>printf("\n"); affichera:</pre>	<ul><li>□ ouvir un bureau partagé (common desktop)</li><li>□ changer de répertoire courant</li></ul>	<ul><li>□ A</li><li>□ B</li><li>□ C</li></ul>
	<ul><li>16. Un programme en langage C doit comporter une et une seule définition de la fonction :</li><li>□ init</li></ul>	□ b  19. Avant de faire appel à une fonction il est nécessaire de :
□ 0 1 2 3 4 □ 0 1 2 3	□ include □ main	☐ l'avoir définie☐ l'avoir déclarée☐ l'avoir declarée☐ l'avoir de
14. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :	□ begin  17. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de x <sup>n</sup> on écrit :	<ul> <li>□ avoir défini une constante symbolique de la taille de cette fonction</li> <li>□ l'avoir déclarée et définie</li> <li>20. Un enregistrement permet de grouper plusieurs valeurs</li> </ul>
<ul> <li>□ un ordre quelconque</li> <li>□ dans lequel ces fonctions sont appelées dans le main</li> <li>□ alphabétique</li> </ul>	□ int exposant(double n, int x); □ exposant(double x, int n, int r); □ void exposant(double x^n);	dans:  □ ses champs □ ses chants □ ses cases
$\Box$ dans lequel vous avez déclaré ces fonction	$\square$ double exposant(double x, int n);	□ ses blocs

т		-1
	acence	

Prénom :	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0,5 points par réponse fausse. Durée: 20 minutes.

1. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt:

□ un débogueur
□ printf("Valeur de n ? %d\n", n);
□ scanf("%d", &n);
□ printf("Valeur de n ? %g\n", n);

 Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit :

☐ int afficher\_date(date\_s d);

□ void afficher\_date(struct date\_s d);
□ void afficher\_date(date\_s d);

☐ struct date\_s afficher\_date(struct date\_s d);

3. Après exécution jusqu'à la ligne 15 du programme C :

```
10 int main() {
11 int x = 5;
12 int y;
13
14 y = x;
15
16 ...
17 }
```

 $\Box$ la variable x vaut 5 et la variable y vaut 0

 $\Box$ la variable x vaut 0

 $\Box\,$ le programme affiche "Faux"

 $\Box\,$ la variable y vaut 5

4. Si le code :

```
struct toto_s
{
  int n;
  double x;
};
```

précède la fonction main(), alors on peut écrire en début de main() :

 $\Box$  int struct toto\_s = {3, -1e10};

```
\square toto_s struct z = {3, 0.5};
     \square toto s n. x:
     \square int toto.n = 3;
     □ struct toto_s toto;
5. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \Box 0
6. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
     \square int k;
     □ loop i;
     ☐ int %d:
     \square int loop n;
7. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
   Alors l'affichage sera:
     \square i
     \Box A
```

☐ ABCDEF

□ ccccc

```
8. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      □ il risque d'afficher bonjour à la place de coucou
      \square il ne compile pas
      □ il n'affiche rien
      □ il comporte une boucle infinie
 9. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x - 1 = racine(x):
      \square x = racine(x * x) - racine(x);
      \square x = racine(racine(x)*racine(x));
      \square x = racine(2/3):
10. Le langage C est un langage
      □ composé
      □ lu, écrit, parlé
      □ compilé
      □ interprété
11. Un fichier source est:
      □ un document qui doit être protégé
      □ un fichier texte qui sera traduit en instructions
      ☐ un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
      \square un document illisible pour les humains
      \square un document de référence du système
12. Sur unix (ou linux), la commande mkdir permet de :
      □ ouvrir un fichier texte
      □ créer un fichier texte
      □ créer un répertoire
      □ changer de répertoire courant
```

13. Le code suivant :	15. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :	☐ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne		
<pre>int age = 20; if (age &lt; 18)</pre>	☐ int factorielle(int x);	peut pas être chargé par le compilateur		
11 (age < 10)	☐ int factorielle(double n);	18. Pour afficher à l'aide de printf("%d\n",tab[i]);		
<pre>printf("Mineur\n"); }</pre>	☐ struct int factorielle(int n);	le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :		
<pre>printf("Majeur\n");</pre>	☐ int factorielle();	☐ for(i=0;i<=5;i=i+1)		
affichera:	16. Quelle étape de la compilation vient d'échouer lors-	☐ for(i=1;i<5;i=i+1)		
	qu'on a un message comme celui-ci :	☐ for(i=1;i<=5;i=i+1)		
☐ Mineur Majeur	Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »	☐ for(i=0;i<5;i=i+1)		
☐ Majeur	☐ l'édition de liens	19. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.		
☐ Mineur	☐ l'analyse des entrées clavier	□ #define N = 3		
$\Box$ rien	☐ l'analyse harmonique	□ #define taille = 3		
	☐ l'analyse sémantique	□ #define N 3		
14. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les ex-	17. Une segmentation fault est une erreur qui survient lorsque:	☐ #define taille = N		
pressions booléennes suivantes, laquelle a pour valeur	☐ le programme tente d'afficher des caractères sur	20. Avant de faire appel à une fonction il est nécessaire		
TRUE?	une ligne qui va au delà de la largeur de la fenêtre	$\mathrm{d}\mathrm{e}$ :		
☐ (A == TRUE) && (B == TRUE)	du terminal	☐ l'avoir déclarée		
□ A && B	☐ la division du programme en zones homogènes	☐ l'avoir définie		
□ !(!A    B) == (A && !B)	échoue	□ avoir défini une constante symbolique de la taille de cette fonction		
□ (!A    B)	☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	☐ l'avoir déclarée et définie		

□ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur
18. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
☐ for(i=0;i<=5;i=i+1)
☐ for(i=1;i<5;i=i+1)
☐ for(i=1;i<=5;i=i+1)
☐ for(i=0;i<5;i=i+1)
19. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
$\square$ #define N = 3
☐ #define taille = 3
$\square$ #define N 3
$\square$ #define taille = N
20. Avant de faire appel à une fonction il est nécessaire de :
□ l'avoir déclarée
□ l'avoir définie

т		-1
	acence	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :

```
□ int n = carre();
□ n = carre(n);
□ n = carre(int n);
□ int carre(2);

2. Le code suivant :
    int i;
    for (i = 0; i < 5; i = i + 1)
    {
        printf("%d ", i);
    }
    printf("\n");

affichera :
□ 4 3 2 1 0
□ 0 1 2 3 4</pre>
```

3. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
□ printf("%d", factorielle(n));
□ n = factorielle(p, q);
□ n = factorielle();
□ int factorielle(int 2);
```

4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :

```
aquelle des analyses sur
apes de la compilation :

□ analyse harmonique
□ analyse lexicale
□ analyse syntaxique
□ analyse sémantique
```

 $\square$  4 3 2 1

 $\square$  0 1 2 3

5. Quel est le problème d'un programme comportant les lignes suivantes?

while (1)
{
 printf("coucou\n");
}

 $\square$  il ne compile pas

□ il n'affiche rien

□ il comporte une boucle infinie□ il risque d'afficher bonjour à la place de coucou

6. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :

```
□ n = pgcd(n, 3);
□ int n = pgcd();
□ int pgcd(2);
□ n = pgcd(int p, int q);
```

7. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise :

```
□ char afficher_menu(printf("menu"));
□ void afficher_menu();
□ double afficher_menu();
□ int afficher_menu(int char);
□ int afficher menu();
```

8. Au début de la fonction main() on place le code :

ссссс	
i	

□ A

9. Avant de faire appel à une fonction il est nécessaire de :

$\hfill \square$ avoir défini une constante symbolique de la tail	le
de cette fonction	

□ l'avoir définie

 $\square$  l'avoir déclarée et définie

□ l'avoir déclarée

10. Si cette erreur apparaît à la compilation :

erreur: conflicting types for 'max', que doiton chercher dans le programme?

	une	fonction	appelée	avant	sa	décla	aration
--	-----	----------	---------	-------	----	-------	---------

□ une fonction déclarée mais non définie

 $\square\,$  une directive préprocesseur  $\# \mathtt{include}$  man quante

un désaccord	entre	la	déclaration	$\operatorname{et}$	la	définition
d'une fonctio	n					

11. Le code suivant :

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 8 2
```

8	6	4	2	

□ 86420

 $\Box 02468$ 

12. Vous utilisez une boucle while quand :

	vous	n	'avez	pas	${\rm d\acute{e}clar\acute{e}}$	de	fonction
--	------	---	-------	-----	---------------------------------	----	----------

vous	ne	connaiss	ez pa	s le	nombre	d'itérations	$\mathrm{d}\epsilon$
la bo	oucl	e à l'avai	nce				

vous	avez	déjà	${\rm fait}$	un	for	${\rm dans}$	le	${\rm m\^{e}me}$	pro-
$\operatorname{gram}$	me p	rincip	al						

 $\square$  l'incrément de la variable de boucle n'est pas 1

13. Une segmentation fault est une erreur qui survient lorsque :  □ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée □ la division du programme en zones homogènes échoue	<pre>16. Pour l'extrait de programme suivant :     int produit = 0;     int serie[4] = {2, 2, 2, 2};     for (i = 0; i &lt; 4; i = i + 1)     {         produit = produit * serie[i]; }</pre>	Alors l'expression g(0) prendra la valeur :   5  0  7  19. Soit la fonction f définie par :
<ul> <li>□ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal</li> <li>□ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur</li> </ul>	printf("produit = %d", produit);  La valeur affichée est :	<pre>int f(int a) {    printf("a = \n", %d);    if (a &gt; 0)    {      return 3;</pre>
<ul> <li>14. Le bus système sert à :</li> <li>☐ Arriver à l'heure en cours</li> <li>☐ Transférer des données et intructions entre processeur et mémoire</li> <li>☐ transporter les processus du tourniquet au processeur</li> <li>☐ Écrire des données sur le dique dur</li> <li>15. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :</li> <li>☐ dans lequel ces fonctions sont appelées dans le main</li> <li>☐ dans lequel vous avez déclaré ces fonction</li> <li>☐ alphabétique</li> <li>☐ un ordre quelconque</li> </ul>	□ 0  17. L'écriture 111 en binaire correspond au nombre naturel: □ 8 □ 3 □ 111 □ 7  18. Soit la fonction g définie par: int g(int a) {    printf("a = \n", %d);    if (1 > 0)    {       return 5;    }    return 7;	return 4; } Alors l'expression f(0) prendra la valeur :  □ 0 □ 4 □ 3  20. L'ordonnancement par tourniquet permet : □ de ne pas perdre de temps avec la commutation de contexte □ d'entretenir l'illusion que les processus tournent en parallèle □ de doubler la mémoire disponible □ d'afficher des ronds colorés à l'écran

т		-1
	acence	

Prénom :	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le langage C est un langage □ composé □ compilé □ lu, écrit, parlé □ interprété 2. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ dans lequel vous avez déclaré ces fonction □ un ordre quelconque □ alphabétique □ dans lequel ces fonctions sont appelées dans le main 3. Après exécution jusqu'à la ligne 14 du programme C: int main() { 11 int x = 5: 12 13 printf(" x =  $%d\n$ ", 2); 14 15 16 } ☐ le terminal affiche "Faux" □ le terminal affiche 5  $\square$  le terminal affiche x = 2  $\square$  le terminal affiche x = 5 4. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme? □ un caractère interdit en C □ une variable non déclarée ☐ une faute de frappe dans un appel de fonction ☐ une directive préprocesseur #include manquante

```
5. Les lignes
  int i;
  int x=0;
  for(i=0,i<5,i=i+1)
     x=x+1;
    □ comportent une erreur qui sera détectée au cours
       de l'édition de lien
    \square ne comportent aucune erreur
    □ comportent une erreur qui sera détectée au cours
       de l'analyse syntaxique
    □ comportent une erreur qui ne sera pas détectée
6. Pour déclarer une fonction saisie_utilisateur qui
  demande à l'utilisateur d'entrer un entier au clavier et
  renvoie cet entier on écrit:
    □ void saisie_utilisateur(int n);
    □ void saisie_utilisateur(char c);
    ☐ int saisie_utilisateur();
    ☐ saisie_utilisateur(scanf(%d));
7. Quels calculs peut-on programmer en programmation
  structurée?
    □ en programmation structurée on peut program-
       mer tous les calculs programmables en langage
       machine
    □ certains programmes sont de vrais plats de spa-
       ghetti
    ☐ il y a des calculs programmables en langage ma-
       chine et qui ne sont pas programmables en pro-
       grammation structurée
    \square il y a des calculs programmables en programma-
       tion structurée qui ne sont pas programmables en
       langage machine
8. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
    □ double afficher_menu();
    ☐ int afficher_menu();
    ☐ int afficher_menu(int char);
    □ void afficher_menu();
```

☐ char afficher\_menu(printf("menu"));

```
9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
      \square void exposant(double x^n);
      \square double exposant(double x, int n);
10. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \square 0.5
      \square 0
      \Box 1
      \square 1.5
11. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il comporte une boucle infinie
      □ il n'affiche rien
      \square il ne compile pas
      \square il risque d'afficher bonjour à la place de coucou
12. Un fichier source est:
      □ un document de référence du système
      □ un document qui doit être protégé
      \square un document illisible pour les humains
      □ un fichier texte qui sera traduit en instructions
         processeur
      □ un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
13. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
    d'écrire :
      \square int n = carre():
      \square n = carre(n);
      \square n = carre(int n):
      \square int carre(2);
```

14.	Une variable booléenne est un variable :
	$\Box$ qui est vraie ou fausse
	$\Box$ à la quelle une valeur vient d'être affectée
	$\square$ NaN (not a number, qui n'est pas un nombre)
	□ réelle positive
	$\square$ jamais nulle
15.	Le type des réels en C est :
	$\square$ int
	□ real
	☐ double
	□ char
16.	Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
	$\square$ int char;
	☐ char 'c';
	☐ char "c";
	□ char c;

```
17. Pour l'extrait de programme suivant :
    int produit = 1;
    int serie[4] = {2, 2, 2, 2};
    for (i = 0; i < 4; i = i + 1)
    {
        produit = produit * serie[i];
    }
    printf("produit = %d", produit);

La valeur affichée est :
    □ 0
    □ 4
    □ 8
    □ 16

18. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :</pre>
```

 $\square$  n = factorielle(p, q);

☐ int factorielle(int 2);

☐ printf("%d", factorielle(n));

□ n = factorielle();

```
19. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
          for (j = 0; j < 3; j = j + 1)
              printf("%d ", j);
         }
   qu'est ce qui sera affiché?
     \Box 0 0 1 1 2 2 3
     \square 0 1 2 3 0 1 2
     \Box 0 1 2 0 1 2
      \Box 0 1 2 0 1 2 3
20. Dans la commande gcc, l'option -Wall signifie :
     \square que l'on veut voir tous les avertissements
      \square qu'il faut lancer un déboggueur
     □ qu'on veut changer alétoirement de fond d'écran
      \square qu'il faut indenter le fichier source
```

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
   int age = 20;
   if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Majeur
    □ Mineur
    □ rien
    ☐ Mineur
       Majeur
2. Pour l'extrait de programme suivant :
   int i;
   int j;
   for(i=4;i>0;i=i-1)
     for(j=i;j<6;j=j+1)
       printf("*");
     printf(" ");
  qu'est ce qui sera affiché?
        **** **** ****
3. Pour l'extrait de programme suivant :
    int produit = 0;
    int serie[4] = \{2, 2, 2, 2\};
    for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
    printf("produit = %d", produit);
```

```
La valeur affichée est :
     \Box 0
     \Box 4
     \square 16
     \square 8
4. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square exposant(double x, int n, int r);
     \square int exposant(double n, int x);
     \square double exposant(double x, int n);
     \square void exposant(double x^n);
5. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     \square il ne compile pas
     □ il n'affiche rien
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il comporte une boucle infinie
6. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %g\n", n);
     ☐ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
     □ un débogueur
7. Vous utilisez une boucle while quand:
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     ☐ l'incrément de la variable de boucle n'est pas 1
```

```
8. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3;
      }
      return 4;
   }
    Alors l'expression f(0) prendra la valeur :
      \square 0
      \square 3
     \Box 4
 9. Soit le programme principal suivant :
    int main()
     int a = 3:
     int b = 5:
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
   {
      a = a + b;
      return a;
   }
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
10. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
      \square une accolade manquante
     ☐ un point-virgule manquant
      ☐ un point-virgule en trop
      \square une accolade en trop
```

1. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et	□ la division du programme en zones homogènes échoue	{ return 5;
renvoie cet entier on écrit :  □ saisie_utilisateur(scanf(%d));	☐ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne	return 7;
□ void saisie_utilisateur(char c);	peut pas être chargé par le compilateur	}
□ void saisie_utilisateur(int n);	14. Une variable booléenne est un variable :	Alors l'expression g(0) prendra la valeur :
☐ int saisie_utilisateur();	☐ réelle positive	□ 0
	$\Box$ jamais nulle	□ 7
2. Soit un programme contenant les lignes suivantes :	$\square$ NaN (not a number, qui n'est pas un nombre)	□ 5
int i = 0;	$\Box$ à la quelle une valeur vient d'être affectée	18. Un fichier source est:
int j = 0; for (i = 0; i < 2; i = i + 1)	$\Box$ qui est vraie ou fausse	$\Box$ un fichier que l'ont doit citer dans les documents
{	15. Un registre du processeur est :	produits sur l'ordinateur
for $(j = 0; j < 3; j = j + 1)$	$\hfill \square$ une unité de calcul spécialisée de l'ordinateur	$\hfill\Box$ un document de référence du système
<pre>{     printf("%d ", i);</pre>	$\Box$ une case mémoire interne au processeur qui sera	□ un document qui doit être protégé
}	manipulée directement lors des calculs	$\square$ un document illisible pour les humains
}	☐ un composant qui contient la liste des fichiers du système	☐ un fichier texte qui sera traduit en instructions processeur
qu'est ce qui sera affiché?	$\Box$ une gamme de fréquence de fonctionnement du	19. Si x est une variable réelle (de type double) alors
	processeur	19. Si $x$ est une variable feene (de type double) alors $x = 3/2$ lui affecte la valeur :
	16. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :	□ 0.5
□ 0 1 2 0 1 2	x = mccarthy(n);	
$\Box$ 1 2 1 2 3	$\square$ n = mccarthy(p, q);	
3. Une segmentation fault est une erreur qui survient	☐ int mccarthy(int 2);	□ 1.5
lorsque:	$\square$ n = mccarthy();	20. Quel est l'opérateur de différence en C :
☐ le programme tente d'afficher des caractères sur	17. Soit la fonction g définie par :	
une ligne qui va au delà de la largeur de la fenêtre	int g(int a)	□ ♦
du terminal	{	☐ !=
☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	<pre>printf("a = \n", %d); if (1 &gt; 0)</pre>	

## Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution du programme :
     lecture 8 r0
      valeur 3 r1
      mult r1 r0
 4
     valeur 1 r2
      add r2 r0
 6
      ecriture r0 8
      stop
 8
     5
    \square le bus explose
    □ le terminal affiche 8
    □ la case mémoire 8 contiendra 0
    □ la case mémoire 8 contiendra 16
2. Soit la fonction g définie par :
  int g(int a)
  {
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
    \square 5
    \Box 7
    \square 0
3. Pour l'extrait de programme suivant :
    int i;
    int j;
    for(i=4;i>0;i=i-1)
    {
      for(j=i;j<6;j=j+1)
        printf("*");
      printf(" ");
```

```
qu'est ce qui sera affiché?
     ** ** ** ** **
      ** *** **** *****
         **** **** ****
4. Soit un programme contenant les lignes suivantes :
   int i = 0;
   int j = 0;
   for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
            printf("%d ", i);
        }
   }
  qu'est ce qui sera affiché?
    \Box 0 1 0 1 0 1 0 1
    \Box 0 1 2 0 1 2
    \Box 1 2 1 2 3
    \Box 0 0 0 1 1 1
5. Pour déclarer une fonction saisie_utilisateur qui
  demande à l'utilisateur d'entrer un entier au clavier et
  renvoie cet entier on écrit:
    □ void saisie_utilisateur(char c);
    ☐ saisie_utilisateur(scanf(%d));
    ☐ int saisie_utilisateur();
    □ void saisie_utilisateur(int n);
6. On considère deux variables booléennes A et B initia-
  lisées à TRUE et FALSE respectivement. Parmi les ex-
  pressions booléennes suivantes, laquelle a pour valeur
  TRUE?
    \square !(!A \mid | B) == (A \&\& !B)
    \square (!A || B)
    \square (A == TRUE) && (B == TRUE)
```

□ A && B

```
7. Vous utilisez une boucle while quand:
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous n'avez pas déclaré de fonction
 8. Le code suivant :
     int age = 18;
     if (age < 18)
    {
         printf("Mineur\n");
    }
     else
     {
         printf("Majeur\n");
   affichera:
     □ Majeur
     □ Mineur
     □ Mineur
        Majeur
     \square rien
9. Au début de la fonction main() on place le code :
     char b = 'A';
    b = b + 2:
    printf("%c\n", b);
    Alors l'affichage sera:
     □ b
     \Box C
     □ A
     □В
10. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \Box printf("x=%x et y=%y\n");
     \Box printf("x=%d et y=%d\n",x y);
     \square printf("x=%d et y=%d\n",x,y);
```

 $\square$  printf("x=%d et y=%d\n,x,y");

11. Le langage C est un langage	14. L'ordonnancement par tourniquet permet :	$\Box$ il comporte une boucle infinie
$\Box$ lu, écrit, parlé	☐ de doubler la mémoire disponible	$\square$ il n'affiche rien
$\square$ compilé	$\Box$ de ne pas perdre de temps avec la commutation	$\Box$ il risque d'afficher bonjour à la place de coucou
□ composé	de contexte	$\Box$ il ne compile pas
□ interprété	☐ d'afficher des ronds colorés à l'écran☐ d'entretenir l'illusion que les processus tournent☐	18. Après la déclaration : int mccarthy(int n);, il est
12. Si cette erreur apparaît à la compilation :  Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit- on chercher dans le programme?  une faute de frappe dans un appel de fonction une directive préprocesseur #include manquante une variable non déclarée un caractère interdit en C  13. Soit la fonction f définie par : int f(int a) {  printf("a = \n", %d); if (a > 0) {	☐ d'entretenir l'illusion que les processus tournent en parallèle  15. Si a et b sont deux variables de type :  struct toto_s {    int n;    double x; };  Alors pour tester l'égalité de a et de b on utilise la condition :  ☐ (a.n == b.n) && (a.x == b.x)  ☐ a{n, x} == b{n, x}  ☐ a == b  ☐ a = b  ☐ a = b	<pre>correct d'écrire :</pre>
t return f(a - 1) + 1;	□ qu'il faut indenter le fichier source	Alors l'expression f(0) prendra la valeur :
}	☐ que l'on veut voir tous les avertissements	$\Box$ 4
return 4;	□ qu'il faut lancer un déboggueur	$\square$ 3
}	☐ qu'on veut changer alétoirement de fond d'écran	$\Box$ 0
Alors l'expression f(1) prendra la valeur :	17. Quel est le problème d'un programme comportant les	20. Un bit est :
$\Box$ 4	lignes suivantes?	$\square$ l'instruction qui met fin à un programme
$\Box$ 5	while (1)	$\square$ un battement d'horloge processeur
$\Box$ 1	{	$\square$ la longueur d'un mot mémoire
$\Box$ 0	<pre>printf("coucou\n"); }</pre>	$\Box$ un chiffre binaire (0 ou 1)

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :  $\square$  n = mccarthy();  $\square$  int mccarthy(int 2);  $\square$  x = mccarthy(n);  $\square$  n = mccarthy(p, q); 2. Après exécution du programme : lecture 8 r0 valeur 3 r1 mult r1 r0 4 valeur 1 r2 add r2 r0 6 ecriture r0 8 7 stop 5 □ la case mémoire 8 contiendra 16 □ le terminal affiche 8  $\square$  la case mémoire 8 contiendra 0  $\Box$  le bus explose 3. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ alphabétique  $\square$  un ordre quelconque □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main 4. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  n = carre(int n):  $\square$  n = carre(n):

 $\square$  int n = carre():

 $\Box$  int carre(2);

```
5. Si a et b sont deux variables de type:
   struct toto s
   {
     int n:
     double x;
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
    □ a == b
    \Box a = b
    \Box (a.n == b.n) && (a.x == b.x)
     \square a{n, x} == b{n, x}
6. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
    ☐ printf("Valeur de n ? %d\n", n);
    □ un débogueur
    □ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     □ ses chants
     \square ses blocs
     □ ses cases
8. Le code suivant:
    int i:
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 1234
     \Box 01234
     \Box \ 4\ 3\ 2\ 1\ 0
```

 $\square$  4 3 2 1

```
9. Quelle étape de la compilation vient d'échouer lors-
    qu'on a un message comme celui-ci :
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
      □ l'analyse sémantique
      □ l'édition de liens
      □ l'analyse des entrées clavier
      □ l'analyse harmonique
10. Après exécution jusqu'à la ligne 14 du programme C :
       int main() {
 11
            int x = 5;
 12
 13
            printf(" x = %d\n", 2);
 14
 15
            . . .
 16
      }
      \square le terminal affiche x = 5
      □ le terminal affiche "Faux"
     \square le terminal affiche 5
     \Box le terminal affiche x = 2
11. Le bus système sert à :
      ☐ Arriver à l'heure en cours
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     □ transporter les processus du tourniquet au pro-
        cesseur
      ☐ Écrire des données sur le dique dur
12. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
      \square exposant(double x, int n, int r);
     \square double exposant(double x, int n);
      \square void exposant(double x^n);
      \square int exposant(double n, int x);
```

```
15. Pour afficher à l'aide de printf("%d\n",tab[i]);
13. Le code suivant :
                                                                le contenu d'un tableau de 5 entiers initialisé au
     int i;
                                                                préalable, on utilise plutôt :
                                                                                                                              return 7;
     for (i = 8; i > 0; i = i - 2)
                                                                  \square for(i=1;i<=5;i=i+1)
         printf("%d ", i);
                                                                  \square for(i=0;i<=5;i=i+1)
     }
                                                                                                                              \Box 0
                                                                  \square for(i=0;i<5;i=i+1)
     printf("\n");
                                                                                                                              \Box 5
                                                                  \square for(i=1;i<5;i=i+1)
    affichera:
                                                                                                                              \Box 7
                                                            16. Pour l'extrait de programme suivant :
      \Box 8 6 4 2 0
                                                                  int somme = 0;
      \Box 02468
                                                                  int serie[4] = \{2, 4, 10, 4\};
                                                                  for (i = 0; i < 4; i = i + 1)
      \square 8 2
      \square 8 6 4 2
                                                                     somme = somme + serie[i];
14. Soit le programme principal suivant :
                                                                  printf("somme = %d",somme);
                                                                                                                                 ab
    int main()
                                                                La valeur de somme affichée est :
    {
     int a = 3;
                                                                  \Box 16
     int b = 5;
                                                                  \square 20
                                                                                                                             int age = 18;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
                                                                                                                             if (age < 18)
                                                                  \Box 6
     return EXIT_SUCCESS;
                                                                  \square 3
                                                            17. L'écriture 111 en binaire correspond au nombre natu-
                                                                                                                             }
    appelant la fonction f ainsi définie :
                                                                rel:
                                                                                                                             else
    int f(int a, int b)
                                                                                                                             {
                                                                  \square 8
    {
                                                                  \square 3
      a = a + b;
      return a;
                                                                  □ 111
                                                                  \Box 7
                                                                                                                            affichera:
    L'affichage dans le main est le suivant :
                                                            18. Soit la fonction g définie par :
                                                                                                                              □ Mineur
      \Box f(a,b)=13, a=8, b=5
                                                                int g(int a)
                                                                                                                              \square rien
      \Box f(a,b)=8, a=8, b=5
                                                                                                                              □ Majeur
                                                                  printf("a = \n", %d);
      \Box f(3,5)=8, a=3, b=5
                                                                                                                              ☐ Mineur
                                                                  if (1 > 0)
      \Box f(a,b)=8, a=3, b=5
                                                                                                                                 Majeur
                                                                  {
```

```
return 5;
   Alors l'expression g(0) prendra la valeur :
19. Sous unix (ou linux), la commande cd permet de :
      \square jouer de la musique
     □ détruire un fichier
      □ ouvir un bureau partagé (common desktop)
      ☐ récupérer un programme arrêté avec la commande
      □ changer de répertoire courant
20. Le code suivant :
         printf("Mineur\n");
         printf("Majeur\n");
```

# Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Quels calculs peut-on programmer en programmation structurée?  $\square$  il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine □ certains programmes sont de vrais plats de spaghetti
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
  - $\square$  il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
- 2. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :
  - $\Box$  0
  - $\square$  1.5
  - $\Box$  1
  - $\square$  0.5
- 3. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max'

, que doit-on chercher dans le programme?

- $\square$  une directive préprocesseur #include manquante
- $\square$  un désaccord entre la déclaration et la définition d'une fonction
- ☐ une fonction appelée avant sa déclaration
- □ une fonction déclarée mais non définie
- 4. Soit la fonction f définie par :

```
int f(int a)
  printf("a = \n", %d);
  if (a > 0)
    return 3;
  return 4;
```

```
Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 0
     \Box 4
5. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire:
    \square x = mccarthy(n);
    \square n = mccarthy(p, q);
     \square n = mccarthy();
     \Box int mccarthy(int 2);
```

- 6. Le code suivant : int somme = 0;
  - int i: for (i = 1; i < 4; i = i + 1)somme = somme + i;
  - printf("%d", somme);

affichera:

- $\Box$  6
- $\Box$  0
- $\square$  42
- $\Box$  1
- 7. Pour compiler un programme prog.c, on utilise la ligne de commande :
  - ☐ gcc -Wall prog.exe -o prog.c
  - ☐ gcc prog.c -o -Wall prog.exe
  - ☐ gcc -Wall prog.c -o prog.exe
  - ☐ gcc prog.exe -Wall -o prog.c
- 8. Les lignes

```
int i;
int x=0;
for(i=0,i<5,i=i+1)
  x=x+1;
```

- □ comportent une erreur qui sera détectée au cours de l'édition de lien
- $\square$  ne comportent aucune erreur
- □ comportent une erreur qui ne sera pas détectée
- □ comportent une erreur qui sera détectée au cours de l'analyse syntaxique
- 9. Au début de la fonction main() on place le code :

```
char b = 'A';
b = b + 2:
printf("%c\n", b);
```

Alors l'affichage sera:

- □ b □ A
- □В
- $\Box$  C
- 10. Dans la commande gcc, l'option -Wall signifie :
  - □ qu'on veut changer alétoirement de fond d'écran
  - ☐ qu'il faut indenter le fichier source
  - ☐ que l'on veut voir tous les avertissements
  - □ qu'il faut lancer un déboggueur
- 11. Soit la fonction g définie par :

```
int g(int a)
 printf("a = \n", %d);
 if (1 > 0)
 {
    return 5:
```

Alors l'expression g(0) prendra la valeur :

 $\Box$  5

return 7;

- $\Box$  7
- $\Box$  0
- 12. Sur unix (ou linux), la commande mkdir permet de :
  - □ changer de répertoire courant
  - □ ouvrir un fichier texte
  - □ créer un répertoire
  - $\square$  créer un fichier texte

13. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
$\square$ #define taille = N
$\square$ #define N = 3
$\square$ #define N 3
$\square$ #define taille = 3
14. Si $n$ est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :
☐ scanf("%d", &n);
$\Box$ un débogueur
$\square$ printf("Valeur de n ? %d\n", n);
$\square$ printf("Valeur de n ? %g\n", n);
15. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
$\square$ n = carre(int n);
$\square$ int n = carre();
☐ int carre(2);
$\square$ n = carre(n);

 $\square$  (!A || B)

```
16. Pour l'extrait de programme suivant :
                                                                   \square !(!A \mid | B) == (A \&\& !B)
      int produit = 1;
                                                                  □ A && B
      int serie[4] = \{2, 2, 2, 2\};
                                                                  ☐ (A == TRUE) && (B == TRUE)
      for (i = 0; i < 4; i = i + 1)
                                                            19. Si a et b sont deux variables de type:
        produit = produit * serie[i];
                                                                 struct toto_s
      printf("produit = %d", produit);
                                                                   int n;
    La valeur affichée est :
                                                                   double x;
                                                                };
      \Box 0
                                                                Alors pour tester l'égalité de a et de b on utilise la
      \Box 16
                                                                condition:
      \Box 4
                                                                  □ a = b
      \square 8
                                                                  □ a == b
17. Le langage C est un langage
                                                                  \square a\{n, x\} == b\{n, x\}
     □ interprété
                                                                   \square (a.n == b.n) && (a.x == b.x)
     □ compilé
     □ lu, écrit, parlé
                                                             20. Pour déclarer une variable qui sera utilisée comme va-
                                                                 riable de boucle on peut utiliser l'instruction
      □ composé
18. On considère deux variables booléennes A et B initia-
                                                                  \square int loop n;
    lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                   \square int k;
    pressions booléennes suivantes, laquelle a pour valeur
                                                                   \square loop i;
    TRUE?
                                                                  \square int %d;
```

# Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Quels calculs peut-on programmer en programmation structurée?
  - □ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
  - □ certains programmes sont de vrais plats de spaghetti
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
  - ☐ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
- 2. Si cette erreur apparaît à la compilation :

Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme?

- $\Box$  un caractère interdit en C
- $\Box$  une variable non déclarée
- $\hfill\square$  une faute de frappe dans un appel de fonction
- $\hfill \square$  une directive préprocesseur # include manquante
- 3. Après exécution jusqu'à la ligne 14 du programme C :

```
10   int main() {
11       int x = 5;
12
13       printf(" x = %d\n", 2);
14
15       ...
16   }
   □ le terminal affiche x = 5
   □ le terminal affiche 5
```

- $\square$  le terminal affiche "Faux"
- $\square$  le terminal affiche x = 2
- 4. Sur unix (ou linux), la commande mkdir permet de :

```
\square créer un répertoire
```

- $\Box$ créer un fichier texte
- $\square$ ouvrir un fichier texte
- □ changer de répertoire courant

```
5. Le code suivant :
```

```
int age = 20;
if (age < 18)
{
    printf("Mineur\n");
}
printf("Majeur\n");
affichera:</pre>
```

- □ Majeur
- □ Mineur Majeur
- $\square$  rien
- ☐ Mineur

int a = 0;

6. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  ${\tt n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  ${\tt n}$ . Soit le programme suivant :

```
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
    scanf("%d", &n);
}
```

Quelle est la condition cond :

- □ a<=n<=b
- $\square$  (n<=a) && (n<=b)
- $\square$  (a<=n) && (n<=b)
- $\square$  (a<n) || (n>b)
- 7. Si cette erreur apparaît à la compilation :

error: expected ';' before '}' token que doiton chercher dans le programme?

- $\Box\,$  une accolade man quante
- $\Box$  un point-virgule man quant
- $\Box$  un point-virgule en trop
- $\square$  une accolade en trop

8. Après exécution jusqu'à la ligne 15 du programme  ${\bf C}$  :

```
10 int main() {
11 int x = 5;
12 int y = 3;
13
14 x = y;
15
16 ...
17 }
```

- $\square$  la variable x vaut 5 et la variable y vaut 3
- □ le programme affiche "Faux"
- $\square$  la variable x vaut 3
- $\square$  la variable y vaut 5
- 9. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\square x - 1 = racine(x);
```

 $\square$  x = racine(2/3);

 $\square$  x = racine(racine(x)\*racine(x));

 $\square$  x = racine(x \* x) - racine(x);

10. Soit la fonction f définie par :

```
int f(int a)
{
   printf("a = \n", %d);
   if (a > 0)
   {
     return 3;
   }
   return 4;
```

Alors l'expression f(0) prendra la valeur :

- $\Box$  4
- $\square$  3
- $\Box$  0

<pre>dans le</pre>
<pre>int toto[taille=5];  19. Le code suivant :  int age = 18;     if (age &lt; 18)     {         printf("Mineur\n");     }     else     { </pre>
19. Le code suivant :  int age = 18;  if (age < 18)  {      printf("Mineur\n"); }  else {
<pre>int age = 18; if (age &lt; 18) {     printf("Mineur\n"); } else {</pre>
<pre>if (age &lt; 18) {      printf("Mineur\n"); } else {</pre>
<pre>printf("Mineur\n"); } else {</pre>
} else {
else {
nrintf("Majeur\n")·
B initia- i les ex- r valeur  affichera:
☐ Mineur
Majeur
□ rien
□ Mineur
□ Majeur
tions de $\begin{bmatrix} 20. & \text{Si } n \text{ est une variable entière pour demander sa valeur} \\ & \text{à l'utilisateur, on utilise plutôt} \end{bmatrix}$
☐ printf("Valeur de n ? %g\n", n);
me pro- □ un débogueur
-
☐ printf("Valeur de n ? %d\n", n);

Lacence	· I

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Au début de la fonction main() on place le code : char i; for  $(i = 'A'; i \le 'F'; i = i + 1)$ printf("%c", i); printf("\n"); Alors l'affichage sera:  $\square$  A  $\Box$  i ☐ ABCDEF 2. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise : ☐ double afficher\_menu(); ☐ int afficher\_menu(int char); ☐ int afficher\_menu(); □ void afficher\_menu(); ☐ char afficher\_menu(printf("menu")); 3. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :  $\square$  n = mccarthy(p, q);  $\square$  int mccarthy(int 2);  $\square$  x = mccarthy(n);  $\square$  n = mccarthy(); 4. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?  $\square$  une accolade manquante ☐ un point-virgule manquant □ un point-virgule en trop

 $\square$  une accolade en trop

5.	Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :
	$\Box$ chacun son tour, après que le processus précédent a terminé
	$\Box$ tour à tour, un petit peu à chaque fois
	$\square$ tous ensemble
	$\Box$ en parallèle, chacun dans un registre
6.	Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :
	$\square$ x = racine(x * x) - racine(x);
	$\square$ x = racine(racine(x)*racine(x));
	$\square$ x - 1 = racine(x);
	$\square$ x = racine(2/3);
7.	Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
	$\square$ n = carre(n);
	□ n = carre(int n);
	☐ int carre(2);
	☐ int n = carre();
8.	L'écriture $\underline{101}$ en binaire correspond au nombre naturel :
	□ 101
	$\Box$ 4
	$\square$ 3
	$\Box$ 5
9.	Quels calculs peut-on programmer en programmation structurée ?
	$\Box$ en programmation structurée on peut programmer tous les calculs programmables en langage machine
	☐ il y a des calculs programmables en langage ma- chine et qui ne sont pas programmables en pro-

grammation structurée

```
□ certains programmes sont de vrais plats de spa-
         ghetti
      \square il y a des calculs programmables en programma-
         tion structurée qui ne sont pas programmables en
         langage machine
10. Une de ces manière de composer les blocs de pro-
    grammes ne fait pas partie des opérations de la pro-
    grammation structurée :
      □ retourner un bloc
      □ sélectionner entre deux blocs à l'aide d'une condi-
      □ répéter un bloc tant qu'une condition est vérifée
      \square mettre les blocs en séquence les uns à la suite des
         autres
11. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     printf("Majeur\n");
    affichera:
      □ Mineur
     □ Mineur
         Majeur
      \square rien
      □ Majeur
12. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 4
      \square 0
```

13. Un bit est:	16. Vous utilisez une boucle while quand:	qu'est ce qui sera affiché?
$\Box$ l'instruction qui met fin à un programme	□ vous avez déjà fait un for dans le même pro-	□ 0 1 2 0 1 2
$\square$ un chiffre binaire (0 ou 1)	gramme principal  ☐ vous n'avez pas déclaré de fonction	□ 0 0 0 1 1 1
<ul><li>□ un battement d'horloge processeur</li><li>□ la longueur d'un mot mémoire</li></ul>	□ vous navez pas declare de lonction □ vous ne connaissez pas le nombre d'itérations de	□ 0 1 0 1 0 1 0 1
14. Le type des réels en C est :	la boucle à l'avance	□ 1 2 1 2 3
□ char □ real	☐ l'incrément de la variable de boucle n'est pas 1  17. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
☐ double	☐ char "c";	$\hfill\Box$ analyse lexicale
$\Box$ int	□ char 'c';	$\hfill\Box$ analyse sémantique
15. Après exécution jusqu'à la ligne 14 du programme C :	□ char c;	$\square$ analyse harmonique
10 int main() { 11 int x = 5;	☐ int char;  18. Soit un programme contenant les lignes suivantes :	$\Box$ analyse syntaxique
12 13	<pre>int i = 0; int j = 0; for (i = 0; i &lt; 2; i = i + 1) {     for (j = 0; j &lt; 3; j = j + 1)     {         printf("%d ", i); }</pre>	20. Quelle étape de la compilation vient d'échouer lors- qu'on a un message comme celui-ci : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »  ☐ l'analyse harmonique ☐ l'analyse sémantique
□ le terminal affiche "Faux" □ le terminal affiche 5	}	☐ l'analyse des entrées clavier
$\Box$ le terminal affiche x = 5	}	☐ l'édition de liens

## Éléments d'informatique – contrôle continue

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N}^{\circ} \text{ INE}: \end{array}$ 

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\square$  3  $\Box$  0  $\Box$  4 2. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise : ☐ char afficher\_menu(printf("menu")); ☐ double afficher\_menu(); ☐ int afficher\_menu(); ☐ int afficher\_menu(int char); □ void afficher\_menu(); 3. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3. ☐ #define taille = N □ #define N 3  $\square$  #define N = 3  $\square$  #define taille = 3 4. Soit le programme principal suivant : int main() int a = 3: int b = 5: printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b$ ); return EXIT\_SUCCESS;

appelant la fonction f ainsi définie :

```
int f(int a. int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
5. Après exécution du programme :
      lecture 8 r0
      valeur 3 r1
      mult r1 r0
      valeur 1 r2
      add r2 r0
      ecriture r0 8
      stop
      5
     □ la case mémoire 8 contiendra 16
     \square la case mémoire 8 contiendra 0
     \square le bus explose
     □ le terminal affiche 8
6. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int n = carre();
     \square n = carre(n);
     \square n = carre(int n);
     \square int carre(2);
7. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(2/3);
    \square x = racine(racine(x)*racine(x));
     \square x - 1 = racine(x);
```

 $\square$  x = racine(x \* x) - racine(x);

```
8. Vous utilisez une boucle while quand:
      □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
      □ l'incrément de la variable de boucle n'est pas 1
 9. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
     printf("\n");
   affichera:
     \Box 0 1 2 3 4 5 6 7
     \Box 0 1 2 3 4 5 6
     \Box 02468
     \Box 0246
10. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle(double n):
     \square int factorielle(int x);
     ☐ int factorielle();
     □ struct int factorielle(int n);
11. Le bus système sert à :
      □ transporter les processus du tourniquet au pro-
        cesseur
      ☐ Arriver à l'heure en cours
     ☐ Écrire des données sur le dique dur
      ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
12. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
      □ en parallèle, chacun dans un registre
     □ tour à tour, un petit peu à chaque fois
     □ chacun son tour, après que le processus précédent
        a terminé
      \square tous ensemble
```

13. L'ordonnancement par tourniquet permet :	16. Laquelle de ces écritures correspond à la déclaration	$\square$ void saisie_utilisate
$\Box$ d'afficher des ronds colorés à l'écran	d'une variable de type caractère en langage C?	$\square$ int saisie_utilisateu
$\Box$ d'entretenir l'illusion que les processus tournent	$\square$ int char;	$\square$ saisie_utilisateur(sc
en parallèle	□ char 'c';	$\square$ void saisie_utilisate
$\Box$ de doubler la mémoire disponible	□ char "c";	19. Si factorielle est une foncti
$\Box$ de ne pas perdre de temps avec la commutation	□ char c;	entier et renvoyant un entier,
de contexte	17. Soit un programme contenant les lignes suivantes :	$\square$ n = factorielle();
14. Après exécution jusqu'à la ligne 15 du programme C :	int i = 0;	$\Box$ n = factorielle(p, q)
10 int main() {	int j = 0;	$\Box$ int factorielle(int 2
11 int $x = 5$ ;	for (i = 0; i < 3; i = i + 1)	$\square$ printf("%d", factorie
12   int y = 3;	{ for (i = 0; i < 5; i = i + 1)	20. Pour l'extrait de programme s
13	for (j = 0; j < 5; j = j + 1)	• 0
14 x = y; 15		int i = 0;
16	}	int j = 0; for (i = 0; i < 2; i = i
17 }	}	{
$\Box$ la variable x vaut 5 et la variable y vaut 3	printf("j = %d\n", j);	for (j = 0; j < 3; j
$\Box$ la variable y vaut 5	<i>m</i> 1 / 1 / 10	{
$\Box$ la variable x vaut 3	qu'est ce qui sera affiché par ce printf?	printf("%d ", j) }
□ le programme affiche "Faux"	□ j = %d	}
15. Le langage C est un langage	□ j = 5	qu'est ce qui sera affiché?
□ composé	□ j = 4	□ 0 1 2 0 1 2 3
□ interprété	□ j = 0	$\Box$ 0 1 2 3 0 1 2
□ compilé	18. Pour déclarer une fonction saisie_utilisateur qui	
-	demande à l'utilisateur d'entrer un entier au clavier et	
$\square$ lu, écrit, parlé	renvoie cet entier on écrit :	$\Box$ 0 1 2 0 1 2

```
□ void saisie_utilisateur(int n);
                         eur();
                         scanf(%d));
                         ceur(char c);
                          tion prenant en entrée un
                          il est correct d'écrire :
                         2);
                         elle(n));
                         suivant :
                           + 1)
                           = j + 1)
```

□ l'analyse sémantique

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points 5. Après exécution jusqu'à la ligne 15 du programme C: 8. Après exécution du programme : par réponse fausse. Durée : 20 minutes. int main() { lecture 8 r0 1. Une variable booléenne est un variable : 11 int x = 5; valeur 3 r1 ☐ jamais nulle 12 int y = 3; mult r1 r0 13 valeur 1 r2 □ à laquelle une valeur vient d'être affectée 14 x = y;add r2 r0 □ NaN (not a number, qui n'est pas un nombre) 15 ecriture r0 8 □ qui est vraie ou fausse 16 stop ☐ réelle positive } 17 2. Quel est le problème d'un programme comportant les □ le programme affiche "Faux" lignes suivantes? □ le terminal affiche 8  $\square$  la variable x vaut 3 while (1) □ la case mémoire 8 contiendra 0  $\square$  la variable x vaut 5 et la variable y vaut 3  $\square$  le bus explose  $\square$  la variable y vaut 5 printf("coucou\n"); □ la case mémoire 8 contiendra 16 6. Pour déclarer une procédure afficher\_date qui prend □ il risque d'afficher bonjour à la place de coucou en argument un struct date\_s et affiche le contenu 9. Le bus système sert à : □ il comporte une boucle infinie du struct, on écrit: ☐ Transférer des données et intructions entre pro-□ il n'affiche rien □ void afficher\_date(struct date\_s d); cesseur et mémoire  $\square$  il ne compile pas ☐ int afficher\_date(date\_s d); □ transporter les processus du tourniquet au pro-3. Soit la fonction f définie par : □ void afficher\_date(date\_s d); cesseur int f(int a) □ struct date\_s afficher\_date(struct date\_s d); ☐ Écrire des données sur le dique dur 7. Soit le programme principal suivant : ☐ Arriver à l'heure en cours  $printf("a = \n", %d);$ int main() if (a > 0)10. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doitint a = 3: return 3; on chercher dans le programme? int b = 5;  $\square$  un désaccord entre la déclaration et la définition printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b$ ); return 4; d'une fonction return EXIT\_SUCCESS; □ une fonction déclarée mais non définie Alors l'expression f(0) prendra la valeur : appelant la fonction f ainsi définie : ☐ une directive préprocesseur #include manquante  $\Box$  0 int f(int a, int b)  $\square$  3 ☐ une fonction appelée avant sa déclaration  $\Box$  4 11. Si cet avertissement apparaît à la compilation : a = a + b: 4. Quelle étape de la compilation vient d'échouer lorswarning: implicit declaration of function 'max' return a; qu'on a un message comme celui-ci : , que doit-on chercher dans le programme? } Undefined symbols :"\_prinft" ou ☐ une fonction appelée avant sa déclaration L'affichage dans le main est le suivant : référence indéfinie vers « prinft » □ une fonction déclarée mais non définie  $\Box$  f(3,5)=8, a=3, b=5 ☐ l'analyse harmonique  $\square$  un désaccord entre la déclaration et la définition  $\Box$  f(a,b)=8, a=8, b=5 □ l'analyse des entrées clavier d'une fonction □ l'édition de liens  $\Box$  f(a,b)=13, a=8, b=5 □ une directive préprocesseur #include manquante

 $\Box$  f(a,b)=8, a=3, b=5

```
12. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     \square rien
     \square Mineur
     ☐ Mineur
        Majeur
     □ Majeur
13. Le langage C est un langage
     □ interprété
     □ compilé
     □ lu, écrit, parlé
     □ composé
14. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     □ char 'c';
     ☐ int char;
     ☐ char "c";
     ☐ char c;
```

```
15. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
    avec la commande:
     \square printf("x=%d et y=%d\n",x,y);
     \Box printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n,x,y");
     \square printf("x=%d et y=%d\n",x y);
16. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
     □ qu'on veut changer alétoirement de fond d'écran
     ☐ que l'on veut voir tous les avertissements
17. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
      \square 3
      \Box 16
      \Box 6
      \square 20
```

```
18. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
19. L'écriture 111 en binaire correspond au nombre natu-
    rel:
     \square 3
     □ 111
     \Box 7
     \square 8
20. Si a et b sont deux variables de type:
    struct toto_s
   {
      int n;
      double x;
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \square (a.n == b.n) & (a.x == b.x)
     □ a == b
     \Box a = b
     \square a\{n, x\} == b\{n, x\}
```

## Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Le langage C est un langage □ composé □ interprété □ compilé □ lu, écrit, parlé 2. Le bus système sert à : □ transporter les processus du tourniquet au pro-☐ Arriver à l'heure en cours ☐ Écrire des données sur le dique dur □ Transférer des données et intructions entre processeur et mémoire 3. Pour l'extrait de programme suivant : int somme = 0; int  $serie[4] = \{2, 4, 10, 4\};$ for (i = 0; i < 4; i = i + 1)somme = somme + serie[i]; printf("somme = %d",somme); La valeur de somme affichée est :  $\square$  20  $\square$  16  $\Box$  6  $\square$  3 4. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci : Undefined symbols : "\_prinft" ou référence indéfinie vers « prinft » □ l'édition de liens ☐ l'analyse harmonique □ l'analyse des entrées clavier

□ l'analyse sémantique

```
5. Le code suivant :
   int i:
   for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
   printf("\n");
  affichera:
    \Box 0 1 2 3 4 5 6 7
    \Box 0123456
    \square 0 2 4 6
    \Box 02468
6. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
    ☐ int factorielle();
    \square int factorielle(int x);
    ☐ int factorielle(double n);
    □ struct int factorielle(int n);
7. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
  mande:
    □ new TP4
    ☐ kwrite TP4
    ☐ yppasswd
    ☐ mkdir TP4
8. Après exécution du programme :
     lecture 8 r0
      valeur 3 r1
      mult r1 r0
     valeur 1 r2
      add r2 r0
      ecriture r0 8
      stop
      5
    \square le bus explose
    □ le terminal affiche 8
    □ la case mémoire 8 contiendra 16
    \square la case mémoire 8 contiendra 0
```

```
9. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
         }
    }
   qu'est ce qui sera affiché?
     \Box 0 1 0 1 0 1 0 1
     \Box 0 1 2 0 1 2
     \Box 1 2 1 2 3
     \Box 0 0 0 1 1 1
10. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
    printf("\n");
   Alors l'affichage sera:
     ☐ ABCDEF
     □ A
     Πi
11. Soit le programme principal suivant :
   int main()
    int a = 3:
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
```

appelant la fonction f ainsi définie :

```
int f(int a, int b)
      a = a + b;
      return a;
    L'affichage dans le main est le suivant :
      \Box f(3,5)=8, a=3, b=5
      \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
12. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \square 7
      \square 3
      \square 8
      \square 111
13. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      \square dans lequel vous avez déclaré ces fonction
      \square un ordre quelconque
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
14. Si a et b sont deux variables de type:
    struct toto_s
      int n;
      double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
```

```
\square (a.n == b.n) & (a.x == b.x)
      \square a{n, x} == b{n, x}
      □ a == b
      \Box a = b
15. Pour compiler un programme prog.c, on utilise la
    ligne de commande:
      ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc -Wall prog.c -o prog.exe
      ☐ gcc -Wall prog.exe -o prog.c
16. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
    {
      printf("coucou\n");
     \square il ne compile pas
      \square il comporte une boucle infinie
      □ il n'affiche rien
      ☐ il risque d'afficher bonjour à la place de coucou
17. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
      \square (A == TRUE) && (B == TRUE)
      \square (!A || B)
      □ A && B
      \square !(!A || B) == (A && !B)
18. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
      \square int pgcd(int x, int x);
     \Box int pgcd(int y, int x);
     \square void pgcd(int x, int y);
      \square int pgcd(int x, y);
```

```
19. Soit la fonction f définie par :
    int f(int a)
    ₹
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
     \Box 1
      \Box 0
      \Box 5
      \Box 4
20. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 5
      \Box 0
```

 $\Box$  7

т		-1
	acence	

## Éléments d'informatique - contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. L'écriture 101 en binaire correspond au nombre naturel:  $\square$  101  $\Box$  4  $\square$  3  $\Box$  5 2. L'ordonnancement par tourniquet permet : ☐ de ne pas perdre de temps avec la commutation de contexte ☐ d'entretenir l'illusion que les processus tournent en parallèle □ d'afficher des ronds colorés à l'écran □ de doubler la mémoire disponible 3. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit: □ void afficher\_date(struct date\_s d); □ void afficher\_date(date\_s d); □ struct date\_s afficher\_date(struct date\_s d) ☐ int afficher\_date(date\_s d); 4. Le code suivant : int age = 20; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); } affichera: □ Mineur  $\square$  rien □ Majeur □ Mineur  $\square$  0 Majeur

5.	Dans la commande gcc, l'option -Wall signifie :	
	$\hfill\Box$ que l'on veut voir tous les avertissements	
	$\Box$ qu'il faut lancer un déboggueur	
	$\hfill\Box$ qu'il faut indenter le fichier source	
	$\Box$ qu'on veut changer alétoirement de fond d'écran	
6.	Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :	
	$\square$ printf("x=%d et y=%d\n,x,y");	
	$\square$ printf("x=%d et y=%d\n",x y);	
	$\square$ printf("x=%d et y=%d\n",x,y);	
	$\square$ printf("x=%x et y=%y\n");	
7.	Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?	
	$\Box$ un désaccord entre la déclaration et la définition d'une fonction	
);	$\Box$ une directive préprocesseur $\# {\tt include}$ man quante	
•	$\Box$ une fonction appelée avant sa déclaration	
	$\Box$ une fonction déclarée mais non définie	
8.	Soit la fonction g définie par :	
	<pre>int g(int a) {</pre>	
	<pre>printf("a = \n", %d); if (1 &gt; 0) {</pre>	
	return 5;	
	return 7;	
	}	
	Alors l'expression $g(0)$ prendra la valeur :	
	□ 7	
	□ 5	
	,	

```
9. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
      □ saisie_utilisateur(scanf(%d));
      □ void saisie_utilisateur(int n);
     □ void saisie_utilisateur(char c);
      ☐ int saisie_utilisateur();
10. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
      \square int pgcd(int y, int x);
     \square int pgcd(int x, int x);
     \Box int pgcd(int x, y);
      \square void pgcd(int x, int y);
11. Un enregistrement permet de grouper plusieurs valeurs
   dans:
      \square ses chants
     \square ses champs
     \square ses blocs
      □ ses cases
12. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
      {
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 4
     \square 8
     \Box 16
     \square 0
13. Un bit est:
      □ la longueur d'un mot mémoire
      □ un battement d'horloge processeur
      ☐ l'instruction qui met fin à un programme
```

 $\square$  un chiffre binaire (0 ou 1)

14. Vous utilisez une boucle while quand:	$\Box$ la variable x vaut 0	}
$\Box$ vous avez déjà fait un for dans le même pro-	$\square$ la variable y vaut 5	else
gramme principal	☐ le programme affiche "Faux"	{
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	$\Box$ la variable x vaut 5 et la variable y vaut 0	<pre>printf("Majeur\n"); }</pre>
$\hfill \square$ l'incrément de la variable de boucle n'est pas $1$	18. Le code suivant :	
$\hfill \square$ vous n'avez pas déclaré de fonction	int age = 15;	affichera:
15. Quel est l'opérateur de différence en C :	if (age < 18)	☐ Mineur
□ !	{	☐ Mineur
	<pre>printf("Mineur\n");</pre>	Majeur
□ !=	else	□ rien
□ ≠	{	
16. Après la déclaration : int mccarthy(int n);, il est	<pre>printf("Majeur\n");</pre>	□ Majeur
correct d'écrire :	}	20. Si a et b sont deux variables de type :
$\square$ x = mccarthy(n);		struct toto_s
□ n = mccarthy();	affichera:	{
$\Box$ n = mccarthy(p, q);	☐ Mineur	int n;
☐ int mccarthy(int 2);	Majeur	<pre>double x;</pre>
17. Après exécution jusqu'à la ligne 15 du programme C :	□ Majeur	};
10 int main() {	□ rien	Alors pour tester l'égalité de a et de b on utilise la
11 int $x = 5$ ;	☐ Mineur	condition:
12 int y; 13	19. Le code suivant :	□ a = b
13 14 y = x;	int age = 18;	$\square (a.n == b.n) \&\& (a.x == b.x)$
15	if (age < 18)	□ a == b
16	{	$\square \ a\{n, x\} == b\{n, x\}$
17 }	<pre>printf("Mineur\n");</pre>	$\sqcup$ ain, x; == bin, x;

## Éléments d'informatique – contrôle continue

}

Prénom: Nom	:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit la fonction f définie par :

```
int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
  Alors l'expression f(1) prendra la valeur :
    \Box 5
     \Box 1
    \Box 0
     \Box 4
2. Un enregistrement permet de grouper plusieurs valeurs
  dans:
     \square ses champs
     \square ses blocs
     \square ses chants
     □ ses cases
3. Quelle étape de la compilation vient d'échouer lors-
  qu'on a un message comme celui-ci :
  Undefined symbols :"_prinft" ou
  référence indéfinie vers « prinft »
     □ l'édition de liens
    ☐ l'analyse des entrées clavier
    ☐ l'analyse sémantique
     □ l'analyse harmonique
4. Vous utilisez une boucle while quand :
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous avez déjà fait un for dans le même pro-
       gramme principal
    \squarevous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
```

□ vous n'avez pas déclaré de fonction

```
5. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
  préalable, on utilise plutôt :
    \square for(i=1;i<=5;i=i+1)
    \square for(i=0;i<=5;i=i+1)
    \square for(i=0;i<5;i=i+1)
    \square for(i=1;i<5;i=i+1)
6. Soit le programme principal suivant :
   int main()
    int a = 3:
    int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
  }
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
7. Si x est une variable réelle (de type double) alors
  x = 3/2 lui affecte la valeur :
    \square 0.5
    \square 1.5
    \Box 1
     \Box 0
8. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
```

```
else
     ₹
          printf("Majeur\n");
   affichera:
     □ Majeur
     ☐ Mineur
        Majeur
     \square rien
      □ Mineur
 9. La virtualisation de la mémoire permet notamment de
    stocker des portions inactives de la mémoire de travail
   sur le disque dur. Mais on perd :
      \square des processus
      \square les fichiers du disque
      □ en temps d'accès
      □ certaines données de la mémoire de travail
10. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     ☐ printf("Valeur de n ? %d\n", n);
     □ un débogueur
     ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
11. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
     \square 8
     \Box 4
      \Box 0
```

 $\Box$  16

```
12. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box j = 5
     \Box j = 0
     \Box j = %d
     \Box j = 4
13. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ créer un fichier texte
     □ créer un répertoire
     □ ouvrir un fichier texte
14. Si le code :
   struct toto_s
      int n;
      double x;
   précède la fonction main(), alors on peut écrire en
   début de main() :
```

```
□ struct toto_s toto;
      \square int toto.n = 3;
      \square toto_s n, x;
      \square toto_s struct z = {3, 0.5};
      \Box int struct toto_s = {3, -1e10};
15. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \Box 7
      \square 3
      \square 111
      \square 8
16. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
         return 3;
      }
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 4
      \Box 0
17. Un fichier source est:
      □ un document de référence du système
      \square un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
```

```
□ un fichier texte qui sera traduit en instructions
        processeur
      □ un document illisible pour les humains
      □ un document qui doit être protégé
18. Pour compiler un programme prog.c, on utilise la
    ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
19. Le code suivant :
     int i;
     for (i = 4; i \ge 0; i = i - 1)
         printf("%d ", i);
     printf("\n");
   affichera:
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 01234
     \square 4 3 2 1
     \square 1 2 3 4
20. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \square int pgcd(int x, int x);
     \square int pgcd(int y, int x);
     \square void pgcd(int x, int y);
     \square int pgcd(int x, y);
```

# Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		
		_

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?  $\square$  une accolade manquante  $\square$  une accolade en trop ☐ un point-virgule manquant □ un point-virgule en trop 2. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.  $\square$  #define taille = 3 ☐ #define taille = N  $\square$  #define N = 3 □ #define N 3 3. Vous utilisez une boucle while quand:  $\square$  vous ne connaissez pas le nombre d'itérations de la boucle à l'avance ☐ l'incrément de la variable de boucle n'est pas 1  $\square$ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction 4. L'écriture 111 en binaire correspond au nombre naturel:  $\square$  3  $\square$  8  $\Box$  7  $\square$  111 5. Quel est le problème d'un programme comportant les lignes suivantes? while (1) printf("coucou\n"); □ il comporte une boucle infinie  $\square$  il ne compile pas ☐ il risque d'afficher bonjour à la place de coucou  $\square$  il n'affiche rien

```
6. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \square 0
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     ☐ int char;
     ☐ char "c";
     □ char c;
     □ char 'c';
8. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ ouvrir un fichier texte
     □ créer un répertoire
     \square créer un fichier texte
9. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
 11
           int x = 5;
 12
           int y = 3;
 13
 14
           x = y;
 15
 16
      }
 17
     ☐ le programme affiche "Faux"
     \square la variable x vaut 5 et la variable y vaut 3
     \square la variable y vaut 5
     \square la variable x vaut 3
```

```
10. L'ordonnancement par tourniquet permet :
      □ d'afficher des ronds colorés à l'écran
      □ de doubler la mémoire disponible
      ☐ d'entretenir l'illusion que les processus tournent
         en parallèle
      ☐ de ne pas perdre de temps avec la commutation
         de contexte
11. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
          printf("%d ", i);
     printf("\n");
    affichera:
      \Box 02468
      \Box 0 1 2 3 4 5 6 7
      \Box 0246
      \Box 0 1 2 3 4 5 6
12. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
      \square une directive préprocesseur #include manquante
      \square une fonction appelée avant sa déclaration
      □ un désaccord entre la déclaration et la définition
         d'une fonction
      □ une fonction déclarée mais non définie
13. Après exécution jusqu'à la ligne 15 du programme C :
  10
 11
       int main() {
  12
            int x = 5;
  13
  14
            x = 3 * x + 1;
  15
  16
  17
      }
      □ le programme affiche ****
      \square la variable x vaut -\frac{1}{2}
      \Box la variable x vaut 16
      \square le programme affiche x
```

tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :    alphabétique   un ordre quelconque   dans lequel vous avez déclaré ces fonction   dans lequel ces fonctions sont appelées dans le main     Un enregistrement permet de grouper plusieurs valeurs dans :   ses champs   ses champs   ses chants   ses blocs   ses cases     Soit la fonction f définie par : int f(int a) {   printf("a = \n", %d); if (a > 0) {   return f(a - 1) + 1; }   return 4; }	4.	Vous avez déclaré préalablement un ensemble de fonc-
est l'ordre :  □ alphabétique □ un ordre quelconque □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main  5. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses champs □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par : int f(int a) {    printf("a = \n", %d);    if (a > 0)    {       return f(a - 1) + 1;    }    return 4;		tions utilisées par votre programme principal. L'ordre
□ alphabétique □ un ordre quelconque □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main  5. Un enregistrement permet de grouper plusieurs valeurs dans: □ ses champs □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par: int f(int a) {    printf("a = \n", %d);    if (a > 0)    {       return f(a - 1) + 1;    }    return 4;		dans lequel vous devez maintenant définir ces fonctions
<ul> <li>□ un ordre quelconque</li> <li>□ dans lequel vous avez déclaré ces fonction</li> <li>□ dans lequel ces fonctions sont appelées dans le main</li> <li>5. Un enregistrement permet de grouper plusieurs valeurs dans :</li> <li>□ ses champs</li> <li>□ ses chants</li> <li>□ ses blocs</li> <li>□ ses cases</li> <li>6. Soit la fonction f définie par :</li> <li>int f(int a)</li> <li>{</li> <li>printf("a = \n", %d);</li> <li>if (a &gt; 0)</li> <li>{</li> <li>return f(a - 1) + 1;</li> <li>}</li> <li>return 4;</li> </ul>		est l'ordre :
<ul> <li>□ dans lequel vous avez déclaré ces fonction</li> <li>□ dans lequel ces fonctions sont appelées dans le main</li> <li>5. Un enregistrement permet de grouper plusieurs valeurs dans :</li> <li>□ ses champs</li> <li>□ ses chants</li> <li>□ ses blocs</li> <li>□ ses cases</li> <li>6. Soit la fonction f définie par :</li> <li>int f(int a)</li> <li>{</li> <li>printf("a = \n", %d);</li> <li>if (a &gt; 0)</li> <li>{</li> <li>return f(a - 1) + 1;</li> <li>}</li> <li>return 4;</li> </ul>		□ alphabétique
□ dans lequel ces fonctions sont appelées dans le main  5. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses champs □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par : int f(int a) {    printf("a = \n", %d);    if (a > 0)    {       return f(a - 1) + 1;    }    return 4;		$\Box$ un ordre quel conque
main  5. Un enregistrement permet de grouper plusieurs valeurs dans:  □ ses champs □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par: int f(int a) {  printf("a = \n", %d); if (a > 0) {  return f(a - 1) + 1; } return 4;		$\Box$ dans lequel vous avez déclaré ces fonction
<pre>5. Un enregistrement permet de grouper plusieurs valeurs dans:</pre>		$\square$ dans lequel ces fonctions sont appelées dans le
<pre>dans:</pre>		main
□ ses champs □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par : int f(int a) {    printf("a = \n", %d);    if (a > 0)    {       return f(a - 1) + 1;    }    return 4;	5.	Un enregistrement permet de grouper plusieurs valeurs
<pre> □ ses chants □ ses blocs □ ses cases  6. Soit la fonction f définie par :    int f(int a)    {       printf("a = \n", %d);       if (a &gt; 0)       {         return f(a - 1) + 1;       }       return 4; </pre>		dans:
<pre> □ ses blocs □ ses cases  6. Soit la fonction f définie par :   int f(int a)   {     printf("a = \n", %d);     if (a &gt; 0)     {        return f(a - 1) + 1;     }     return 4; </pre>		$\square$ ses champs
☐ ses cases  6. Soit la fonction f définie par :  int f(int a) {     printf("a = \n", %d);     if (a > 0)    {       return f(a - 1) + 1;    }     return 4;		$\square$ ses chants
<pre>6. Soit la fonction f définie par :     int f(int a)     {         printf("a = \n", %d);         if (a &gt; 0)         {             return f(a - 1) + 1;         }         return 4;</pre>		$\square$ ses blocs
<pre>int f(int a) {   printf("a = \n", %d);   if (a &gt; 0)   {     return f(a - 1) + 1;   }   return 4;</pre>		□ ses cases
<pre>{   printf("a = \n", %d);   if (a &gt; 0)   {     return f(a - 1) + 1;   }   return 4;</pre>	6.	Soit la fonction f définie par :
<pre>printf("a = \n", %d); if (a &gt; 0) {     return f(a - 1) + 1; } return 4;</pre>		int f(int a)
<pre>if (a &gt; 0) {     return f(a - 1) + 1; } return 4;</pre>		{
<pre>{    return f(a - 1) + 1; } return 4;</pre>		<pre>printf("a = \n", %d);</pre>
return f(a - 1) + 1; } return 4;		if (a > 0)
<pre>} return 4;</pre>		{
return 4;		return f(a - 1) + 1;
		}
}		
		}

```
Alors l'expression f(1) prendra la valeur :
      \Box 4
      \Box 5
      \Box 1
      \Box 0
17. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      \square une fonction déclarée mais non définie
      □ une directive préprocesseur #include manquante
      \Box un désaccord entre la déclaration et la définition
         d'une fonction
      \Box\,une fonction appelée avant sa déclaration
18. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ l'avoir déclarée
      ☐ l'avoir définie
     □ avoir défini une constante symbolique de la taille
         de cette fonction
```

```
19. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
    {
        for (j = 0; j < 3; j = j + 1)
        {
            printf("%d ", i);
        }
    }

qu'est ce qui sera affiché?
    □ 1 2 1 2 3
    □ 0 0 0 1 1 1
    □ 0 1 2 0 1 2
    □ 0 1 0 1 0 1 0 1</pre>
```

20. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :

```
□ printf("x=%x et y=%y\n");
□ printf("x=%d et y=%d\n",x,y);
□ printf("x=%d et y=%d\n,x,y");
□ printf("x=%d et y=%d\n",x y);
```

## Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Quel est l'opérateur de différence en C :
    \Box \neq
     \sqcap \iff
     □ !=
     □ !
2. Si factorielle est une fonction prenant en entrée un
  entier et renvoyant un entier, il est correct d'écrire :
    ☐ printf("%d", factorielle(n));
     ☐ int factorielle(int 2);
    \square n = factorielle():
    \square n = factorielle(p, q);
3. Le code suivant :
    int i:
    for (i = 0; i < 5; i = i + 1)
        printf("%d ", i);
   printf("\n");
  affichera:
     \Box 0123
    \square 4 3 2 1
    \Box \ 4\ 3\ 2\ 1\ 0
    \Box 01234
4. Au début de la fonction main() on place le code :
    char i:
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
  Alors l'affichage sera:
    \square A
    Πi
     ☐ ABCDEF
```

```
5. Soit la fonction f définie par :
  int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
  Alors l'expression f(0) prendra la valeur :
    \Box 0
    \Box 4
    \square 3
6. Le bus système sert à :
    ☐ Écrire des données sur le dique dur
    ☐ Arriver à l'heure en cours
    ☐ Transférer des données et intructions entre pro-
       cesseur et mémoire
    □ transporter les processus du tourniquet au pro-
       cesseur
7. Soit le programme principal suivant :
  int main()
   int a = 3;
   int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
```

 $\Box$  f(a,b)=8, a=8, b=5

```
8. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
    Quelle est la condition cond :
     □ (a<=n) && (n<=b)
      \square (a<n) || (n>b)
      □ (n<=a) && (n<=b)
      □ a<=n<=b
 9. Sur un ordinateur avec un seul processeur, habituelle-
    ment les processus sont exécutés :
     □ tour à tour, un petit peu à chaque fois
     □ en parallèle, chacun dans un registre
     □ chacun son tour, après que le processus précédent
        a terminé
     \square tous ensemble
10. Le type des réels en C est :
      □ real
     \square int
     □ char
     □ double
11. Pour compiler un programme prog.c, on utilise la
    ligne de commande :
      ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc -Wall prog.c -o prog.exe
```

☐ gcc -Wall prog.exe -o prog.c

12. Si pgcd est une fonction prenant en entrée deux entiers	<pre>     for(i=1;i&lt;=5;i=i+1) </pre>	$\sqcup$ (a.n == b.n) && (a.x == b.x)
et renvoyant un entier, il est correct d'écrire :	☐ for(i=0;i<5;i=i+1)	□ a = b
$\square$ int n = pgcd();	☐ for(i=0;i<=5;i=i+1)	□ a == b
$\square$ n = pgcd(int p, int q);	16. Vous utilisez une boucle while quand:	$\square a\{n, x\} == b\{n, x\}$
$\square$ n = pgcd(n, 3);	$\square$ l'incrément de la variable de boucle n'est pas 1	
☐ int pgcd(2);	□ vous ne connaissez pas le nombre d'itérations de	19. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est
13. Si cet avertissement apparaı̂t à la compilation :	la boucle à l'avance	une variable entière définie et initialisée, il est correct
warning: implicit declaration of function 'max	' □ vous avez déjà fait un for dans le même pro-	d'écrire :
, que doit-on chercher dans le programme?	gramme principal	☐ int carre(2);
☐ une directive préprocesseur #include manquante	$\square$ vous n'avez pas déclaré de fonction	<pre>□ int n = carre();</pre>
☐ une fonction appelée avant sa déclaration	17. Avant de faire appel à une fonction il est nécessaire	$\square$ n = carre(n);
$\square$ une fonction déclarée mais non définie	de:	· · · •
un désaccord entre la déclaration et la définition	☐ l'avoir définie	$\square$ n = carre(int n);
d'une fonction	☐ l'avoir déclarée et définie	20. Une segmentation fault est une erreur qui survient
14. Pour déclarer une procédure afficher_menu sans ar-	□ l'avoir déclarée	lorsque:
gument et qui ne renvoie rien on utilise :	$\square$ avoir défini une constante symbolique de la taille	$\Box$ le programme tente d'afficher des caractères sur
☐ char afficher_menu(printf("menu"));	de cette fonction	une ligne qui va au delà de la largeur de la fenêtre
□ void afficher_menu();	18. Si a et b sont deux variables de type:	du terminal
☐ double afficher_menu();	struct toto_s	☐ le programme source a été enregistré sur le disque
☐ int afficher_menu(int char);	{	dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur
☐ int afficher_menu();	int n;	
15. Pour afficher à l'aide de printf("%d\n",tab[i]);	double x;	☐ la division du programme en zones homogènes échoue
le contenu d'un tableau de 5 entiers initialisé au	};	
préalable, on utilise plutôt :	Alors pour tester l'égalité de a et de b on utilise la	☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée
☐ for(i=1;i<5;i=i+1)	condition:	memone qui ne iui est pas reservee

т		-1
	100000	- 1
	acence	

## Éléments d'informatique – contrôle continue

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Au début de la fonction main() on place le code : char b = 'A';b = b + 2;printf("%c\n", b); Alors l'affichage sera:  $\Box$  A □ b  $\Box$  C □В 2. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise : □ void afficher\_menu(); ☐ int afficher menu(): ☐ char afficher\_menu(printf("menu")); ☐ int afficher menu(int char): □ double afficher\_menu(); 3. Le bus système sert à : □ transporter les processus du tourniquet au processeur ☐ Arriver à l'heure en cours ☐ Écrire des données sur le dique dur □ Transférer des données et intructions entre processeur et mémoire 4. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #appart <stdlib.h> ☐ #include <studio.h>

☐ #include <studlib.h>

☐ #include <stdio.h>

5. Pour l'extrait de programme suivant : int i; int j; for(i=4;i>0;i=i-1) for(j=i;j<6;j=j+1) printf("\*"); printf(" "); qu'est ce qui sera affiché? 6. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :  $\square$  n = pgcd(int p, int q);  $\square$  int n = pgcd();  $\square$  n = pgcd(n, 3);  $\square$  int pgcd(2); 7. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée: ☐ retourner un bloc □ sélectionner entre deux blocs à l'aide d'une condition □ répéter un bloc tant qu'une condition est vérifée  $\square$  mettre les blocs en séquence les uns à la suite des autres 8. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?  $\square$  (!A | | B) ☐ (A == TRUE) && (B == TRUE)

□ A && B

 $\square$  !(!A || B) == (A && !B)

```
9. Après exécution jusqu'à la ligne 15 du programme C:
  10
       int main() {
 11
             int x = 5:
  12
            int y = 3;
  13
 14
            x = y;
  15
  16
      }
 17
      \square la variable x vaut 3
      □ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 3
      \square la variable y vaut 5
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il ne compile pas
      □ il n'affiche rien
      □ il risque d'afficher bonjour à la place de coucou
      \square il comporte une boucle infinie
11. Un bit est:
      \square un chiffre binaire (0 ou 1)
      \square un battement d'horloge processeur
      ☐ l'instruction qui met fin à un programme
      \Box la longueur d'un mot mémoire
12. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x = racine(racine(x)*racine(x));
      \square x = racine(2/3);
      \square x - 1 = racine(x);
```

 $\square$  x = racine(x \* x) - racine(x):

```
13. Après exécution jusqu'à la ligne 14 du programme C:
                                                                   ☐ Mineur
                                                                                                                              appelant la fonction f ainsi définie :
                                                                   ☐ Mineur
       int main() {
  10
                                                                                                                              int f(int a, int b)
                                                                      Majeur
  11
            int x = 5;
                                                                   \square rien
  12
                                                                                                                                 a = a + b;
  13
            printf(" x = %d\n", 2);
                                                                   □ Majeur
                                                                                                                                 return a;
  14
                                                             16. Le code suivant :
  15
                                                                  int i;
                                                                                                                              L'affichage dans le main est le suivant :
  16
       }
                                                                  for (i = 1; i < 5; i = i + 1)
                                                                                                                                \Box f(3,5)=8, a=3, b=5
      \square le terminal affiche x = 2
                                                                       printf("%d ", i);
                                                                                                                                \Box f(a,b)=8, a=8, b=5
      \square le terminal affiche 5
                                                                  }
                                                                                                                                \Box f(a,b)=13, a=8, b=5
      \square le terminal affiche x = 5
                                                                  printf("\n");
                                                                                                                                \Box f(a,b)=8, a=3, b=5
      □ le terminal affiche "Faux"
                                                                 affichera:
                                                                   \square 1 2 3 4
                                                                                                                          19. Pour déclarer une fonction saisie_utilisateur qui
14. Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                              demande à l'utilisateur d'entrer un entier au clavier et
                                                                   \square 4 3 2 1
    tions utilisées par votre programme principal. L'ordre
                                                                                                                              renvoie cet entier on écrit :
   dans lequel vous devez maintenant définir ces fonctions
                                                                   \Box 01234
    est l'ordre :
                                                                   \Box \ 4\ 3\ 2\ 1\ 0
                                                                                                                                □ void saisie_utilisateur(int n);
      □ dans lequel ces fonctions sont appelées dans le
                                                             17. Si x est une variable réelle (de type double) alors
                                                                                                                                ☐ int saisie_utilisateur();
         main
                                                                 x = 3/2 lui affecte la valeur :
                                                                                                                                □ void saisie_utilisateur(char c);
      \square dans lequel vous avez déclaré ces fonction
                                                                   \square 0.5
                                                                                                                                ☐ saisie_utilisateur(scanf(%d));
                                                                   \Box 0
      \square un ordre quelconque
                                                                   \square 1.5
      □ alphabétique
                                                                                                                          20. Un fichier source est:
                                                                   \Box 1
                                                                                                                                □ un fichier texte qui sera traduit en instructions
15. Le code suivant :
                                                             18. Soit le programme principal suivant :
                                                                                                                                   processeur
     int age = 20;
                                                                 int main()
                                                                                                                                □ un document illisible pour les humains
     if (age < 18)
                                                                 {
                                                                                                                                \square un fichier que l'ont doit citer dans les documents
                                                                  int a = 3;
          printf("Mineur\n");
                                                                                                                                   produits sur l'ordinateur
                                                                  int b = 5;
                                                                  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                                □ un document qui doit être protégé
     printf("Majeur\n");
                                                                  return EXIT_SUCCESS;
                                                                                                                                \square un document de référence du système
    affichera:
```

т.	-1
Licence	

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise : ☐ int afficher\_menu(int char); ☐ double afficher\_menu(); ☐ char afficher\_menu(printf("menu")); □ void afficher\_menu(); ☐ int afficher\_menu(); 2. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :  $\square$  un débogueur  $\square$  printf("Valeur de n ? %g\n", n); □ scanf("%d", &n); ☐ printf("Valeur de n ? %d\n", n); 3. L'ordonnancement par tourniquet permet :  $\square$  de ne pas perdre de temps avec la commutation de contexte ☐ d'entretenir l'illusion que les processus tournent en parallèle  $\square$  de doubler la mémoire disponible □ d'afficher des ronds colorés à l'écran 4. Soit le programme principal suivant : int main() int a = 3; int b = 5: printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b$ ); return EXIT\_SUCCESS; appelant la fonction f ainsi définie : int f(int a, int b) a = a + b; return a;

L'affichage dans le main est le suivant :

```
\Box f(a,b)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=13, a=8, b=5
5. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
 11
           int x = 5;
 12
           int y = 3;
 13
 14
           x = y;
 15
 16
      }
 17
     \square la variable v vaut 5
     ☐ le programme affiche "Faux"
     \square la variable x vaut 3
     \square la variable x vaut 5 et la variable y vaut 3
6. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \square 101
     \Box 5
     \Box 4
     \square 3
7. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit:
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
     □ void saisie_utilisateur(int n);
     ☐ int saisie_utilisateur();
8. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
```

	Alor	s l'expression g(0) prendra la valeur :
		5
		0
		7
9.		nelle de ces écritures correspond à la déclaration e variable de type caractère en langage C?
		char 'c';
		char "c";
		int char;
		char c;
10.	Le t	ype des réels en C est :
		real
		double
		char
		int
11.	erre	ette erreur apparaît à la compilation : eur: conflicting types for 'max', que doit- hercher dans le programme?
		une fonction appelée avant sa déclaration
		une fonction déclarée mais non définie
		un désaccord entre la déclaration et la définition d'une fonction
		une directive préprocesseur $\verb"#include"$ manquante
12.		s calculs peut-on programmer en programmation eturée?
		en programmation structurée on peut programmer tous les calculs programmables en langage machine
		certains programmes sont de vrais plats de spaghetti
		il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
		il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée

<pre>□ j = 4 □ j = %d □ j = 5 Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande : □ printf("x=%d et y=%d\n",x y); □ printf("x=%x et y=%y\n"); □ printf("x=%d et y=%d\n,x,y"); □ printf("x=%d et y=%d\n,x,y"); Pour déclarer un tableau d'entiers de taille 5, on peut</pre>	printf("%d ", i); }  printf("\n");  qu'est ce qui sera affiché?  □ 0 1 2 0 1 2  □ 1 2 3 1 2  □ 0 0 1 1 2 2  □ 0 1 0 1 0 1  19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
☐ j = 5  Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :  ☐ printf("x=%d et y=%d\n",x y);  ☐ printf("x=%x et y=%y\n");  ☐ printf("x=%d et y=%d\n,x,y");  ☐ printf("x=%d et y=%d\n,x,y);	<pre>printf("\n");   qu'est ce qui sera affiché?</pre>
Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :  □ printf("x=%d et y=%d\n",x y); □ printf("x=%x et y=%y\n"); □ printf("x=%d et y=%d\n,x,y"); □ printf("x=%d et y=%d\n",x,y);	<pre>printf("\n");   qu'est ce qui sera affiché?</pre>
<pre>5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :</pre>	☐ 0 1 2 0 1 2 ☐ 1 2 3 1 2 ☐ 0 0 1 1 2 2 ☐ 0 1 0 1 0 1  19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
<pre>5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :</pre>	☐ 0 1 2 0 1 2 ☐ 1 2 3 1 2 ☐ 0 0 1 1 2 2 ☐ 0 1 0 1 0 1  19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
<pre>□ printf("x=%x et y=%y\n"); □ printf("x=%d et y=%d\n,x,y"); □ printf("x=%d et y=%d\n",x,y);</pre>	□ 0 0 1 1 2 2 □ 0 1 0 1 0 1  19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
<pre>□ printf("x=%d et y=%d\n,x,y");</pre> □ printf("x=%d et y=%d\n",x,y);	□ 0 1 0 1 0 1  19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
<pre>□ printf("x=%d et y=%d\n,x,y");</pre> □ printf("x=%d et y=%d\n",x,y);	19. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
$\square \   \text{printf("x=\%d et y=\%d\n",x,y);}$	étapes de la compilation :
Pour déclarer un tableau d'entiers de taille 5, on peut	
1	$\square$ analyse harmonique
utiliser l'instruction	□ analyse lexicale
	$\Box$ analyse syntaxique
,	$\Box$ analyse sémantique
· ·	20. Un registre du processeur est :
☐ int toto[5];	_
$\square$ int tab[] = 5;	☐ une unité de calcul spécialisée de l'ordinateur
Pour l'extrait de programme suivant :	☐ une case mémoire interne au processeur qui sera manipulée directement lors des calculs
	$\hfill \square$ une gamme de fréquence de fonctionnement du
	processeur
{	☐ un composant qui contient la liste des fichiers du système
	<pre>□ int toto[taille=5]; □ char tableau[5]; □ int[] new tableau(5); □ int toto[5]; □ int tab[] = 5;  Pour l'extrait de programme suivant: int i = 0; int j = 0; for (i = 0; i &lt; 3; i = i + 1)</pre>

# Éléments d'informatique – contrôle continue

- 1		
Prénom:	Nom:	
Nº INE ·		
N° INE :		

Barème : 1 points par réponse juste (unique) ; $-0, 5$ points par réponse fausse. Durée : 20 minutes.	(
1. Un bit est:	
□ un chiffre binaire (0 ou 1)	
un battement d'horloge processeur	
☐ l'instruction qui met fin à un programme	
□ la longueur d'un mot mémoire	7
2. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :	
☐ void afficher_date(struct date_s d);	
☐ int afficher_date(date_s d);	
☐ void afficher_date(date_s d);	
☐ struct date_s afficher_date(struct date_s	d);
3. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :	
☐ int saisie_utilisateur();	
☐ saisie_utilisateur(scanf(%d));	
☐ void saisie_utilisateur(char c);	8
☐ void saisie_utilisateur(int n);	
4. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?	
$\square$ une fonction appelée avant sa déclaration	(
☐ un désaccord entre la déclaration et la définition d'une fonction	
$\square$ une directive préprocesseur #include manquante	
$\Box$ une fonction déclarée mais non définie	
5. Un fichier source est:	10
☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur	10
□ un document qui doit être protégé	
☐ un fichier texte qui sera traduit en instructions processeur	
$\Box$ un document illisible pour les humains	

 $\Box$  un document de référence du système

```
6. Pour déclarer une fonction pgcd qui calcule et renvoie
  le plus grand diviseur commun de deux entiers positifs
  passés en arguments on écrit :
    \Box int pgcd(int x, y);
    \square int pgcd(int y, int x);
    \Box int pgcd(int x, int x);
    \square void pgcd(int x, int y);
7. Le code suivant :
   int age = 20;
   if (age < 18)
   {
        printf("Mineur\n");
   }
   printf("Majeur\n");
  affichera:
    ☐ Mineur
       Majeur
    ☐ Mineur
    \square rien
    □ Majeur
8. Le type des réels en C est :
    □ double
    \square int
    □ char
    \square real
9. L'écriture 111 en binaire correspond au nombre natu-
  rel:
    □ 111
    \square 7
    \square 8
    \square 3
0. Le code suivant :
   int somme = 0;
   int i;
   for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
   printf("%d", somme);
```

```
affichera:
     \Box 0
     \square 42
     \Box 1
      \Box 6
11. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 10
     \Box 6
     \Box 0
     \square 15
12. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      □ struct int factorielle(int n);
     ☐ int factorielle(int x);
      ☐ int factorielle(double n):
      ☐ int factorielle();
13. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
      ☐ #include <studio.h>
     ☐ #include <studlib.h>
      ☐ #include <stdio.h>
      ☐ #appart <stdlib.h>
```

```
16. Pour compiler un programme prog.c, on utilise la
14. Soit la fonction g définie par :
                                                                                                                           □ un composant qui contient la liste des fichiers du
                                                              ligne de commande :
   int g(int a)
                                                                ☐ gcc prog.exe -Wall -o prog.c
                                                                ☐ gcc -Wall prog.exe -o prog.c
      printf("a = \n", %d);
      if (1 > 0)
                                                                ☐ gcc -Wall prog.c -o prog.exe
      {
                                                                ☐ gcc prog.c -o -Wall prog.exe
        return 5;
                                                          17. Pour l'extrait de programme suivant :
      return 7;
                                                                int produit = 0;
                                                                 int serie[4] = \{2, 2, 2, 2\};
                                                                 for (i = 0; i < 4; i = i + 1)
   Alors l'expression g(0) prendra la valeur :
     \Box 7
                                                                   produit = produit * serie[i];
     \Box 0
                                                                 printf("produit = %d", produit);
     \Box 5
                                                              La valeur affichée est :
15. Le bus système sert à :
                                                                \square 8
     ☐ Écrire des données sur le dique dur
                                                                \Box 4
     ☐ Arriver à l'heure en cours
                                                                \Box 0
     ☐ Transférer des données et intructions entre pro-
                                                                \Box 16
        cesseur et mémoire
                                                          18. Un registre du processeur est :
     \square transporter les processus du tourniquet au pro-
                                                                □ une case mémoire interne au processeur qui sera
        cesseur
                                                                                                                           \Box char c;
                                                                   manipulée directement lors des calculs
```

	système
	$\Box$ une gamme de fréquence de fonctionnement du processeur
	$\Box$ une unité de calcul spécialisée de l'ordinateur
19.	Quel est le problème d'un programme comportant les lignes suivantes?
	<pre>while (1) {    printf("coucou\n"); }</pre>
	<ul> <li>□ il risque d'afficher bonjour à la place de coucou</li> <li>□ il n'affiche rien</li> <li>□ il ne compile pas</li> <li>□ il comporte une boucle infinie</li> </ul>
20.	Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C ?
	☐ int char;
	□ char 'c';
	□ char "c";
	□ -1

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un fichier source est: □ un document qui doit être protégé □ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur  $\square$  un document de référence du système □ un fichier texte qui sera traduit en instructions processeur  $\square$  un document illisible pour les humains 2. Le langage C est un langage □ interprété □ lu, écrit, parlé □ composé □ compilé 3. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :  $\square$  for(i=1;i<5;i=i+1)  $\square$  for(i=0;i<=5;i=i+1)  $\square$  for(i=0;i<5;i=i+1) ☐ for(i=1;i<=5;i=i+1) 4. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit: □ void saisie\_utilisateur(int n); ☐ int saisie\_utilisateur(); □ void saisie\_utilisateur(char c); □ saisie\_utilisateur(scanf(%d)); 5. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ alphabétique □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans le main

 $\square$  un ordre quelconque

```
6. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
  valeur de x^n on écrit :
    \square double exposant(double x, int n);
    \square exposant(double x, int n, int r);
    \square int exposant(double n, int x);
    \square void exposant(double x^n);
7. Une segmentation fault est une erreur qui survient
  lorsque:
    □ le programme source a été enregistré sur le disque
        dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
    □ le programme tente d'afficher des caractères sur
       une ligne qui va au delà de la largeur de la fenêtre
        du terminal
     □ la division du programme en zones homogènes
        échoue
     \square le programme tente d'accèder à une partie de la
       mémoire qui ne lui est pas réservée
8. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
    \Box 7
     \Box 5
     \square 0
9. Un registre du processeur est :
    □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
    \square une gamme de fréquence de fonctionnement du
        processeur
    □ une unité de calcul spécialisée de l'ordinateur
    □ un composant qui contient la liste des fichiers du
        système
```

```
10. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i:
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 6
     \Box 0
     \square 10
     \square 15
11. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
     \square n = pgcd(n, 3);
     \square n = pgcd(int p, int q);
     \square int pgcd(2);
12. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise:
     ☐ int afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu(int char);
     □ void afficher_menu();
     □ double afficher_menu();
13. Vous utilisez une boucle while quand:
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ l'incrément de la variable de boucle n'est pas 1
```

```
14. Le code suivant :
    int age = 15;
    if (age < 18)
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
     □ rien
     ☐ Mineur
        Majeur
     ☐ Mineur
     □ Majeur
15. Une variable booléenne est un variable :
     □ NaN (not a number, qui n'est pas un nombre)
     ☐ réelle positive
     □ à laquelle une valeur vient d'être affectée
     □ jamais nulle
     □ qui est vraie ou fausse
```

```
16. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
      □ char 'c';
      ☐ int char;
     ☐ char "c";
      \Box char c;
17. Sous unix (ou linux), la commande cd permet de :
      □ détruire un fichier
      □ jouer de la musique
     □ changer de répertoire courant
     □ récupérer un programme arrêté avec la commande
      □ ouvir un bureau partagé (common desktop)
18. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée
      □ l'avoir définie
     □ avoir défini une constante symbolique de la taille
         de cette fonction
     □ l'avoir déclarée et définie
```

```
19. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
          printf("%d ", i);
     }
     printf("\n");
    affichera:
     \Box 0246
     \Box 02468
     \Box 0 1 2 3 4 5 6 7
     \Box 0 1 2 3 4 5 6
20. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
     \square un désaccord entre la déclaration et la définition
         d'une fonction
      \square une fonction appelée avant sa déclaration
      \square une directive préprocesseur \#include manquante
```

 $\square$  une fonction déclarée mais non définie

т		-1
	acence	

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

	reponse lausse. Durce: 20 innutes.
1.	Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
	☐ for(i=1;i<=5;i=i+1)
	☐ for(i=1;i<5;i=i+1)
	☐ for(i=0;i<=5;i=i+1)
	☐ for(i=0;i<5;i=i+1)
2.	L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
	□ 111
	□ 8
	$\square$ 3
	□ 7
3.	Une variable booléenne est un variable :
	□ réelle positive
	$\Box$ qui est vraie ou fausse
	$\Box$ NaN (not a number, qui n'est pas un nombre)
	$\square$ jamais nulle
	$\Box$ à laquelle une valeur vient d'être affectée
4.	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	$\Box$ un point-virgule en trop
	$\Box$ un point-virgule man quant
	$\Box$ une accolade en trop
	$\Box$ une accolade manquante
5.	Un enregistrement permet de grouper plusieurs valeurs dans :
	$\square$ ses chants
	$\square$ ses champs

 $\square$  ses blocs

 $\square$  ses cases

```
6. Soit la fonction g définie par :
   int g(int a)
   {
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 5
     \square 7
    \Box 0
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     ☐ int char;
     \Box char c;
     □ char 'c';
    ☐ char "c";
8. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
     ☐ l'analyse des entrées clavier
     ☐ l'analyse sémantique
     ☐ l'analyse harmonique
     □ l'édition de liens
9. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <stdio.h>
     ☐ #include <studlib.h>
     ☐ #include <studio.h>
     \square #appart <stdlib.h>
```

```
10. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
   Quelle est la condition cond:
     □ (n<=a) && (n<=b)
     □ (a<=n) && (n<=b)
     □ a<=n<=b
     □ (a<n) || (n>b)
11. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 15
     \Box 10
     \Box 6
     \Box 0
12. Le type des réels en C est :
     □ char
     ☐ double
     □ real
```

 $\square$  int

```
13. Si a et b sont deux variables de type :
                                                               {
                                                                                                                              {
                                                                   printf("Majeur\n");
   struct toto_s
                                                               }
                                                                                                                              }
   {
                                                                                                                         }
      int n;
                                                                                                                         printf("j = %d\n", j);
                                                              affichera:
      double x;
   };
                                                                □ rien
                                                                                                                         }
   Alors pour tester l'égalité de a et de b on utilise la
                                                                □ Mineur
   condition:
                                                                □ Majeur
                                                                                                                        qu'est ce qui sera affiché?
     \square (a.n == b.n) && (a.x == b.x)
                                                                ☐ Mineur
                                                                                                                          \Box j = 5
     \Box a\{n, x\} == b\{n, x\}
                                                                   Majeur
                                                                                                                          \Box j = %d
     \Box a = b
                                                          16. Le code suivant :
                                                                                                                          \Box j = 0
     □ a == b
                                                               int age = 18;
                                                                                                                          \Box j = 4
14. Soit la fonction f définie par :
                                                               if (age < 18)
                                                                                                                    18. Afin de représenter la taille d'un tableau, définir une
   int f(int a)
                                                                                                                        constante symbolique N valant 3.
                                                                   printf("Mineur\n");
                                                                                                                          \square #define N = 3
                                                               }
      printf("a = \n", %d);
                                                               else
                                                                                                                          \square #define taille = 3
      if (a > 0)
                                                               {
                                                                                                                          ☐ #define taille = N
                                                                   printf("Majeur\n");
        return 3;
                                                                                                                          □ #define N 3
                                                               }
                                                                                                                    19. Dans la commande gcc, l'option -Wall signifie :
      return 4;
                                                              affichera:
                                                                                                                          □ qu'il faut lancer un déboggueur
                                                                ☐ Mineur
   Alors l'expression f(0) prendra la valeur :
                                                                                                                          ☐ que l'on veut voir tous les avertissements
                                                                □ rien
                                                                                                                          □ qu'on veut changer alétoirement de fond d'écran
     \square 3
     \Box 0
                                                                ☐ Mineur
                                                                                                                          □ qu'il faut indenter le fichier source
                                                                   Majeur
     \Box 4
                                                                                                                    20. Pour déclarer une procédure afficher_menu sans ar-
                                                                □ Majeur
                                                                                                                        gument et qui ne renvoie rien on utilise:
15. Le code suivant :
                                                          17. Soit un programme contenant les lignes suivantes :
                                                                                                                          □ void afficher_menu();
     int age = 15;
                                                                                                                          ☐ char afficher_menu(printf("menu"));
    if (age < 18)
                                                               int i = 0;
                                                               int j = 0;
                                                                                                                          ☐ int afficher_menu();
         printf("Mineur\n");
                                                               for (i = 0; i < 0; i = i + 1)
                                                                                                                          ☐ double afficher_menu();
                                                                                                                          ☐ int afficher_menu(int char);
                                                                   for (j = 0; j < 5; j = j + 1)
     else
```

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int somme = 0;
    int i;
    for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
    printf("%d", somme);
   affichera:
     \Box 6
     \square 42
     \Box 1
     \Box 0
2. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \Box int mccarthy(int 2);
     \square x = mccarthy(n);
     \square n = mccarthy();
     \square n = mccarthy(p, q);
3. Le code suivant :
    int i;
    for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
   printf("\n");
   affichera:
     \square 0 1 2 3
     \Box 01234
     \Box \ 4\ 3\ 2\ 1\ 0
```

 $\square$  4 3 2 1

```
4. Vous utilisez une boucle while quand :
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous n'avez pas déclaré de fonction
5. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     □ new TP4
     ☐ yppasswd
     ☐ kwrite TP4
     □ mkdir TP4
6. Le type des réels en C est :
     □ char
     \square real
     \square int
     □ double
7. Un fichier source est:
     □ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
     □ un fichier texte qui sera traduit en instructions
        processeur
     □ un document qui doit être protégé
     \square un document illisible pour les humains
     □ un document de référence du système
8. Après exécution jusqu'à la ligne 14 du programme C:
      int main() {
 10
           int x = 5;
 11
 12
 13
           printf(" x = %d\n", 2);
 14
 15
           . . .
      }
 16
     \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
     \square le terminal affiche x = 5
     \square le terminal affiche 5
```

```
9. Le code suivant :
     int age = 15;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
    affichera:
      □ Mineur
        Majeur
      □ Majeur
      □ Mineur
      □ rien
10. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      ☐ int factorielle(double n):
     \Box int factorielle(int x);
      □ struct int factorielle(int n);
      ☐ int factorielle();
11. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      \square dans lequel ces fonctions sont appelées dans le
         main
      □ alphabétique
      \square dans lequel vous avez déclaré ces fonction
      □ un ordre quelconque
12. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x = racine(racine(x)*racine(x));
     \square x = racine(2/3);
     \square x = racine(x * x) - racine(x);
```

 $\square$  x - 1 = racine(x);

```
13. Pour l'extrait de programme suivant :
                                                                 scanf("%d", &n);
                                                                                                                        19. Le bus système sert à :
                                                                 while(cond)
     int i = 0;
                                                                                                                              □ transporter les processus du tourniquet au pro-
     int j = 0;
                                                                   scanf("%d", &n);
                                                                                                                                 cesseur
     for (i = 0; i < 3; i = i + 1)
                                                                                                                              \square Arriver à l'heure en cours
                                                                Quelle est la condition cond :
         for (j = 0; j < 2; j = j + 1)
                                                                  \square (a<=n) && (n<=b)
                                                                                                                              ☐ Écrire des données sur le dique dur
              printf("%d ", i);
                                                                  □ a<=n<=b
         }
                                                                  \square (n<=a) && (n<=b)
                                                                                                                              ☐ Transférer des données et intructions entre pro-
                                                                                                                                 cesseur et mémoire
                                                                 ☐ (a<n) || (n>b)
     printf("\n");
                                                            16. Pour déclarer une fonction exposant qui prend en ar-
   qu'est ce qui sera affiché?
                                                                gument un réel x et un entier positif n et renvoie la
                                                                                                                        20. Soit la fonction g définie par :
                                                                valeur de x^n on écrit :
      \Box 0 1 2 0 1 2
                                                                 \square int exposant(double n, int x);
     \Box 1 2 3 1 2
                                                                                                                            int g(int a)
                                                                 ☐ void exposant(double x^n);
     \Box 0 0 1 1 2 2
                                                                 \square double exposant(double x, int n);
                                                                                                                              printf("a = \n", %d);
     \Box 0 1 0 1 0 1
                                                                                                                              if (1 > 0)
                                                                  \square exposant(double x, int n, int r);
14. Lorsqu'un programme utilise printf ou scanf il faut
                                                                                                                              {
                                                            17. Si n est une variable entière pour demander sa valeur
   qu'il contienne l'instruction préprocesseur :
                                                                                                                                 return 5;
                                                                à l'utilisateur, on utilise plutôt :
      ☐ #include <stdio.h>
                                                                                                                              }
                                                                  ☐ printf("Valeur de n ? %g\n", n);
                                                                                                                              return 7;
      ☐ #appart <stdlib.h>
                                                                 □ scanf("%d", &n);
                                                                 □ printf("Valeur de n ? %d\n", n);
      ☐ #include <studio.h>
                                                                  \square un débogueur
      ☐ #include <studlib.h>
                                                                                                                            Alors l'expression g(0) prendra la valeur :
                                                            18. Laquelle des analyses suivantes ne fait pas partie des
15. On souhaite faire une boucle de contrôle de saisie : tant
                                                                étapes de la compilation :
   que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                                                                              \Box 0
                                                                 \square analyse harmonique
   recommence la saisie de n. Soit le programme suivant :
                                                                 \square analyse syntaxique
                                                                                                                              \Box 5
     int a = 0;
                                                                 □ analyse sémantique
     int b = 20;
                                                                                                                              \Box 7
                                                                  \square analyse lexicale
     int n;
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

 $1.\ {\rm Si}$ a et b<br/> sont deux variables de type :

```
struct toto_s
{
  int n;
  double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition :

```
\square a = b
```

```
\square (a.n == b.n) && (a.x == b.x)
```

```
\square a{n, x} == b{n, x}
```

□ a == b

2. Sous unix (ou linux), la commande 1s permet de :

```
\Box afficher le contenu d'un fichier texte
```

 $\square\,$ voir des clips musicaux

□ compiler un programme

afficher	la	liste	$\mathrm{d}\mathrm{e}$	fichiers	contenus	$\operatorname{dans}$	uı
répertoi	re						

3. Pour compiler un programme prog.c, on utilise la ligne de commande :

```
\square gcc prog.c -o -Wall prog.exe
```

```
☐ gcc -Wall prog.exe -o prog.c
```

```
☐ gcc -Wall prog.c -o prog.exe
```

 $\square$  gcc prog.exe -Wall -o prog.c

4. Le langage C est un langage

```
\square composé
```

 $\Box$ compilé

 $\Box$ interprété

 $\square$  lu, écrit, parlé

5. Pour l'extrait de programme suivant :

```
int produit = 1;
int serie[4] = {2, 2, 2, 2};
for (i = 0; i < 4; i = i + 1)
{
   produit = produit * serie[i];
}
printf("produit = %d", produit);</pre>
```

La valeur affichée est :

	4
--	---

16

 $\Box$  0  $\Box$  8

6. Le code suivant :

```
int age = 15;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");
}</pre>
```

affichera:

Mineur

☐ Mineur Majeur

 $\square$  Majeur

 $\square$  rien

7. Une segmentation fault est une erreur qui survient lorsque:

$\hfill \square$ le programme source a été enregistré sur le disque
dur en plusieurs morceaux et l'un d'entre eux ne
peut pas être chargé par le compilateur

la	${\rm division}$	$\mathrm{d} u$	programme	en	zones	homogène
écl	houe					

le programme tente d'accèder à une partie	de	1
mémoire qui ne lui est pas réservée		

□ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal

8. Vous utilisez une boucle while quand:

vous	avez	déjà	fait	un	for	dans	le	${\rm m\^{e}me}$	pro
gram	me pi	rincip	al						

 $\Box$  l'incrément de la variable de boucle n'est pas 1

```
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
```

□ vous n'avez pas déclaré de fonction

9. Un registre du processeur est :

```
\Box une case mémoire interne au processeur qui sera manipulée directement lors des calculs
```

 $\Box$  une gamme de fréquence de fonctionnement du processeur

□ un composant qui contient la liste des fichiers du système

 $\Box$  une unité de calcul spécialisée de l'ordinateur

10. Pour l'extrait de programme suivant :

```
int somme = 0;
int serie[4] = {2, 4, 10, 4};
for (i = 0; i < 4; i = i + 1)
{
   somme = somme + serie[i];
}
printf("somme = %d",somme);</pre>
```

La valeur de somme affichée est :

	20
П	16

 $\Box$  3

 $\Box$  6

11. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
☐ int factorielle(int 2);
```

□ n = factorielle();

 $\hfill\Box$  printf("%d", factorielle(n));

 $\square$  n = factorielle(p, q);

12. Si racine est une fonction prenant en entrée un réel	$\square$ créer un fichier texte	18. Le type des réels en C est :
et renvoyant la racine carrée de cet réel, et que x est	□ créer un répertoire	$\square$ real
une variable réelle définie et initialisée, il est incorrect d'écrire :	□ ouvrir un fichier texte	☐ double
$\square$ x = racine(x * x) - racine(x);	15. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max	, $\Box$ int
$\square$ x - 1 = racine(x);	, que doit-on chercher dans le programme?	☐ char
$\square$ x = racine(racine(x)*racine(x));	$\square$ une directive préprocesseur <b>#include</b> manquante	19. Quel est le problème d'un programme comportant les
$\square$ x = racine(2/3);	un désaccord entre la déclaration et la définition	lignes suivantes?
13. Soit la fonction g définie par :	d'une fonction  ☐ une fonction appelée avant sa déclaration	while (1)
int g(int a)	☐ une fonction déclarée mais non définie	<pre>{    printf("coucou\n");</pre>
<pre>{   printf("a = \n", %d);   if (1 &gt; 0)</pre>	16. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :	}
{	☐ int pgcd(2);	☐ il comporte une boucle infinie
return 5;	$\square$ n = pgcd(int p, int q);	□ il n'affiche rien
}	$\square n = pgcd(n, 3);$	$\square$ il risque d'afficher bonjour à la place de coucou
return 7;	$\square$ int n = pgcd();	$\Box$ il ne compile pas
Alors l'expression g(0) prendra la valeur :  □ 0 □ 5	17. L'ordonnancement par tourniquet permet :  □ d'entretenir l'illusion que les processus tournent en parallèle	20. L'écriture $\underline{101}$ en binaire correspond au nombre naturel : \$\square\$ 4
	☐ d'afficher des ronds colorés à l'écran	□ 5
□ 7	☐ de ne pas perdre de temps avec la commutation de contexte	
14. Sur unix (ou linux), la commande mkdir permet de :		$\Box$ 101
$\Box$ changer de répertoire courant	$\square$ de doubler la mémoire disponible	$\square$ 3

лсепсе	

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Avant de faire appel à une fonction il est nécessaire de: □ l'avoir déclarée □ l'avoir déclarée et définie □ avoir défini une constante symbolique de la taille de cette fonction □ l'avoir définie 2. Vous utilisez une boucle while quand: ☐ l'incrément de la variable de boucle n'est pas 1  $\square$  vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ vous n'avez pas déclaré de fonction □ vous avez déjà fait un for dans le même programme principal 3. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd : □ certaines données de la mémoire de travail □ en temps d'accès  $\square$  les fichiers du disque  $\square$  des processus 4. Pour compiler un programme prog.c, on utilise la ligne de commande : ☐ gcc -Wall prog.exe -o prog.c ☐ gcc prog.c -o -Wall prog.exe ☐ gcc -Wall prog.c -o prog.exe ☐ gcc prog.exe -Wall -o prog.c 5. Soit la fonction g définie par : int g(int a)  $printf("a = \n", %d);$ if (1 > 0)return 5; return 7;

```
Alors l'expression g(0) prendra la valeur :
      \square 7
     \Box 0
     \Box 5
6. Le bus système sert à :
     ☐ Arriver à l'heure en cours
     □ transporter les processus du tourniquet au pro-
     ☐ Écrire des données sur le dique dur
     ☐ Transférer des données et intructions entre pro-
         cesseur et mémoire
 7. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     □ struct date_s afficher_date(struct date_s | d);
     □ void afficher_date(struct date_s d);
     ☐ int afficher_date(date_s d);
      □ void afficher_date(date_s d);
 8. Sous unix (ou linux), la commande cd permet de :
     □ changer de répertoire courant
     □ ouvir un bureau partagé (common desktop)
     □ détruire un fichier
     \square jouer de la musique
     □ récupérer un programme arrêté avec la commande
         ab
 9. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x - 1 = racine(x):
     \square x = racine(racine(x)*racine(x));
     \square x = racine(x * x) - racine(x);
     \square x = racine(2/3);
10. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 4
     \Box 5
     \square 101
      \square 3
```

```
11. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 8
      \Box 4
      \Box 0
      \Box 16
12. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      □ une fonction déclarée mais non définie
      ☐ une directive préprocesseur #include manquante
      \square une fonction appelée avant sa déclaration
      \Box un désaccord entre la déclaration et la définition
         d'une fonction
13. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il risque d'afficher bonjour à la place de coucou
      □ il n'affiche rien
      \square il ne compile pas
      \square il comporte une boucle infinie
```

14.	dans le répertoire courant on peut utiliser la com-
	mande:
	□ mkdir TP4
	□ new TP4
	□ kwrite TP4
	$\square$ yppasswd
15.	Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
	☐ void saisie_utilisateur(char c);
	☐ saisie_utilisateur(scanf(%d));
	☐ int saisie_utilisateur();
	☐ void saisie_utilisateur(int n);
16.	Pour l'extrait de programme suivant :
	<pre>int somme = 0; int serie[4] = {2, 4, 10, 4}; for (i = 0; i &lt; 4; i = i + 1) {     somme = somme + serie[i]; } printf("somme = %d",somme);</pre>
	La valeur de somme affichée est :

```
\square 20
      \square 3
      \Box 16
17. Après exécution du programme :
       lecture 8 r0
       valeur 3 r1
       mult r1 r0
       valeur 1 r2
       add r2 r0
       ecriture r0 8
       stop
       5
      □ la case mémoire 8 contiendra 16
     \square la case mémoire 8 contiendra 0
     \square le bus explose
     \square le terminal affiche 8
18. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     □ n = factorielle();
     \square n = factorielle(p, q);
     ☐ printf("%d", factorielle(n));
     ☐ int factorielle(int 2);
```

```
19. Le code suivant :
    int age = 18;
    if (age < 18)
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
   affichera:
     □ Majeur
     ☐ Mineur
        Majeur
     \square rien
     ☐ Mineur
20. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=1;i<=5;i=i+1)
     ☐ for(i=0;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
```

 $\square$  for(i=1;i<5;i=i+1)

т.	-1
Licence	

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1.	Un enregistrement permet de grouper plusieurs valeurs dans : $% \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} $
	□ ses blocs
	□ ses cases
	$\square$ ses champs
	$\square$ ses chants
2.	Le bus système sert à :
	☐ Arriver à l'heure en cours
	$\Box$ Écrire des données sur le dique dur
	$\Box$ Transférer des données et intructions entre processeur et mémoire
	$\Box$ transporter les processus du tourniquet au processeur
3.	Avant de faire appel à une fonction il est nécessaire de :
	$\hfill \square$ avoir défini une constante symbolique de la taille de cette fonction
	□ l'avoir définie
	□ l'avoir déclarée
	$\Box$ l'avoir déclarée et définie
4.	Si $n$ est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :
	☐ printf("Valeur de n ? %g\n", n);
	☐ printf("Valeur de n ? %d\n", n);
	□ scanf("%d", &n);
	$\Box$ un débogueur
5.	Dans la commande gcc, l'option -Wall signifie :
	$\hfill \square$ qu'on veut changer alétoirement de fond d'écran
	□ qu'il faut lancer un déboggueur
	$\hfill\Box$ qu'il faut indenter le fichier source
	$\hfill\Box$ que l'on veut voir tous les avertissements

```
6. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x - 1 = racine(x);
     \Box x = racine(x * x) - racine(x);
     \square x = racine(2/3);
     \square x = racine(racine(x)*racine(x));
7. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \Box 1
     \square 1.5
     \Box 0
 8. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \Box int mccarthy(int 2);
     \square x = mccarthy(n);
     \square n = mccarthy();
     \square n = mccarthy(p, q);
9. Pour compiler un programme prog.c, on utilise la
   ligne de commande:
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
10. Quelle étape de la compilation vient d'échouer lors-
    qu'on a un message comme celui-ci :
   Undefined symbols :"_prinft" ou
   référence indéfinie vers « prinft »
     ☐ l'analyse harmonique
     \square l'analyse des entrées clavier
     □ l'analyse sémantique
     □ l'édition de liens
```

INE .
11. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonction est l'ordre :
$\hfill\Box$ un ordre quel conque
$\Box$ dans lequel ces fonctions sont appelées dans le main
$\square$ alphabétique
$\Box$ dans lequel vous avez déclaré ces fonction
12. L'écriture $\underline{101}$ en binaire correspond au nombre naturel :
$\Box$ 4
$\Box$ 5
$\Box$ 3
□ 101
13. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :
$\square$ n = factorielle(p, q);
☐ int factorielle(int 2);
<pre>printf("%d", factorielle(n));</pre>
☐ n = factorielle();
14. Après exécution jusqu'à la ligne 15 du programme C
<pre>10  int main() { 11    int x = 5; 12    int y = 3; 13</pre>
14 x = y; 15
16 17 }
$\square$ la variable x vaut 3
$\Box$ le programme affiche "Faux"
$\Box$ la variable x vaut 5 et la variable y vaut 3
$\Box$ la variable y vaut 5

15. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction	{     for(j=i;j<6;j=j+1)	$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres
<ul><li>□ int %d;</li><li>□ loop i;</li></ul>	{ printf("*");	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition
☐ int k;	<pre>} printf(" ");</pre>	□ répéter un bloc tant qu'une condition est vérifée
☐ int loop n;	}	19. Un programme en langage C doit comporter une et une seule définition de la fonction :
16. L'écriture <u>111</u> en binaire correspond au nombre naturel :	qu'est ce qui sera affiché?	$\Box$ init
□ 111		$\square$ main
□ 8	_ ** ** ** ** **	$\square$ include
□ 7		$\Box$ begin
$\Box$ 3	**** *** ***	20. Quel est l'opérateur de différence en C :
17. Pour l'extrait de programme suivant :	18. Une de ces manière de composer les blocs de pro-	□ !=
	grammes ne fait pas partie des opérations de la pro-	□ !
<pre>int i; int j;</pre>	grammation structurée:	□≠
for(i=4;i>0;i=i-1)	$\square$ retourner un bloc	$\Box \Leftrightarrow$

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
Nº INE ·		
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
  - □ new TP4
  - □ yppasswd
  - ☐ mkdir TP4
  - ☐ kwrite TP4
- 2. Quels calculs peut-on programmer en programmation structurée?
  - □ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
  - □ certains programmes sont de vrais plats de spaghetti
  - □ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée
  - □ en programmation structurée on peut programmer tous les calculs programmables en langage machine
- 3. Les lignes

```
int i;
int x=0;
for(i=0,i<5,i=i+1)
{
    x=x+1;
}</pre>
```

- □ comportent une erreur qui sera détectée au cours de l'édition de lien
- □ comportent une erreur qui sera détectée au cours de l'analyse syntaxique
- $\square\,$  comportent une erreur qui ne sera pas détectée
- $\Box\,$ ne comportent aucune erreur

```
4. Si a et b sont deux variables de type:
   struct toto s
   {
     int n;
     double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
    \square a = b
    \square a{n, x} == b{n, x}
    \Box (a.n == b.n) && (a.x == b.x)
     □ a == b
5. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
    \Box int mccarthy(int 2);
     \square n = mccarthy();
    \square x = mccarthy(n);
     \square n = mccarthy(p, q);
6. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
```

```
7. Sur unix (ou linux), la commande mkdir permet de :
      □ créer un répertoire
      □ changer de répertoire courant
      □ ouvrir un fichier texte
      □ créer un fichier texte
 8. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
    on chercher dans le programme?
      □ un point-virgule en trop
      ☐ une accolade manquante
      \square une accolade en trop
      ☐ un point-virgule manquant
 9. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      □ analyse sémantique
      \square analyse harmonique
      \square analyse lexicale
      \square analyse syntaxique
10. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square double exposant(double x, int n);
      \square void exposant(double x^n);
      \square int exposant(double n, int x);
      \square exposant(double x, int n, int r);
11. Si le code :
    struct toto_s
      int n;
      double x;
    précède la fonction main(), alors on peut écrire en
    début de main():
     □ struct toto_s toto;
      \square toto_s struct z = {3, 0.5};
     \square int toto.n = 3;
      \square toto_s n, x;
      \Box int struct toto_s = {3, -1e10};
```

12. Un enregistrement permet de grouper plusieurs valeurs	16. Le code suivant :	{
dans:	int i;	for(j=i;j<6;j=j+1)
$\square$ ses champs	for $(i = 0; i < 5; i = i + 1)$	<pre>printf("*");</pre>
$\square$ ses chants	{	}
☐ ses cases	<pre>printf("%d ", i); }</pre>	<pre>printf(" ");</pre>
☐ ses blocs	<pre>printf("\n");</pre>	}
13. Vous avez déclaré préalablement un ensemble de fonc-	affichera:	
tions utilisées par votre programme principal. L'ordre	$\Box \ 0\ 1\ 2\ 3$	qu'est ce qui sera affiché?
dans lequel vous devez maintenant définir ces fonctions	$\Box \ 0\ 1\ 2\ 3\ 4$	☐ ** ** ** ** ** **
est l'ordre :	$\Box \ 4\ 3\ 2\ 1\ 0$	**** *** *** ***
□ alphabétique	$\square \ 4\ 3\ 2\ 1$	☐ ** *** **** ****
$\square$ dans lequel vous avez déclaré ces fonction	17. Une variable booléenne est un variable :	
$\square$ un ordre quelconque	$\Box$ à la quelle une valeur vient d'être affectée	
$\Box$ dans lequel ces fonctions sont appelées dans le .	☐ réelle positive	20. Après exécution jusqu'à la
main	$\Box$ jamais nulle	10
14. Si factorielle est une fonction prenant en entrée un	$\Box$ qui est vraie ou fausse	11 int main() {
entier et renvoyant un entier, il est correct d'écrire :	$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)	$12 \qquad \text{int } x = 5;$
□ n = factorielle(p, q);	18. Si pgcd est une fonction prenant en entrée deux entiers	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
☐ printf("%d", factorielle(n));	et renvoyant un entier, il est correct d'écrire :	15
☐ int factorielle(int 2);	$\square$ n = pgcd(int p, int q);	16
☐ n = factorielle();	$\square$ int n = pgcd();	17 }
15. Le langage C est un langage	☐ int pgcd(2);	$\Box$ le programme affiche :
$\square$ composé	$\square n = pgcd(n, 3);$	□ la variable x vaut 16
$\square$ compilé	19. Pour l'extrait de programme suivant :	
□ lu, écrit, parlé	int i;	☐ le programme affiche :
□ interprété	int j;	$\Box$ la variable x vaut $-\frac{1}{2}$
	for(i=4;i>0;i=i-1)	

```
for(j=i;j<6;j=j+1)
  printf("*");
 printf(" ");
l'est ce qui sera affiché?
  ** ** ** ** ** **
 ***** **** ***
    ** *** **** *****
 **** **** ****
près exécution jusqu'à la ligne 15 du programme C:
 int main() {
     int x = 5;
     x = 3 * x + 1;
 }
\Box le programme affiche x
\Boxla variable x vaut 16
\Boxle programme affiche ****
```

## Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Dans la commande gcc, l'option -Wall signifie : □ qu'il faut lancer un déboggueur □ qu'on veut changer alétoirement de fond d'écran ☐ que l'on veut voir tous les avertissements □ qu'il faut indenter le fichier source 2. Vous utilisez une boucle while quand: □ vous n'avez pas déclaré de fonction □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance ☐ l'incrément de la variable de boucle n'est pas 1 □ vous avez déjà fait un for dans le même programme principal 3. Pour compiler un programme prog.c, on utilise la ligne de commande : ☐ gcc -Wall prog.c -o prog.exe ☐ gcc prog.exe -Wall -o prog.c ☐ gcc prog.c -o -Wall prog.exe ☐ gcc -Wall prog.exe -o prog.c 4. Le code suivant : int i; for (i = 4; i > 0; i = i - 1)printf("%d ", i); printf("\n"); affichera:  $\square$  4 3 2 1  $\square$  0 1 2 3  $\Box \ 4\ 3\ 2\ 1\ 0$ 

 $\Box 01234$ 

```
5. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre:
     □ alphabétique
     \square dans lequel ces fonctions sont appelées dans le
        main
     \square un ordre quelconque
     □ dans lequel vous avez déclaré ces fonction
 6. Sur unix (ou linux), la commande mkdir permet de :
      □ ouvrir un fichier texte
     □ créer un fichier texte
     □ changer de répertoire courant
     □ créer un répertoire
 7. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     □ void saisie_utilisateur(int n);
     ☐ int saisie_utilisateur();
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
 8. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
     □ #define N 3
     ☐ #define taille = 3
     \square #define N = 3
     \square #define taille = N
9. Sous unix (ou linux), la commande 1s permet de :
     □ compiler un programme
     □ afficher la liste de fichiers contenus dans un
        répertoire
     □ voir des clips musicaux
      □ afficher le contenu d'un fichier texte
10. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %d\n", n);
     ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
```

□ un débogueur

```
11. Le bus système sert à :
     ☐ Écrire des données sur le dique dur
     ☐ Arriver à l'heure en cours
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     □ transporter les processus du tourniquet au pro-
12. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     ☐ int afficher_date(date_s d);
     □ void afficher_date(struct date_s d);
     ☐ struct date_s afficher_date(struct date_s d);
     □ void afficher_date(date_s d);
13. Quel est le problème d'un programme comportant les
   lignes suivantes?
    while (1)
      printf("coucou\n");
     ☐ il risque d'afficher bonjour à la place de coucou
     □ il comporte une boucle infinie
     □ il n'affiche rien
     \square il ne compile pas
14. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
              printf("%d ", j);
         }
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 3 0 1 2
     \Box 0 0 1 1 2 2 3
```

15.	Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :
	☐ int afficher_menu(int char);
	☐ char afficher_menu(printf("menu"));
	☐ void afficher_menu();
	☐ double afficher_menu();
	☐ int afficher_menu();
16.	Un enregistrement permet de grouper plusieurs valeurs dans : $% \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2$
	$\square$ ses champs
	□ ses cases
	$\square$ ses chants
	$\square$ ses blocs
17.	Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?
	$\Box$ une fonction appelée avant sa déclaration
	$\Box$ une directive préprocesseur #include manquante
	$\square$ une fonction déclarée mais non définie
	$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction

```
18. Soit le programme principal suivant :
   int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
19. Un registre du processeur est :
     \square un composant qui contient la liste des fichiers du
        système
```

```
□ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     \Box une gamme de fréquence de fonctionnement du
        processeur
     \Box\, une unité de calcul spécialisée de l'ordinateur
20. Soit un programme contenant les lignes suivantes :
     int i = 0;
    int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
     }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box j = %d
     \Box j = 4
     \Box j = 0
     □ j = 5
```

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution jusqu'à la ligne 15 du programme C:
 10
      . . .
 11
      int main() {
 12
           int x = 5;
 13
 14
           x = 3 * x + 1:
 15
 16
           . . .
      }
 17
     \square le programme affiche ****
     \square le programme affiche x
     \square la variable x vaut -\frac{1}{2}
     □ la variable x vaut 16
2. Le langage C est un langage
     □ lu, écrit, parlé
     □ composé
     □ interprété
     □ compilé
3. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera:
     □ A
     ПВ
     \sqcap C
     □ b
4. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
```

 $\square$  ses champs

 $\square$  ses chants

 $\square$  ses cases

```
5. Le code suivant :
    int age = 20;
    if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     □ Majeur
     □ Mineur
        Majeur
     □ rien
     □ Mineur
6. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \Box 0
     \square 1.5
     \Box 1
     \square 0.5
7. Avant de faire appel à une fonction il est nécessaire
     ☐ l'avoir déclarée et définie
     □ avoir défini une constante symbolique de la taille
        de cette fonction
     □ l'avoir déclarée
     □ l'avoir définie
8. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     ☐ mkdir TP4
     □ new TP4
     ☐ yppasswd
     ☐ kwrite TP4
```

```
9. Si le code:
   struct toto_s
      int n;
      double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main():
     \Box int struct toto_s = {3, -1e10};
     \Box toto_s struct z = {3, 0.5};
     \square toto_s n, x;
     □ struct toto s toto:
     \square int toto.n = 3;
10. Quels calculs peut-on programmer en programmation
   structurée?
     \square il y a des calculs programmables en langage ma-
        chine et qui ne sont pas programmables en pro-
        grammation structurée
     □ certains programmes sont de vrais plats de spa-
        ghetti
     \square il y a des calculs programmables en programma-
        tion structurée qui ne sont pas programmables en
        langage machine
     □ en programmation structurée on peut program-
        mer tous les calculs programmables en langage
        machine
11. Pour l'extrait de programme suivant :
     int i = 0:
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
         }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 3 0 1 2
     \Box 0 1 2 0 1 2 3
```

```
12. L'ordonnancement par tourniquet permet :
                                                                                                                           {
                                                                 □ cccccc
     □ d'entretenir l'illusion que les processus tournent
                                                                                                                                printf("Majeur\n");
                                                                 □ A
        en parallèle
                                                                                                                           }
                                                                 \Box i
     □ de doubler la mémoire disponible
                                                           15. Quel est le problème d'un programme comportant les
     \square d'afficher des ronds colorés à l'écran
                                                                                                                          affichera:
                                                               lignes suivantes?
     \square de ne pas perdre de temps avec la commutation
                                                                                                                            □ Mineur
        de contexte
                                                               while (1)
                                                                                                                               Majeur
                                                               {
                                                                                                                            ☐ Mineur
13. Soit le programme principal suivant :
                                                                 printf("coucou\n");
                                                                                                                            \square rien
   int main()
                                                                                                                            □ Majeur
                                                                 \square il ne compile pas
     int a = 3;
                                                                 \square il comporte une boucle infinie
                                                                                                                      19. Pour l'extrait de programme suivant :
     int b = 5;
                                                                 □ il n'affiche rien
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                           int i = 0;
    return EXIT_SUCCESS;
                                                                 □ il risque d'afficher bonjour à la place de coucou
                                                                                                                           int j = 0;
                                                                                                                           for (i = 0; i < 3; i = i + 1)
                                                           16. On considère deux variables booléennes A et B initia-
   appelant la fonction f ainsi définie :
                                                               lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                                                                                for (j = 0; j < 2; j = j + 1)
                                                               pressions booléennes suivantes, laquelle a pour valeur
   int f(int a, int b)
                                                               TRUE?
                                                                                                                                    printf("%d ", i);
      a = a + b;
                                                                 \square (A == TRUE) && (B == TRUE)
                                                                                                                                }
      return a;
                                                                                                                           }
                                                                 □ A && B
                                                                                                                           printf("\n");
                                                                 \square (!A || B)
   L'affichage dans le main est le suivant :
                                                                                                                          qu'est ce qui sera affiché?
                                                                 \square !(!A || B) == (A && !B)
     \Box f(3,5)=8, a=3, b=5
                                                                                                                            \Box 1 2 3 1 2
                                                           17. Pour compiler un programme prog.c, on utilise la
     \Box f(a,b)=8, a=3, b=5
                                                               ligne de commande:
                                                                                                                            \Box 0 1 2 0 1 2
     \Box f(a,b)=8, a=8, b=5
                                                                 ☐ gcc -Wall prog.exe -o prog.c
                                                                                                                            \Box 0 1 0 1 0 1
     \Box f(a,b)=13, a=8, b=5
                                                                 ☐ gcc -Wall prog.c -o prog.exe
                                                                                                                            \Box 0 0 1 1 2 2
                                                                 ☐ gcc prog.exe -Wall -o prog.c
14. Au début de la fonction main() on place le code :
                                                                                                                      20. Vous utilisez une boucle while quand:
                                                                 ☐ gcc prog.c -o -Wall prog.exe
     char i:
                                                                                                                            □ l'incrément de la variable de boucle n'est pas 1
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                           18. Le code suivant :
                                                                                                                            □ vous n'avez pas déclaré de fonction
                                                                int age = 18;
                                                                                                                            □ vous avez déjà fait un for dans le même pro-
       printf("%c", i);
                                                                if (age < 18)
                                                                                                                               gramme principal
                                                                {
     printf("\n");
                                                                                                                            \square vous ne connaissez pas le nombre d'itérations de
                                                                    printf("Mineur\n");
                                                                                                                               la boucle à l'avance
   Alors l'affichage sera:
                                                                }
```

else

☐ ABCDEF

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int age = 18;
    if (age < 18)
        printf("Mineur\n");
    else
        printf("Majeur\n");
    }
  affichera:
     □ rien
     ☐ Mineur
        Majeur
     □ Majeur
     ☐ Mineur
2. Laquelle de ces écritures correspond à la déclaration
  d'une variable de type caractère en langage C?
     ☐ char "c";
    ☐ char 'c';
     □ char c;
    ☐ int char;
3. Un enregistrement permet de grouper plusieurs valeurs
  dans:
     \square ses chants
     \square ses blocs
     \square ses champs
     \square ses cases
4. Pour déclarer un tableau d'entiers de taille 5, on peut
  utiliser l'instruction
     □ char tableau[5];
    \square int toto[5]:
    \square int[] new tableau(5);
     \square int tab[] = 5;
     ☐ int toto[taille=5];
```

```
5. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
  est l'ordre:
    \square un ordre quelconque
    □ alphabétique
    □ dans lequel ces fonctions sont appelées dans le
       main
     □ dans lequel vous avez déclaré ces fonction
6. Vous utilisez une boucle while quand :
     ☐ l'incrément de la variable de boucle n'est pas 1
    □ vous n'avez pas déclaré de fonction
    □ vous avez déjà fait un for dans le même pro-
       gramme principal
    □ vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
7. Le langage C est un langage
    □ lu, écrit, parlé
    □ compilé
    □ composé
    □ interprété
8. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
    int a = 0;
    int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
    ₹
      scanf("%d", &n);
   Quelle est la condition cond:
    □ (n<=a) && (n<=b)
    \square (a<n) || (n>b)
     □ a<=n<=b
     \square (a<=n) && (n<=b)
```

```
9. La virtualisation de la mémoire permet notamment de
   stocker des portions inactives de la mémoire de travail
   sur le disque dur. Mais on perd :
     □ en temps d'accès
     □ les fichiers du disque
     \square des processus
     □ certaines données de la mémoire de travail
10. Soit le programme principal suivant :
   int main()
   {
     int a = 3:
     int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   }
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b:
      return a;
   }
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
11. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %g\n", n);
     □ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
     □ un débogueur
12. Le code suivant :
     int i;
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
```

printf("\n");

```
affichera:
     \Box 43210
     \square 4 3 2 1
     \Box 01234
     \Box 1 2 3 4
13. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i:
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 15
      \Box 0
     \Box 10
     \Box 6
14. Sous unix (ou linux), la commande cd permet de :
      □ récupérer un programme arrêté avec la commande
        ab
     □ détruire un fichier
     □ ouvir un bureau partagé (common desktop)
      \square jouer de la musique
                                                                 }
     \square changer de répertoire courant
```

```
15. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square int exposant(double n, int x);
     \square double exposant(double x, int n);
     \square exposant(double x, int n, int r);
     \square void exposant(double x^n);
16. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
    Alors l'affichage sera:
     □ A
     □b
     \square B
     \Box C
17. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     \square n = factorielle();
     ☐ int factorielle(int 2);
     ☐ printf("%d", factorielle(n));
     \square n = factorielle(p, q);
18. Le code suivant :
     int i:
     for (i = 0; i < 5; i = i + 1)
         printf("%d ", i);
    printf("\n");
```

```
affichera:
      \Box 0123
      \Box 01234
      \Box \ 4\ 3\ 2\ 1\ 0
      \square 4 3 2 1
19. Après exécution jusqu'à la ligne 15 du programme C:
        int main() {
 11
             int x = 5:
  12
             int y;
  13
  14
             y = x;
  15
  16
 17
      }
      \square la variable y vaut 5
      \square la variable x vaut 0
      □ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 0
20. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x = racine(x * x) - racine(x);
      \square x - 1 = racine(x);
      \square x = racine(racine(x)*racine(x));
```

 $\square$  x = racine(2/3);

т		-1
	100000	- 1
	acence	

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

r r	réponse fausse. Durée : 20 minutes.
1.	Au début de la fonction main() on place le code :
	<pre>char b = 'A'; b = b + 2; printf("%c\n", b);</pre>
	Alors l'affichage sera :
	□b
	□ A
	□В
	□ C
2.	La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
	$\square$ des processus
	$\Box$ les fichiers du disque
	$\Box$ certaines données de la mémoire de travail
	$\Box$ en temps d'accès
3.	Vous utilisez une boucle while quand:
3.	Vous utilisez une boucle while quand : $ \  \   \square \   l'incrément de la variable de boucle n'est pas 1$
3.	-
3.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de</li> </ul>
3.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même pro-</li> </ul>
	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> </ul>
	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> </ul>
	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> </ul>
	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> </ul>
	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> <li>□ créer un fichier texte</li> </ul>
4.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> <li>□ créer un fichier texte</li> <li>□ ouvrir un fichier texte</li> <li>□ créer un répertoire</li> <li>Un enregistrement permet de grouper plusieurs valeurs</li> </ul>
4.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> <li>□ créer un fichier texte</li> <li>□ ouvrir un fichier texte</li> <li>□ créer un répertoire</li> <li>Un enregistrement permet de grouper plusieurs valeurs dans :</li> </ul>
4.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> <li>□ créer un fichier texte</li> <li>□ ouvrir un fichier texte</li> <li>□ créer un répertoire</li> <li>Un enregistrement permet de grouper plusieurs valeurs</li> </ul>
4.	<ul> <li>□ l'incrément de la variable de boucle n'est pas 1</li> <li>□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance</li> <li>□ vous avez déjà fait un for dans le même programme principal</li> <li>□ vous n'avez pas déclaré de fonction</li> <li>Sur unix (ou linux), la commande mkdir permet de :</li> <li>□ changer de répertoire courant</li> <li>□ créer un fichier texte</li> <li>□ ouvrir un fichier texte</li> <li>□ créer un répertoire</li> <li>Un enregistrement permet de grouper plusieurs valeurs dans :</li> </ul>

 $\square$  ses champs

```
6. Le code suivant :
   int age = 18;
   if (age < 18)
       printf("Mineur\n");
   }
   else
   {
       printf("Majeur\n");
  affichera:
    □ rien
    □ Majeur
    ☐ Mineur
    □ Mineur
       Majeur
7. Pour déclarer une procédure afficher_date qui prend
  en argument un struct date_s et affiche le contenu
  du struct, on écrit :
    ☐ int afficher_date(date_s d);
    □ struct date_s afficher_date(struct date_s |d);
    □ void afficher_date(date_s d);
    □ void afficher_date(struct date_s d);
8. Soit un programme contenant les lignes suivantes :
   int i = 0;
   int j = 0;
   for (i = 0; i < 2; i = i + 1)
       for (j = 0; j < 3; j = j + 1)
            printf("%d ", i);
       }
   }
  qu'est ce qui sera affiché?
    \Box 0 1 2 0 1 2
    \Box 0 0 0 1 1 1
    \Box 1 2 1 2 3
```

 $\Box$  0 1 0 1 0 1 0 1

```
9. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square int n = pgcd();
      \square int pgcd(2);
      \square n = pgcd(int p, int q);
      \square n = pgcd(n, 3);
10. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
      ☐ int char;
      □ char 'c';
      \Box char c;
      ☐ char "c";
11. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire:
      \square x = racine(racine(x)*racine(x));
      \square x = racine(x * x) - racine(x);
      \square x = racine(2/3);
      \square x - 1 = racine(x);
12. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
      ☐ int factorielle(int 2);
      \square n = factorielle(p, q);
      ☐ printf("%d", factorielle(n));
      \square n = factorielle();
13. Le bus système sert à :
      ☐ Arriver à l'heure en cours
      □ transporter les processus du tourniquet au pro-
         cesseur
      ☐ Transférer des données et intructions entre pro-
         cesseur et mémoire
      ☐ Écrire des données sur le dique dur
```

```
14. Pour l'extrait de programme suivant :
                                                            17. Le code suivant :
                                                                                                                              else
                                                                                                                              {
      int produit = 1;
                                                                 int i;
                                                                                                                                   printf("Majeur\n");
      int serie[4] = \{2, 2, 2, 2\};
                                                                 for (i = 1; i < 5; i = i + 1)
      for (i = 0; i < 4; i = i + 1)
      {
                                                                      printf("%d ", i);
        produit = produit * serie[i];
                                                                                                                             affichera:
                                                                 printf("\n");
                                                                                                                               □ Mineur
      printf("produit = %d", produit);
                                                                affichera:
                                                                                                                               ☐ Mineur
    La valeur affichée est :
                                                                  \Box 1 2 3 4
                                                                                                                                  Majeur
      \Box 16
                                                                  \Box 01234
                                                                                                                               \square rien
      \Box 0
                                                                  \square 4 3 2 1
                                                                                                                               □ Majeur
      \Box 4
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
      \square 8
                                                                                                                         20. Soit la fonction f définie par :
                                                            18. Si cette erreur apparaît à la compilation :
15. L'ordonnancement par tourniquet permet :
                                                                erreur: conflicting types for 'max', que doit-
                                                                                                                             int f(int a)
      \square de doubler la mémoire disponible
                                                                on chercher dans le programme?
                                                                                                                             {
      \square d'afficher des ronds colorés à l'écran
                                                                                                                               printf("a = \n", %d);
                                                                  \square une fonction déclarée mais non définie
                                                                                                                               if (a > 0)
      \square de ne pas perdre de temps avec la commutation
                                                                  □ une directive préprocesseur #include manquante
                                                                                                                               {
         de contexte
                                                                  □ un désaccord entre la déclaration et la définition
                                                                                                                                 return 3;
      \Box d'entretenir l'illusion que les processus tournent
                                                                     d'une fonction
                                                                                                                               }
         en parallèle
                                                                  \square une fonction appelée avant sa déclaration
                                                                                                                               return 4;
16. Pour déclarer une fonction factorielle qui prend en
                                                            19. Le code suivant :
    argument un entier et renvoie sa factorielle on écrit :
                                                                                                                             Alors l'expression f(0) prendra la valeur :
                                                                 int age = 20;
      ☐ int factorielle(double n);
                                                                 if (age < 18)
                                                                                                                               \Box 0
      \Box int factorielle(int x);
                                                                 {
                                                                                                                               \square 3
      ☐ int factorielle();
                                                                      printf("Mineur\n");
                                                                                                                               \Box 4
      □ struct int factorielle(int n);
                                                                 }
```

т •	-1
1.00000	- 1
Licence	

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
    x = racine(x * x) - racine(x);
    x = racine(2/3);
    x = racine(racine(x)*racine(x));
    x - 1 = racine(x);
```

2. Le code suivant :

```
int somme = 0;
int i;
for (i = 1; i < 4; i = i + 1)
{
   somme = somme + i;
}
printf("%d", somme);</pre>
```

affichera:

 $\begin{array}{c} \square & 42 \\ \square & 1 \end{array}$ 

 $\Box$  6

 $\Box$  0

3. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction

```
□ int tab[] = 5;
□ int[] new tableau(5);
□ char tableau[5];
□ int toto[5];
□ int toto[taille=5];
```

4. Au début de la fonction main() on place le code :

□ b

□ B□ C

```
5. Pour l'extrait de programme suivant :
     int produit = 1;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 16
     \square 8
     \Box 4
     \Box 0
6. Le code suivant :
    int i;
    for (i = 8; i > 0; i = i - 2)
        printf("%d ", i);
    printf("\n");
   affichera:
    \square 8 2
    \Box 8 6 4 2 0
    \Box 8 6 4 2
    \Box 02468
7. Dans la commande gcc, l'option -Wall signifie :
     ☐ qu'il faut indenter le fichier source
     □ qu'on veut changer alétoirement de fond d'écran
     ☐ que l'on veut voir tous les avertissements
     □ qu'il faut lancer un déboggueur
8. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.c -o -Wall prog.exe
```

☐ gcc prog.exe -Wall -o prog.c

☐ gcc -Wall prog.c -o prog.exe

☐ gcc -Wall prog.exe -o prog.c

9. Sur unix (ou linux), la commande mkdir permet de :
□ changer de répertoire courant
□ créer un fichier texte
□ créer un répertoire
□ ouvrir un fichier texte
10. Une variable booléenne est un variable :
□ à laquelle une valeur vient d'être affectée
□ réelle positive
□ NaN (not a number, qui n'est pas un nombre)
□ qui est vraie ou fausse
□ jamais nulle
11. Une segmentation fault est une erreur qui survien lorsque:
le programme tente d'afficher des caractères su une ligne qui va au delà de la largeur de la fenêtr du terminal
☐ la division du programme en zones homogène échoue
☐ le programme source a été enregistré sur le disqu dur en plusieurs morceaux et l'un d'entre eux n peut pas être chargé par le compilateur
☐ le programme tente d'accèder à une partie de l mémoire qui ne lui est pas réservée
12. Vous utilisez une boucle while quand:
□ vous ne connaissez pas le nombre d'itérations d la boucle à l'avance
$\hfill \square$ vous n'avez pas déclaré de fonction
$\hfill \square$ l'incrément de la variable de boucle n'est pas $1$
□ vous avez déjà fait un for dans le même programme principal
13. L'écriture <u>111</u> en binaire correspond au nombre naturel :
□ 7
□ 8
$\Box$ 3
□ 111

```
14. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 0 1 2 3 4 5 6
     \Box 02468
     \Box 0 1 2 3 4 5 6 7
     \Box 0246
15. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     ☐ int saisie_utilisateur();
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
     □ void saisie_utilisateur(int n);
16. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
     printf("produit = %d", produit);
```

```
La valeur affichée est :
      \Box 0
      \square 16
      \Box 4
      \square 8
17. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
      \Box int pgcd(int y, int x);
      \square void pgcd(int x, int y);
      \square int pgcd(int x, int x);
      \Box int pgcd(int x, y);
18. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      \Box dans lequel vous avez déclaré ces fonction
      □ un ordre quelconque
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
19. Soit le programme principal suivant :
    int main()
    {
```

```
int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
20. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square n = pgcd(n, 3);
     \square int n = pgcd();
```

 $\square$  n = pgcd(int p, int q);

 $\square$  int pgcd(2);

# Éléments d'informatique – contrôle continue

Prénom: Nom	:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Au début de la fonction main() on place le code : char i; for  $(i = 'A'; i \le 'F'; i = i + 1)$ printf("%c", i); printf("\n"); Alors l'affichage sera : ☐ ABCDEF □ A □ ccccc  $\Box$  i 2. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :  $\square$  analyse harmonique □ analyse sémantique  $\square$  analyse syntaxique  $\square$  analyse lexicale 3. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.  $\square$  #define N = 3 ☐ #define taille = N  $\square$  #define taille = 3 □ #define N 3 4. Quel est le problème d'un programme comportant les lignes suivantes? while (1) printf("coucou\n"); ☐ il risque d'afficher bonjour à la place de coucou  $\square$  il ne compile pas □ il n'affiche rien

 $\square$  il comporte une boucle infinie

```
5. Un fichier source est:
     □ un fichier texte qui sera traduit en instructions
        processeur
     □ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
     □ un document qui doit être protégé
     □ un document de référence du système
     □ un document illisible pour les humains
6. L'écriture <u>111</u> en binaire correspond au nombre natu-
   rel:
     \square 3
     \square 111
     \square 8
     \square 7
7. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre :
     □ un ordre quelconque
     □ dans lequel ces fonctions sont appelées dans le
        main
     □ alphabétique
     □ dans lequel vous avez déclaré ces fonction
8. Le code suivant :
    int age = 15;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     ☐ Mineur
        Majeur
     \square rien
     □ Majeur
```

☐ Mineur

```
9. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
     \Box 0
     \square 16
     \square 8
     \Box 4
10. Pour l'extrait de programme suivant :
     int i;
     int j;
    for(i=4;i>0;i=i-1)
       for(j=i;j<6;j=j+1)
         printf("*");
       printf(" ");
   qu'est ce qui sera affiché?
11. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
```

☐ gcc -Wall prog.c -o prog.exe

```
12. Soit la fonction g définie par :
                                                                 ☐ Mineur
                                                                 □ Majeur
   int g(int a)
                                                                \square rien
                                                                 ☐ Mineur
      printf("a = \n", %d);
                                                                                                                           int a = 0;
      if (1 > 0)
                                                                                                                           int b = 20;
                                                                   Majeur
                                                                                                                           int n:
                                                           15. Pour déclarer une fonction factorielle qui prend en
        return 5;
                                                                                                                           scanf("%d", &n);
                                                               argument un entier et renvoie sa factorielle on écrit :
                                                                                                                           while(cond)
                                                                 \square int factorielle(int x);
      return 7;
                                                                ☐ int factorielle(double n);
                                                                                                                             scanf("%d", &n);
                                                                □ struct int factorielle(int n);
   Alors l'expression g(0) prendra la valeur :
                                                                 ☐ int factorielle();
      \Box 5
                                                           16. Soit un programme contenant les lignes suivantes :
      \Box 0
                                                                int i = 0;
      \square 7
                                                                int i = 0;
                                                                for (i = 0; i < 3; i = i + 1)
13. Pour déclarer une fonction exposant qui prend en ar-
                                                                                                                            □ a<=n<=b
   gument un réel x et un entier positif n et renvoie la
                                                                                                                            □ (a<n) || (n>b)
                                                                    for (j = 0; j < 5; j = j + 1)
   valeur de x^n on écrit :
      \square exposant(double x, int n, int r);
     \square double exposant(double x, int n);
                                                                    }
     \square int exposant(double n, int x);
                                                                                                                         TRUE?
                                                               printf("j = %d\n", j);
     \square void exposant(double x^n);
14. Le code suivant :
                                                              qu'est ce qui sera affiché par ce printf?
                                                                                                                            \square (!A || B)
     int age = 18;
                                                                \Box j = 4
     if (age < 18)
                                                                \Box i = 0
                                                                                                                            □ A && B
     {
                                                                □ j = %d
         printf("Mineur\n");
                                                                 \Box j = 5
     }
                                                           17. Laquelle de ces écritures correspond à la déclaration
                                                                                                                         du struct, on écrit :
     else
                                                               d'une variable de type caractère en langage C?
     {
                                                                □ char 'c';
         printf("Majeur\n");
     }
                                                                ☐ int char;
                                                                ☐ char c:
                                                                ☐ char "c";
   affichera:
```

18. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
Quelle est la condition cond:
```

```
□ (a<=n) && (n<=b)
□ (n<=a) && (n<=b)
```

19. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur

```
\square !(!A \mid | B) == (A \&\& !B)
☐ (A == TRUE) && (B == TRUE)
```

20. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu

```
□ void afficher_date(struct date_s d);
☐ int afficher_date(date_s d);
□ void afficher_date(date_s d);
☐ struct date_s afficher_date(struct date_s d);
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1. Un enregistrement permet de grouper plusieurs valeurs dans: □ ses cases  $\square$  ses champs  $\square$  ses blocs  $\square$  ses chants 2. Le code suivant : int i: for (i = 1; i < 5; i = i + 1)printf("%d ", i); } printf("\n"); affichera:  $\square$  4 3 2 1  $\Box 01234$  $\Box 43210$  $\square$  1 2 3 4 3. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :  $\square$  int factorielle(int x); ☐ int factorielle(): ☐ int factorielle(double n); □ struct int factorielle(int n); 4. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?  $\square$  une fonction appelée avant sa déclaration  $\square$  une fonction déclarée mais non définie  $\Box$  un désaccord entre la déclaration et la définition d'une fonction ☐ une directive préprocesseur #include manquante

```
5. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
   printf("Majeur\n");
   affichera:
     □ Mineur
       Majeur
    □ Majeur
    ☐ Mineur
     □ rien
6. Laquelle de ces écritures correspond à la déclaration
  d'une variable de type caractère en langage C?
    ☐ char "c";
    □ char 'c';
    □ char c:
    \square int char:
7. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
 10
11
           int x = 5:
12
           int y;
 13
 14
          y = x;
 15
 16
 17
     ☐ le programme affiche "Faux"
     \square la variable y vaut 5
    \square la variable x vaut 5 et la variable y vaut 0
     \Box la variable x vaut 0
8. Le bus système sert à :
    ☐ Écrire des données sur le dique dur
     ☐ Arriver à l'heure en cours
    □ transporter les processus du tourniquet au pro-
    ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
```

```
9. Vous utilisez une boucle while quand:
      ☐ l'incrément de la variable de boucle n'est pas 1
      □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
10. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     □ 111
     \square 7
      \square 8
11. Lorsqu'un programme utilise printf ou scanf il faut
    qu'il contienne l'instruction préprocesseur :
      ☐ #include <stdio.h>
      ☐ #include <studio.h>
      ☐ #appart <stdlib.h>
      ☐ #include <studlib.h>
12. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
     \square dans lequel vous avez déclaré ces fonction
     □ un ordre quelconque
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
        main
13. Le code suivant :
     int age = 15;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
          printf("Majeur\n");
```

}

```
affichera:
      □ Majeur
     □ rien
      ☐ Mineur
      ☐ Mineur
        Majeur
14. Sous unix (ou linux), la commande cd permet de :
      \square détruire un fichier
     □ ouvir un bureau partagé (common desktop)
     □ changer de répertoire courant
     □ jouer de la musique
     \square récupérer un programme arrêté avec la commande
15. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
      \square printf("x=%d et y=%d\n",x y);
     \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n,x,y");
     \Box printf("x=%x et y=%y\n");
16. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
```

```
printf("%d ", i);
         }
     }
    printf("\n");
    qu'est ce qui sera affiché?
     \Box 0 1 0 1 0 1
     \Box 1 2 3 1 2
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2
17. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
     {
       scanf("%d", &n);
    Quelle est la condition cond:
     □ (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
      \square (a<n) || (n>b)
```

□ a<=n<=b

```
18. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \square 3
     \square 20
     \Box 6
     \Box 16
19. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     □ void afficher_date(struct date_s d);
     □ void afficher_date(date_s d);
     ☐ int afficher date(date s d):
     □ struct date_s afficher_date(struct date_s d);
20. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square exposant(double x, int n, int r);
```

 $\square$  double exposant(double x, int n);

 $\square$  int exposant(double n, int x);

☐ void exposant(double x^n);

## Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. L'ordonnancement par tourniquet permet :
  - $\square$  de doubler la mémoire disponible
  - $\square$  d'afficher des ronds colorés à l'écran
  - □ d'entretenir l'illusion que les processus tournent en parallèle
  - $\square$  de ne pas perdre de temps avec la commutation de contexte
- 2. Après exécution du programme :

```
1 lecture 8 r0
```

- 2 valeur 3 r1
- 3 mult r1 r0
- 4 valeur 1 r2
- 5 add r2 r0
- 6 ecriture r0 8
- 7 stop
- 8 5
  - $\Box$  le bus explose
  - □ le terminal affiche 8
  - □ la case mémoire 8 contiendra 16
  - $\Box \;$  la case mémoire 8 contiendra 0
- 3. Après exécution jusqu'à la ligne 15 du programme C :

```
10 int main() {
11 int x = 5;
12 int y;
13
14 y = x;
15
16 ...
17 }
```

- $\Box$ la variable x vaut 5 et la variable y vaut 0
- $\Box$ la variable x vaut 0
- $\Box$ le programme affiche "Faux"
- $\square$  la variable y vaut 5

```
4. Quelle étape de la compilation vient d'échouer lors-
qu'on a un message comme celui-ci :
```

```
Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »
```

- □ l'édition de liens
- □ l'analyse des entrées clavier
- □ l'analyse sémantique
- $\square$  l'analyse harmonique
- 5. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?

```
□ A && B
```

- $\square$  (A == TRUE) && (B == TRUE)
- $\square$  (!A || B)
- $\square$  !(!A || B) == (A && !B)
- 6. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

```
\hfill\Box void saisie_utilisateur(char c);
```

- $\hfill\Box$  void saisie\_utilisateur(int n);
- ☐ saisie\_utilisateur(scanf(%d));
- ☐ int saisie\_utilisateur();
- 7. Un programme en langage C doit comporter une et une seule définition de la fonction :

```
□ include
```

- □ init
- $\square$  begin
- $\square$  main
- 8. Soit la fonction  ${\tt f}$  définie par :

```
int f(int a)
{
  printf("a = \n", %d);
  if (a > 0)
  {
    return 3;
  }
  return 4;
```

```
Alors l'expression f(0) prendra la valeur :
```

- $\Box$  0
- $\Box$  4
- $\square$  3

{

9. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie :
```

- L'affichage dans le main est le suivant :
  - $\Box$  f(a,b)=8, a=8, b=5

int f(int a, int b)

a = a + b;

return a;

- $\Box$  f(a,b)=13, a=8, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=8, a=3, b=5
- 10. Pour l'extrait de programme suivant :

```
int somme = 0;
int serie[4] = {2, 4, 10, 4};
for (i = 0; i < 4; i = i + 1)
{
   somme = somme + serie[i];
}
printf("somme = %d",somme);</pre>
```

La valeur de somme affichée est :

- $\Box$  16
- $\Box$  6
- $\square$  20
- $\square$  3

```
11. Au début de la fonction main() on place le code :
                                                             14. Au début de la fonction main() on place le code :
                                                                                                                           17. Pour déclarer un tableau d'entiers de taille 5, on peut
                                                                                                                               utiliser l'instruction
     char i;
                                                                  char b = 'A';
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                                  b = b + 2;
                                                                                                                                 \square int[] new tableau(5);
                                                                  printf("%c\n", b);
                                                                                                                                 ☐ int toto[taille=5];
       printf("%c", i);
                                                                 Alors l'affichage sera:
                                                                                                                                 ☐ char tableau[5];
                                                                    □ b
     printf("\n");
                                                                                                                                 \square int tab[] = 5;
                                                                   \square A
    Alors l'affichage sera:
                                                                                                                                 \square int toto[5];
                                                                   □В
      ☐ ABCDEF
                                                                   \Box C
                                                                                                                           18. Vous utilisez une boucle while quand :
      □ ccccc
                                                             15. Quel est le problème d'un programme comportant les
      \Box i
                                                                                                                                 □ l'incrément de la variable de boucle n'est pas 1
                                                                 lignes suivantes?
      □ A
                                                                                                                                 □ vous avez déjà fait un for dans le même pro-
                                                                 while (1)
12. Pour l'extrait de programme suivant :
                                                                                                                                    gramme principal
     int i = 0;
                                                                    printf("coucou\n");
                                                                                                                                 □ vous n'avez pas déclaré de fonction
     int j = 0;
                                                                                                                                 \square vous ne connaissez pas le nombre d'itérations de
     for (i = 0; i < 3; i = i + 1)
                                                                   \square il n'affiche rien
                                                                                                                                    la boucle à l'avance
                                                                   \square il ne compile pas
         for (j = 0; j < 2; j = j + 1)
                                                                                                                           19. L'écriture 111 en binaire correspond au nombre natu-
                                                                   □ il comporte une boucle infinie
                                                                                                                               rel:
              printf("%d ", i);
                                                                   \square il risque d'afficher bonjour à la place de coucou
         }
                                                                                                                                 □ 111
                                                             16. Soit la fonction g définie par :
     }
                                                                                                                                 \square 7
    printf("\n");
                                                                 int g(int a)
                                                                                                                                 \square 8
    qu'est ce qui sera affiché?
                                                                    printf("a = \n", %d);
                                                                                                                                 \square 3
      \Box 0 0 1 1 2 2
                                                                    if (1 > 0)
      \Box 1 2 3 1 2
                                                                                                                           20. Si cette erreur apparaît à la compilation :
                                                                    {
      \Box 0 1 2 0 1 2
                                                                                                                               erreur: conflicting types for 'max', que doit-
                                                                      return 5;
      \Box 0 1 0 1 0 1
                                                                                                                               on chercher dans le programme?
13. Sous unix (ou linux), la commande 1s permet de :
                                                                    return 7;
                                                                                                                                 ☐ une fonction appelée avant sa déclaration
      \square afficher le contenu d'un fichier texte
                                                                                                                                 □ un désaccord entre la déclaration et la définition
      \square voir des clips musicaux
                                                                 Alors l'expression g(0) prendra la valeur :
                                                                                                                                    d'une fonction
                                                                   \Box 7
      □ compiler un programme
                                                                                                                                 ☐ une directive préprocesseur #include manquante
      \square afficher la liste de fichiers contenus dans un
                                                                   \Box 5
                                                                                                                                 \square une fonction déclarée mais non définie
         répertoire
                                                                   \Box 0
```

## Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int somme = 0;
    int i;
    for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
    printf("%d", somme);
  affichera:
     \Box 0
     \Box 6
     \Box 1
     \square 42
2. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
         {
        }
    }
   printf("j = %d\n", j);
  qu'est ce qui sera affiché par ce printf?
     \Box j = 4
     \Box j = 5
     \Box j = %d
     \Box i = 0
3. Un enregistrement permet de grouper plusieurs valeurs
  dans:
     \square ses chants
     \square ses champs
     \square ses cases
     \square ses blocs
```

```
4. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
5. Le code suivant :
    int i:
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \square 4 3 2 1
     \Box \ 4\ 3\ 2\ 1\ 0
     \square 1 2 3 4
6. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \square 3
     \square 8
     \square 7
7. Un bit est:
     \square un chiffre binaire (0 ou 1)
     \square un battement d'horloge processeur
     ☐ l'instruction qui met fin à un programme
     □ la longueur d'un mot mémoire
8. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square void exposant(double x^n);
     \square double exposant(double x, int n);
     \square exposant(double x, int n, int r);
     \square int exposant(double n, int x);
```

```
9. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ int char;
     ☐ char "c";
     □ char 'c';
     \Box char c;
10. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
      return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 4
     \square 5
      \square 1
     \square 0
11. Le code suivant :
     int age = 20;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
   affichera:
     □ Mineur
     □ rien
     ☐ Mineur
        Majeur
      □ Majeur
```

12. Après la déclaration : int mccarthy(int n);, il est	☐ l'avoir déclarée et définie	while(cond)
correct d'écrire :	$\hfill \square$ avoir défini une constante symbolique de la taille	{
$\square$ x = mccarthy(n);	de cette fonction	scanf("%d", &n); }
☐ int mccarthy(int 2);	16. Après exécution jusqu'à la ligne 14 du programme C :	
$\square$ n = mccarthy(p, q);	10 int main() {	Quelle est la condition cond :
□ n = mccarthy();	11 int x = 5;	☐ a<=n<=b
13. Vous avez déclaré préalablement un ensemble de fonc-	12	□ (a<=n) && (n<=b)
tions utilisées par votre programme principal. L'ordre	13 printf(" x = %d\n", 2); 14	☐ (n<=a) && (n<=b)
dans lequel vous devez maintenant définir ces fonctions est l'ordre :	15	☐ (a <n) (n=""   ="">b)</n)>
□ dans lequel vous avez déclaré ces fonction	16 }	
□ alphabétique	$\square$ le terminal affiche 5	19. On considère deux variables booléennes A et B initia-
□ un ordre quelconque	□ le terminal affiche "Faux"	lisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur
	$\Box$ le terminal affiche x = 5	TRUE?
$\square$ dans lequel ces fonctions sont appelées dans le main	$\Box$ le terminal affiche x = 2	☐ (A == TRUE) && (B == TRUE)
14. Après exécution jusqu'à la ligne 15 du programme $\mathcal{C}$ :	17. Quel est le problème d'un programme comportant les	□ (!A    B)
10 int main() {	lignes suivantes?	☐ !(!A    B) == (A && !B)
11 int $x = 5$ ;	while (1)	□ A && B
12 int y;	{	
13 14	<pre>printf("coucou\n");</pre>	20. Le code suivant :
15 y - x,	5	int i;
16	□ il n'affiche rien	for (i = 0; i < 7; i = i + 2)
17 }	$\square$ il comporte une boucle infinie	{
$\Box$ la variable x vaut 5 et la variable y vaut 0	☐ il risque d'afficher bonjour à la place de coucou	<pre>printf("%d ", i); }</pre>
$\Box$ la variable y vaut 5	$\square$ il ne compile pas	<pre>printf("\n");</pre>
$\Box$ la variable x vaut 0	18. On souhaite faire une boucle de contrôle de saisie : tant	affichera :
□ le programme affiche "Faux"	que l'entier $\mathbf{n}$ n'appartient pas à l'intervalle $[ab]$ , on	
15. Avant de faire appel à une fonction il est nécessaire	recommence la saisie de n. Soit le programme suivant :	$\square$ 0 1 2 3 4 5 6 7
de :	int a = 0;	$\Box \ 0\ 2\ 4\ 6\ 8$
☐ l'avoir définie	int b = 20;	$\Box \ 0 \ 2 \ 4 \ 6$
☐ l'avoir déclarée	<pre>int n; scanf("%d", &amp;n);</pre>	$\square \ 0\ 1\ 2\ 3\ 4\ 5\ 6$

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 0
     \square 3
     \Box 4
2. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
        }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box j = 4
     \Box j = 0
    \Box j = %d
    \Box j = 5
3. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
    int a = 0;
```

int b = 20;

int n;

```
scanf("%d", &n);
    while(cond)
      scanf("%d", &n);
   Quelle est la condition cond :
     □ a<=n<=b
     □ (n<=a) && (n<=b)
     \square (a<n) || (n>b)
     \square (a<=n) && (n<=b)
4. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il comporte une boucle infinie
     □ il n'affiche rien
     \square il ne compile pas
     ☐ il risque d'afficher bonjour à la place de coucou
5. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
    \square n = pgcd(int p, int q);
    \square n = pgcd(n, 3);
     \square int pgcd(2);
6. Un bit est:
     ☐ l'instruction qui met fin à un programme
     \square un chiffre binaire (0 ou 1)
     □ un battement d'horloge processeur
     □ la longueur d'un mot mémoire
7. Le langage C est un langage
     □ lu, écrit, parlé
     □ compilé
     □ composé
     □ interprété
```

```
8. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
     \square x = racine(2/3):
     \square x = racine(x * x) - racine(x):
 9. Soit le programme principal suivant :
    int main()
     int a = 3:
     int b = 5:
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
    appelant la fonction f ainsi définie :
   int f(int a, int b)
   {
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
10. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \square 15
     \square 10
      \Box 0
      \Box 6
```

11. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :	14. Si x est une variable réelle (de type double) alors x = $3/2$ lui affecte la valeur :	<ul><li>□ l'analyse des entrées clavier</li><li>□ l'analyse harmonique</li><li>□ l'édition de liens</li></ul>
<ul><li>□ en temps d'accès</li><li>□ les fichiers du disque</li></ul>	□ 0 □ 0.5	☐ l'analyse sémantique  18. Après exécution jusqu'à la ligne 14 du programme C:
☐ des processus ☐ certaines données de la mémoire de travail  12. Un fichier source est :	☐ 1.5  15. Soit la fonction g définie par :  int g(int a) {	10 int main() {     11 int x = 5;     12     13 printf(" x = %d\n", 2);
<ul> <li>□ un fichier texte qui sera traduit en instructions processeur</li> <li>□ un document qui doit être protégé</li> <li>□ un document illisible pour les humains</li> <li>□ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur</li> </ul>	<pre>printf("a = \n", %d); if (1 &gt; 0) {     return 5; } return 7;</pre>	14 15 16 }  □ le terminal affiche x = 2 □ le terminal affiche 5
□ un document de référence du système  13. Soit la fonction f définie par :	} Alors l'expression g(0) prendra la valeur :  □ 7	□ le terminal affiche "Faux" $\Box$ le terminal affiche $x = 5$
<pre>int f(int a) {     printf("a = \n", %d);     if (a &gt; 0)     {        return f(a - 1) + 1;     }     return 4; } Alors l'expression f(1) prendra la valeur :     □ 5     □ 1     □ 4</pre>	□ 5 □ 0  16. L'ordonnancement par tourniquet permet : □ de ne pas perdre de temps avec la commutation de contexte □ de doubler la mémoire disponible □ d'afficher des ronds colorés à l'écran □ d'entretenir l'illusion que les processus tournent en parallèle  17. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci : Undefined symbols :"_prinft" ou	19. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :  □ int afficher_menu(); □ double afficher_menu(printf("menu")); □ void afficher_menu(printf("menu")); □ int afficher_menu(int char);  20. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses blocs □ ses cases □ ses chants
	référence indéfinie vers « prinft »	$\square$ ses champs

## Éléments d'informatique – contrôle continue

 $\square$  15

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si a et b sont deux variables de type: struct toto\_s { int n; double x; }; Alors pour tester l'égalité de a et de b on utilise la condition:  $\Box$  a = b  $\Box$  (a.n == b.n) && (a.x == b.x)  $\square$  a{n, x} == b{n, x}  $\square$  a == b 2. L'écriture 101 en binaire correspond au nombre naturel:  $\square$  5  $\square$  3  $\Box$  4  $\square$  101 3. Le type des réels en C est :  $\square$  real □ double □ char □ int. 4. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\Box$  0  $\Box$  1  $\square$  1.5  $\square$  0.5 5. Le code suivant : int age = 20;

if (age < 18)

printf("Majeur\n");

printf("Mineur\n");

```
affichera:
     □ rien
    □ Mineur
    □ Majeur
    □ Mineur
        Majeur
6. Un fichier source est:
     \square un fichier texte qui sera traduit en instructions
        processeur
    □ un document qui doit être protégé
    □ un document de référence du système
    \square un document illisible pour les humains
    \square un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
7. Dans la commande gcc, l'option -Wall signifie :
     ☐ que l'on veut voir tous les avertissements
    □ qu'il faut lancer un déboggueur
    □ qu'il faut indenter le fichier source
    □ qu'on veut changer alétoirement de fond d'écran
8. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
    \square en parallèle, chacun dans un registre
    □ chacun son tour, après que le processus précédent
        a terminé
    \square tous ensemble
    □ tour à tour, un petit peu à chaque fois
9. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
       somme = somme + i:
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
  La valeur de somme affichée est :
     \Box 6
    \Box 10
     \square 0
```

```
10. Soit le programme principal suivant :
   int main()
     int a = 3:
     int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   }
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
11. Soit la fonction g définie par :
   int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
        return 5;
      return 7;
   Alors l'expression g(0) prendra la valeur :
     \square 7
     \Box 0
```

 $\Box$  5

12. Une segmentation fault est une erreur qui survient lorsque :	15. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :	19. Le code suivant : int age = 15;
□ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre	☐ int mccarthy(int 2);	if (age < 18)
	$\Box$ n = mccarthy(p, q);	{
du terminal	□ n = mccarthy();	<pre>printf("Mineur\n");</pre>
☐ la division du programme en zones homogènes échoue	$\Box$ x = mccarthy(n);	} else
☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	16. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	<pre>{     printf("Majeur\n");</pre>
$\hfill \square$ le programme source a été enregistré sur le disque	□ char c;	}
dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur	☐ int char;	affichera:
13. Le langage C est un langage	☐ char "c";	□ Majeur
□ interprété	☐ char 'c';	☐ Mineur
□ compilé	17. La virtualisation de la mémoire permet notamment de	$\Box$ rien
□ composé	stocker des portions inactives de la mémoire de travail	☐ Mineur
□ lu, écrit, parlé	sur le disque dur. Mais on perd :	Majeur
14. Si le code :	$\Box$ certaines données de la mémoire de travail	20. Pour l'extrait de programme suivant :
struct toto_s	$\Box$ des processus	<pre>int somme = 0;</pre>
{	$\square$ les fichiers du disque	int serie[4] = {2, 4, 10, 4};
<pre>int n; double x;</pre>	$\Box$ en temps d'accès	for (i = 0; i < 4; i = i + 1) {
};	18. On considère deux variables booléennes A et B initia-	<pre>somme = somme + serie[i];</pre>
précède la fonction main(), alors on peut écrire en	lisées à TRUE et FALSE respectivement. Parmi les ex- pressions booléennes suivantes, laquelle a pour valeur	<pre>} printf("somme = %d",somme);</pre>
début de main():	TRUE?	La valeur de somme affichée est :
$\Box \text{ toto\_s struct } z = \{3, 0.5\};$	□ (!A    B)	
☐ int struct toto_s = {3, -1e10};	☐ !(!A    B) == (A && !B)	
□ struct toto_s toto;		
☐ int toto.n = 3;	□ A && B	□ 3 □
□ toto_s n, x;	☐ (A == TRUE) && (B == TRUE)	$\Box$ 16

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme? □ une fonction déclarée mais non définie □ un désaccord entre la déclaration et la définition d'une fonction ☐ une fonction appelée avant sa déclaration ☐ une directive préprocesseur #include manquante 2. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0){ return 3; } return 4: Alors l'expression f(0) prendra la valeur :  $\Box$  4  $\square$  3  $\Box$  0 3. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :  $\square$  printf("Valeur de n ? %g\n", n); □ printf("Valeur de n ? %d\n", n);  $\square$  un débogueur □ scanf("%d", &n); 4. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  int carre(2);  $\square$  int n = carre();  $\square$  n = carre(int n);  $\square$  n = carre(n):

```
5. Le code suivant :
   int age = 18;
   if (age < 18)
   {
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
  affichera:
    □ Mineur
       Majeur
    □ Mineur
    □ rien
    □ Majeur
6. Les lignes
  int i;
  int x=0;
  for(i=0,i<5,i=i+1)
    x=x+1;
    □ comportent une erreur qui ne sera pas détectée
    \square ne comportent aucune erreur
    □ comportent une erreur qui sera détectée au cours
       de l'analyse syntaxique
    □ comportent une erreur qui sera détectée au cours
       de l'édition de lien
7. Sous unix (ou linux), pour créer un répertoire TP4
  dans le répertoire courant on peut utiliser la com-
  mande:
    □ mkdir TP4
    □ kwrite TP4
    □ new TP4
    ☐ yppasswd
```

```
8. Si a et b sont deux variables de type:
   struct toto_s
      int n;
      double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \square a{n, x} == b{n, x}
     \square a = b
     \Box (a.n == b.n) && (a.x == b.x)
     \square a == b
9. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
10. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \square 101
     \Box 5
     \Box 4
     \square 3
11. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     ☐ une fonction appelée avant sa déclaration
     □ une fonction déclarée mais non définie
     ☐ une directive préprocesseur #include manquante
     □ un désaccord entre la déclaration et la définition
        d'une fonction
```

```
15. Afin de représenter la taille d'un tableau, définir une
12. Pour l'extrait de programme suivant :
                                                                                                                           18. Soient deux variables entières x et y initialisées à 4 et
                                                                 constante symbolique N valant 3.
                                                                                                                               5 respectivement. L'affichage x=4 et y=5 est obtenu
      int produit = 0;
                                                                                                                               avec la commande :
      int serie[4] = \{2, 2, 2, 2\};
                                                                   ☐ #define taille = N
      for (i = 0; i < 4; i = i + 1)
                                                                                                                                 \square printf("x=%x et y=%y\n");
                                                                   ☐ #define taille = 3
      {
                                                                                                                                \square printf("x=%d et y=%d\n",x y);
         produit = produit * serie[i];
                                                                   ☐ #define N 3
                                                                                                                                \square printf("x=%d et y=%d\n,x,y");
                                                                   \square #define N = 3
                                                                                                                                \square printf("x=%d et y=%d\n",x,y);
      printf("produit = %d", produit);
    La valeur affichée est :
                                                                                                                           19. Un bit est:
                                                             16. Lorsqu'un programme utilise printf ou scanf il faut
      \square 8
                                                                 qu'il contienne l'instruction préprocesseur :
                                                                                                                                 □ un battement d'horloge processeur
      \Box 0
                                                                                                                                \square la longueur d'un mot mémoire
                                                                   ☐ #appart <stdlib.h>
      \Box 16
                                                                                                                                 ☐ l'instruction qui met fin à un programme
                                                                   ☐ #include <studio.h>
      \Box 4
                                                                                                                                \square un chiffre binaire (0 ou 1)
                                                                   ☐ #include <studlib.h>
13. Un fichier source est:
                                                                                                                          20. Soit la fonction g définie par :
      \Box un fichier texte qui sera traduit en instructions
                                                                   ☐ #include <stdio.h>
         processeur
                                                                                                                               int g(int a)
                                                             17. Quel est le problème d'un programme comportant les
      □ un fichier que l'ont doit citer dans les documents
                                                                                                                              {
                                                                 lignes suivantes?
         produits sur l'ordinateur
                                                                                                                                 printf("a = \n", %d);
      \Box un document de référence du système
                                                                                                                                 if (1 > 0)
                                                                 while (1)
                                                                                                                                 {
      \square un document illisible pour les humains
                                                                 {
                                                                                                                                   return 5;
      □ un document qui doit être protégé
                                                                   printf("coucou\n");
14. Pour déclarer une fonction exposant qui prend en ar-
                                                                                                                                 return 7:
    gument un réel x et un entier positif n et renvoie la
                                                                   \Box\,il ne compile pas
    valeur de x^n on écrit :
                                                                                                                               Alors l'expression g(0) prendra la valeur :
      \square exposant(double x, int n, int r);
                                                                   \square il risque d'afficher bonjour à la place de coucou
                                                                                                                                \Box 7
      \square int exposant(double n, int x);
                                                                   \square il comporte une boucle infinie
                                                                                                                                 \Box 0
      \square void exposant(double x^n);
                                                                   \square il n'affiche rien
      \square double exposant(double x, int n);
                                                                                                                                 \square 5
```

## Éléments d'informatique - contrôle continue

 $\square$  7

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int i:
    for (i = 0; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
  affichera:
    \Box 01234
    \Box 43210
    \square 0 1 2 3
    \square 4 3 2 1
2. Soit la fonction g définie par :
  int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
     \Box 7
     \Box 0
     \square 5
3. Le code suivant :
    int somme = 0;
    int i;
    for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
    printf("%d", somme);
  affichera:
```

```
\Box 0
    \Box 6
    \Box 1
    \square 42
4. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
5. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
    □ un désaccord entre la déclaration et la définition
        d'une fonction
    \square une fonction déclarée mais non définie
    ☐ une fonction appelée avant sa déclaration
    ☐ une directive préprocesseur #include manquante
6. L'écriture 111 en binaire correspond au nombre natu-
  rel:
    \square 3
    \square 111
    \square 8
```

```
7. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
          printf("%d ", i);
     printf("\n");
   affichera:
     \Box 02468
     \Box 8 6 4 2
      \square 8 2
      \Box 8 6 4 2 0
 8. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
      \square for(i=1;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
      \square for(i=0;i<5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
 9. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square n = pgcd(int p, int q);
     \square n = pgcd(n, 3);
     \square int pgcd(2);
     \square int n = pgcd();
10. Le langage C est un langage
      □ composé
      □ compilé
      □ lu, écrit, parlé
      □ interprété
11. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
      \square double exposant(double x, int n);
      \square int exposant(double n, int x);
```

 $\square$  exposant(double x, int n, int r);

 $\square$  void exposant(double x^n);

```
12. Le code suivant :
                                                              15. Si cette erreur apparaît à la compilation :
                                                                  error: expected ';' before '}' token que doit-
     int age = 18;
                                                                  on chercher dans le programme?
     if (age < 18)
                                                                                                                               d'écrire :
                                                                    □ un point-virgule manquant
     {
         printf("Mineur\n");
                                                                    ☐ une accolade manquante
     }
                                                                    \square une accolade en trop
     else
                                                                    \square un point-virgule en trop
     {
                                                              16. Une variable booléenne est un variable :
         printf("Majeur\n");
                                                                    □ jamais nulle
     }
                                                                    □ à laquelle une valeur vient d'être affectée
                                                                    □ qui est vraie ou fausse
    affichera:
                                                                    ☐ réelle positive
      □ Majeur
                                                                    □ NaN (not a number, qui n'est pas un nombre)
      ☐ Mineur
                                                             17. On considère deux variables booléennes A et B initia-
      ☐ Mineur
                                                                  lisées à TRUE et FALSE respectivement. Parmi les ex-
         Majeur
                                                                  pressions booléennes suivantes, laquelle a pour valeur
      \square rien
                                                                  TRUE?
13. Dans la commande gcc, l'option -Wall signifie :
                                                                    \square !(!A \mid | B) == (A \&\& !B)
      \square que l'on veut voir tous les avertissements
                                                                    \square (A == TRUE) && (B == TRUE)
      □ qu'il faut lancer un déboggueur
      □ qu'on veut changer alétoirement de fond d'écran
                                                                    \square (!A || B)
                                                                                                                                La valeur de somme affichée est :
                                                                    □ A && B
      \square qu'il faut indenter le fichier source
14. Après la déclaration : int mccarthy(int n);, il est
                                                              18. Laquelle de ces écritures correspond à la déclaration
                                                                                                                                  \Box 16
                                                                  d'une variable de type caractère en langage C?
    correct d'écrire :
                                                                                                                                  \square 20
                                                                    ☐ int char;
      \square n = mccarthy();
      \square n = mccarthy(p, q);
                                                                    \Box char c;
                                                                                                                                  \Box 6
      \square x = mccarthy(n);
                                                                    ☐ char "c";
                                                                                                                                  \square 3
      ☐ int mccarthy(int 2);
                                                                    □ char 'c';
```

```
19. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
    une variable entière définie et initialisée, il est correct
     \square int n = carre();
     \square n = carre(int n);
      \Box int carre(2);
      \square n = carre(n);
20. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
```

## Eléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique); -0.5 points

r reponse lausse. Duree : 20 minutes.	
1. On considère deux variables booléennes A et B initi- lisées à TRUE et FALSE respectivement. Parmi les et pressions booléennes suivantes, laquelle a pour valet TRUE?  \[ \begin{aligned} \left(!A \right  B) \\ \begin{aligned} \left(A \infty B) \\ \begin{aligned} \left(!A \right  B) \\ \begin{aligned} \left(!A \right  B) \\ \begin{aligned} \left(A \infty B) \\ \ext{aligned} \\ align	X-
2. Pour l'extrait de programme suivant :	
<pre>int i; int j; for(i=4;i&gt;0;i=i-1) {    for(j=i;j&lt;6;j=j+1)    {       printf("*");    }    printf(" "); }</pre>	
qu'est ce qui sera affiché?	
_ ** ** ** ** **	
<pre> ***** *** ***</pre>	
<pre> ** *** **** *****</pre>	
<pre> **** **** ****</pre>	
3. Après exécution du programme :	
<pre>1 lecture 8 r0 2 valeur 3 r1 3 mult r1 r0 4 valeur 1 r2 5 add r2 r0 6 ecriture r0 8 7 stop 8 5</pre>	
$\Box$ le terminal affiche 8	
$\Box$ la case mémoire 8 contiendra 0	
$\square$ le bus explose	
□ la case mémoire 8 contiendra 16	

```
4. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <stdio.h>
     ☐ #include <studio.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studlib.h>
5. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     ☐ int factorielle(int 2):
     □ printf("%d", factorielle(n));
     \square n = factorielle();
     \square n = factorielle(p, q);
6. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int n = carre();
    \square n = carre(int n);
     \square int carre(2);
     \square n = carre(n);
7. Sous unix (ou linux), la commande 1s permet de :
     □ afficher le contenu d'un fichier texte
     □ afficher la liste de fichiers contenus dans un
        répertoire
     □ compiler un programme
     □ voir des clips musicaux
8. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il risque d'afficher bonjour à la place de coucou
    \square il ne compile pas
     □ il n'affiche rien
     □ il comporte une boucle infinie
```

```
9. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
    on chercher dans le programme?
      ☐ une fonction appelée avant sa déclaration
      □ un désaccord entre la déclaration et la définition
         d'une fonction
      ☐ une directive préprocesseur #include manquante
      \square une fonction déclarée mais non définie
10. Si a et b sont deux variables de type :
    struct toto s
      int n:
      double x;
   };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
     \square (a.n == b.n) \&\& (a.x == b.x)
     \Box a = b
     \square a\{n, x\} == b\{n, x\}
      □ a == b
11. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
      \square #define N = 3
      \square #define taille = 3
      ☐ #define N 3
      \square #define taille = N
12. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre:
      □ alphabétique
      □ un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
      □ dans lequel ces fonctions sont appelées dans le
```

main

15.	Un nemer source est:	15. Pour de
	$\Box$ un document qui doit être protégé	en argu
	$\Box$ un document de référence du système	du struc
	$\square$ un fichier texte qui sera traduit en instructions	□ in
	processeur	□ vo
	$\Box$ un fichier que l'ont doit citer dans les documents	□ vo
	produits sur l'ordinateur	□ st:
	$\Box$ un document illisible pour les humains	16. Le code
14.	Soit un programme contenant les lignes suivantes :	int i
	int i = 0;	for (:
	int $j = 0$ ;	{
	for (i = 0; i < 0; i = i + 1)	pı
	{	}
	for (j = 0; j < 5; j = j + 1) {	print
	ì	affichera
	}	
	}	
	<pre>printf("j = %d\n", j);</pre>	
	•••	
	}	17. Pour at
		le conte
	qu'est ce qui sera affiché?	préalabl
	□ j = %d	☐ fo
	$\Box$ j = 0	☐ fo
	□ j = 5	☐ fo:
	$\Box$ j = 4	☐ fo

13. Un fichier source est :	15. Pour déclarer une procédure afficher_date qui prend	18. Laquelle de ces écritures correspond à la déclaration
□ un document qui doit être protégé	en argument un struct date_s et affiche le contenu	d'une variable de type caractère en langage C?
$\square$ un document de référence du système	du struct, on écrit :	☐ char 'c';
$\square$ un fichier texte qui sera traduit en instructions	☐ int afficher_date(date_s d);	
processeur	☐ void afficher_date(date_s d);	$\square$ int char;
$\square$ un fichier que l'ont doit citer dans les documents	$\square$ void afficher_date(struct date_s d);	□ char c;
produits sur l'ordinateur	☐ struct date_s afficher_date(struct date_s	
$\square$ un document illisible pour les humains	16. Le code suivant :	☐ char "c";
14. Soit un programme contenant les lignes suivantes :	int i;	19. Un enregistrement permet de grouper plusieurs valeurs
int i = 0;	for $(i = 4; i \ge 0; i = i - 1)$	dans:
int j = 0;	{	
for (i = 0; i < 0; i = i + 1)	<pre>printf("%d ", i);</pre>	$\square$ ses champs
(	}	$\square$ ses chants
for (j = 0; j < 5; j = j + 1) {	<pre>printf("\n");</pre>	
	affichera:	☐ ses blocs
· · · · }	$\square \ 4\ 3\ 2\ 1$	□ ses cases
}	$\square$ 1 2 3 4	
printf("j = %d\n", j);	$\Box \ 4\ 3\ 2\ 1\ 0$	20. Si cette erreur apparaît à la compilation :
•••	$\Box \ 0\ 1\ 2\ 3\ 4$	error: expected ';' before '}' token que doit-
}	17. Pour afficher à l'aide de printf("%d\n",tab[i]);	on chercher dans le programme?
	le contenu d'un tableau de 5 entiers initialisé au	$\Box$ une accolade en trop
qu'est ce qui sera affiché?	préalable, on utilise plutôt :	inc accolate on trop
□ j = %d	$\square for(i=0;i<=5;i=i+1)$	$\square$ une accolade manquante
□ j = 0	☐ for(i=1;i<=5;i=i+1)	☐ un point-virgule en trop
□ j = 5	☐ for(i=0;i<5;i=i+1)	
□ j = 4	☐ for(i=1;i<5;i=i+1)	$\Box$ un point-virgule manquant
	•	

## Éléments d'informatique – contrôle continue

D /	NT.
Prénom:	Nom:
N° INE ·	
1, 11,12,	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
  - $\Box$  dans lequel ces fonctions sont appelées dans le main
  - □ alphabétique
  - $\square$  un ordre quelconque
  - $\hfill \square$  dans lequel vous avez déclaré ces fonction
- 2. Après exécution jusqu'à la ligne 14 du programme C :

```
10  int main() {
11     int x = 5;
12
13     printf(" x = %d\n", 2);
14
15     ...
16  }
```

- $\Box$ le terminal affiche x = 5
- $\square$  le terminal affiche x = 2
- □ le terminal affiche "Faux"
- $\square$  le terminal affiche 5
- 3. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
  printf("coucou\n");
}
```

- $\Box\:$ il comporte une boucle infinie
- $\square$  il ne compile pas
- $\Box$ il risque d'afficher bonjour à la place de coucou
- $\Box\,$ il n'affiche rien

4. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

	x	=	racine(2/3);	
	x	=	<pre>racine(x * x) - racine(x)</pre>	);
_				

- \[ x 1 = racine(x);
  \[ x = racine(racine(x)\*racine(x));
- 5. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

```
\hfill\Box void saisie_utilisateur(int n);
```

- ☐ int saisie\_utilisateur();
- $\hfill\Box$  void saisie\_utilisateur(char c);
- ☐ saisie\_utilisateur(scanf(%d));
- 6. Dans la commande gcc, l'option -Wall signifie :
- □ qu'on veut changer alétoirement de fond d'écran
  - $\hfill\Box$  qu'il faut indenter le fichier source
  - $\square$  qu'il faut lancer un déboggueur
  - $\square$  que l'on veut voir tous les avertissements
- 7. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);
...
}</pre>
```

qu'est ce qui sera affiché?

- $\Box$  j = 4
- $\Box$  j = %d
- $\Box$  j = 0
- □ j = 5

- 8. Avant de faire appel à une fonction il est nécessaire de :
  - □ l'avoir déclarée
  - □ l'avoir définie
  - □ avoir défini une constante symbolique de la taille de cette fonction
  - $\Box$ l'avoir déclarée et définie
- 9. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
  - $\square$  #define N 3
  - $\square$  #define N = 3
  - $\square$  #define taille = 3
  - $\square$  #define taille = N
- 10. Le type des réels en C est :
  - $\square$  char
  - $\square$  int
  - $\square$  real
  - $\square$  double
- 11. Le code suivant :

```
int age = 18;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");
}</pre>
```

- affichera:
  - $\square$  rien
  - ☐ Mineur Majeur
- ☐ Mineur
- $\square$  Majeur

12. Après exécution jusqu'à la ligne 15 du programme ${\bf C}$ :	□ a<=n<=b	☐ int afficher_date(date_s d);
10 int main() {	□ (a <n) (n=""   ="">b)</n)>	☐ void afficher_date(date_s d);
$ \begin{array}{ll} 11 & \text{int } x = 5; \\ 12 & \text{int } x = 5; \end{array} $	14. Lorsqu'un programme utilise printf ou scanf il faut	☐ void afficher_date(struct date_s d);
12 int y = 3; 13	qu'il contienne l'instruction préprocesseur :	18. Si cette erreur apparaît à la compilation :
14   x = y;	☐ #appart <stdlib.h></stdlib.h>	erreur: conflicting types for 'max', que doit- on chercher dans le programme?
15 16	☐ #include <stdio.h></stdio.h>	<u> </u>
16 17 }	☐ #include <studlib.h></studlib.h>	☐ une fonction déclarée mais non définie
•	$\square$ #include <studio.h></studio.h>	$\Box$ une directive préprocesseur $\# \mathtt{include}$ manquante
$\Box$ la variable y vaut 5 $\Box$ la variable x vaut 3	15. Un enregistrement permet de grouper plusieurs valeurs dans :	☐ un désaccord entre la déclaration et la définition d'une fonction
$\Box$ la variable x vaut 5 et la variable y vaut 3	□ ses blocs	$\hfill\Box$ une fonction appelée avant sa déclaration
$\Box$ le programme affiche "Faux"	$\square$ ses champs	19. Soient deux variables entières x et y initialisées à 4 et
13. On souhaite faire une boucle de contrôle de saisie : tant	□ ses cases	5 respectivement. L'affichage x=4 et y=5 est obtenu
que l'entier $\mathbf{n}$ n'appartient pas à l'intervalle $[ab]$ , on	□ ses chants	avec la commande :
recommence la saisie de ${\tt n}.$ Soit le programme suivant :	16. Le bus système sert à :	$\Box$ printf("x=%d et y=%d\n",x,y);
<pre>int a = 0;</pre>	☐ Arriver à l'heure en cours	$\Box$ printf("x=%d et y=%d\n,x,y");
<pre>int b = 20; int n;</pre>	☐ Écrire des données sur le dique dur	$\Box$ printf("x=%d et y=%d\n",x y);
scanf("%d", &n);	☐ transporter les processus du tourniquet au pro-	$\Box$ printf("x=%x et y=%y\n");
while(cond)	cesseur	20. Sous unix (ou linux), la commande 1s permet de :
{	☐ Transférer des données et intructions entre pro-	□ voir des clips musicaux
scanf("%d", &n); }	cesseur et mémoire	□ compiler un programme
Quelle est la condition cond :	17. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu	□ afficher le contenu d'un fichier texte
☐ (a<=n) && (n<=b)	du struct, on écrit :	$\square$ afficher la liste de fichiers contenus dans un
☐ (n<=a) && (n<=b)	☐ struct date_s afficher_date(struct date_s	d); répertoire

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE ·	
1, 11,12,	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Sur unix (ou linux), la commande mkdir permet de : ouvrir un fichier texte  $\square$  créer un fichier texte □ changer de répertoire courant  $\square$  créer un répertoire 2. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ dans lequel ces fonctions sont appelées dans le main □ alphabétique □ dans lequel vous avez déclaré ces fonction  $\square$  un ordre quelconque 3. Quels calculs peut-on programmer en programmation structurée? □ certains programmes sont de vrais plats de spaghetti □ en programmation structurée on peut programmer tous les calculs programmables en langage machine ☐ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée  $\square$  il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine 4. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1)for (j = 0; j < 5; j = j + 1)}

 $printf("j = %d\n", j);$ 

```
qu'est ce qui sera affiché par ce printf?
    \Box i = 0
     \Box j = 4
     \Box j = %d
    \Box j = 5
5. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
     ☐ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
     □ printf("Valeur de n ? %g\n", n);
     □ un débogueur
6. L'écriture 111 en binaire correspond au nombre natu-
  rel:
     \Box 7
     □ 111
     \square 8
     \square 3
7. Un registre du processeur est :
     \square un composant qui contient la liste des fichiers du
        système
     □ une unité de calcul spécialisée de l'ordinateur
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     □ une gamme de fréquence de fonctionnement du
8. Si pgcd est une fonction prenant en entrée deux entiers
  et renvoyant un entier, il est correct d'écrire :
     \square int pgcd(2);
     \square int n = pgcd();
    \square n = pgcd(n, 3);
    \square n = pgcd(int p, int q);
9. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
  on chercher dans le programme?
     \square une accolade manquante
     ☐ un point-virgule manquant
    \square un point-virgule en trop
     \square une accolade en trop
```

```
10. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square n = mccarthy();
     \Box int mccarthy(int 2);
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
11. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
    Alors l'affichage sera:
     ПЪ
     \square A
     \Box C
      ПВ
12. Le code suivant :
     int age = 15;
     if (age < 18)
     {
          printf("Mineur\n");
     else
     {
          printf("Majeur\n");
     }
   affichera:
     □ Majeur
     ☐ Mineur
        Majeur
     □ Mineur
      □ rien
13. Un bit est:
      □ la longueur d'un mot mémoire
     \square un chiffre binaire (0 ou 1)
     ☐ l'instruction qui met fin à un programme
      □ un battement d'horloge processeur
```

14. Pour déclarer une fonction saisie_utilisateur qui	17. Avant de faire appel
demande à l'utilisateur d'entrer un entier au clavier et	de:
renvoie cet entier on écrit :	☐ l'avoir déclarée e
☐ saisie_utilisateur(scanf(%d));	□ avoir défini une
☐ void saisie_utilisateur(char c);	de cette fonction
$\square$ void saisie_utilisateur(int n);	□ l'avoir définie
☐ int saisie_utilisateur();	□ l'avoir déclarée
15. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les ex-	18. Soit un programme c
pressions booléennes suivantes, laquelle a pour valeur	int i = 0;
TRUE?	int j = 0;
$\square !(!A    B) == (A \&\& !B)$	for (i = 0; i < 0
□ A && B	{
□ (!A    B)	for (j = 0; j
☐ (A == TRUE) && (B == TRUE)	{
16. Soient deux variables entières x et y initialisées à 4 et	}
5 respectivement. L'affichage x=4 et y=5 est obtenu	}
avec la commande :	printf("j = %d\n"
$\square$ printf("x=%x et y=%y\n");	
$\square$ printf("x=%d et y=%d\n",x,y);	}
$\Box$ printf("x=%d et y=%d\n,x,y");	
$\Box$ printf("x=%d et y=%d\n",x y);	qu'est ce qui sera affi

```
à une fonction il est nécessaire
                                        \Box j = 5
                                        \Box j = 4
et définie
                                        \Box j = 0
constante symbolique de la taille
                                        \Box j = %d
                                  19. Pour déclarer une procédure afficher_menu sans ar-
                                      gument et qui ne renvoie rien on utilise:
                                        ☐ int afficher_menu(int char);
contenant les lignes suivantes :
                                        □ void afficher_menu();
                                        ☐ char afficher_menu(printf("menu"));
0; i = i + 1)
                                        ☐ int afficher_menu();
 < 5; j = j + 1)
                                        ☐ double afficher_menu();
                                  20. Le bus système sert à :
                                        ☐ Arriver à l'heure en cours
                                        □ Transférer des données et intructions entre pro-
', j);
                                           cesseur et mémoire
                                        ☐ Écrire des données sur le dique dur
                                        \Box transporter les processus du tourniquet au pro-
iché?
                                           cesseur
```

# Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

par réponse fausse. Durée : 20 minutes.

- Barème: 1 points par réponse juste (unique); -0.5 points 1. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de  $x^n$  on écrit :  $\square$  exposant(double x, int n, int r);  $\square$  int exposant(double n, int x);  $\square$  void exposant(double x^n);  $\square$  double exposant(double x, int n); 2. L'ordonnancement par tourniquet permet : □ d'afficher des ronds colorés à l'écran  $\square$  de ne pas perdre de temps avec la commutation de contexte  $\square$  de doubler la mémoire disponible □ d'entretenir l'illusion que les processus tournent en parallèle
  - 3. Le code suivant :

 $\square$  1 2 3 4

```
int i:
 for (i = 4; i \ge 0; i = i - 1)
     printf("%d ", i);
printf("\n");
affichera:
  \Box 01234
 \Box 43210
  \square 4 3 2 1
```

4. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise:

```
☐ int afficher_menu();
□ void afficher_menu();
☐ int afficher_menu(int char);
☐ char afficher_menu(printf("menu"));
☐ double afficher_menu();
```

```
5. Vous utilisez une boucle while quand:
    □ vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
```

- □ vous n'avez pas déclaré de fonction
- ☐ l'incrément de la variable de boucle n'est pas 1
- □ vous avez déjà fait un for dans le même programme principal
- 6. Pour l'extrait de programme suivant :

```
int produit = 1;
int serie[4] = \{2, 2, 2, 2\};
for (i = 0; i < 4; i = i + 1)
 produit = produit * serie[i];
printf("produit = %d", produit);
```

La valeur affichée est :

- $\Box$  16  $\square$  8
- $\square$  0
- $\Box$  4
- 7. Le langage C est un langage
  - □ lu, écrit, parlé
  - □ compilé
  - □ interprété
  - □ composé
- 8. Une variable booléenne est un variable :
  - ☐ qui est vraie ou fausse
  - ☐ réelle positive
  - □ à laquelle une valeur vient d'être affectée
  - □ jamais nulle
  - □ NaN (not a number, qui n'est pas un nombre)
- 9. Soit le programme principal suivant :

```
int main()
{
 int a = 3;
 int b = 5;
 printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
 return EXIT_SUCCESS;
```

```
appelant la fonction f ainsi définie :
int f(int a, int b)
  a = a + b;
  return a;
L'affichage dans le main est le suivant :
 \Box f(a,b)=8, a=3, b=5
 \Box f(a,b)=8, a=8, b=5
 \Box f(3,5)=8, a=3, b=5
 \Box f(a,b)=13, a=8, b=5
```

- 10. Avant de faire appel à une fonction il est nécessaire de:
  - □ l'avoir déclarée et définie
  - □ l'avoir déclarée
  - □ l'avoir définie
  - □ avoir défini une constante symbolique de la taille de cette fonction
- 11. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\square x = racine(x * x) - racine(x):
\square x - 1 = racine(x):
\square x = racine(2/3);
\square x = racine(racine(x)*racine(x));
```

- 12. Un fichier source est:
  - □ un document qui doit être protégé
  - □ un document illisible pour les humains
  - □ un fichier texte qui sera traduit en instructions processeur
  - $\square$  un document de référence du système
  - $\square$  un fichier que l'ont doit citer dans les documents produits sur l'ordinateur

```
13. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
     printf("\n");
    affichera:
      \Box 0 1 2 3 4 5 6
      \Box 0246
      \Box 0 1 2 3 4 5 6 7
      \Box 02468
14. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse syntaxique
     \square analyse lexicale
      □ analyse sémantique
      \square analyse harmonique
15. Le bus système sert à :
      ☐ Écrire des données sur le dique dur
     □ Transférer des données et intructions entre pro-
         cesseur et mémoire
      ☐ Arriver à l'heure en cours
      □ transporter les processus du tourniquet au pro-
         cesseur
```

```
16. Le type des réels en C est :
     □ real
     \square int
     □ double
     \Box char
17. Si le code:
    struct toto_s
      int n;
      double x;
   };
    précède la fonction main(), alors on peut écrire en
   début de main():
     □ struct toto_s toto;
     \square int toto.n = 3;
     \Box toto_s struct z = {3, 0.5};
     \Box toto_s n, x;
     \Box int struct toto_s = {3, -1e10};
18. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
```

 $\square$  (A == TRUE) && (B == TRUE)

```
□ A && B
     \square (!A || B)
      \square !(!A \mid | B) == (A \&\& !B)
19. Dans la commande gcc, l'option -Wall signifie :
      \square qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut indenter le fichier source
     \square que l'on veut voir tous les avertissements
     □ qu'il faut lancer un déboggueur
20. Si a et b sont deux variables de type:
    struct toto_s
   {
      int n:
      double x;
   };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
     \Box a = b
     \square (a.n == b.n) && (a.x == b.x)
      □ a == b
```

 $\square a\{n, x\} == b\{n, x\}$ 

## Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
  - □ dans lequel ces fonctions sont appelées dans le main
  - $\Box\,$  dans lequel vous avez déclaré ces fonction
  - $\Box$ alphabétique
  - $\Box$  un ordre quel<br/>conque
- 2. Dans la commande gcc, l'option -Wall signifie :
  - $\square$  que l'on veut voir tous les avertissements
  - $\Box$  qu'on veut changer alétoirement de fond d'écran
  - □ qu'il faut lancer un déboggueur
  - $\square$  qu'il faut indenter le fichier source
- 3. Au début de la fonction main() on place le code :

```
char i;
for (i = 'A'; i <= 'F'; i = i + 1)
{
   printf("%c", i);
}
printf("\n");</pre>
```

Alors l'affichage sera :

- $\square$  ABCDEF
- $\square$  i
- $\square$  ccccc
- $\square$  A
- 4. Sous unix (ou linux), la commande cd permet de :
  - $\Box$ jouer de la musique
  - $\Box$  changer de répertoire courant
  - $\Box$ ouvir un bureau partagé (common desktop)
  - $\hfill \square$ récupérer un programme arrêté avec la commande ab
  - □ détruire un fichier

```
5. Soit la fonction f définie par :
```

```
int f(int a)
{
   printf("a = \n", %d);
   if (a > 0)
   {
      return 3;
   }
   return 4;
}
```

Alors l'expression f(0) prendra la valeur :

- $\square$  3
- $\Box$  0
- $\Box$  4
- 6. Au début de la fonction main() on place le code :

```
char b = 'A';
b = b + 2;
printf("%c\n", b);
```

 ${\bf Alors\ l'affichage\ sera:}$ 

- $\square$  A
- $\Box$  b
- $\Box$  C
- $\square$  B

7. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :

- $\Box$  en temps d'accès
- $\square$  des processus
- $\Box$  certaines données de la mémoire de travail
- $\square$  les fichiers du disque
- 8. L'écriture <u>111</u> en binaire correspond au nombre naturel :
  - $\square$  3
  - $\Box$  7
  - $\square$  8
  - □ 111

9. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :

```
\square n = mccarthy();
```

- $\square$  x = mccarthy(n);
- $\square$  int mccarthy(int 2);
- $\square$  n = mccarthy(p, q);
- 10. Après exécution jusqu'à la ligne 14 du programme C :

```
10  int main() {
11    int x = 5;
12
13    printf(" x = %d\n", 2);
14
15    ...
16 }
```

- $\square$  le terminal affiche x = 2
- $\square$  le terminal affiche x = 5
- $\Box$ le terminal affiche "Faux"
- $\square$  le terminal affiche 5
- 11. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie:
int f(int a, int b)
```

- {
   a = a + b;
   return a;
- .

L'affichage dans le main est le suivant :

- $\Box$  f(a,b)=8, a=8, b=5
- $\Box$  f(a,b)=8, a=3, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=13, a=8, b=5

```
12. Si a et b sont deux variables de type :
                                                                 qu'est ce qui sera affiché?
                                                                                                                               affichera:
                                                                   \Box 1 2 3 1 2
   struct toto_s
                                                                                                                                 \Box 0
   {
                                                                   \Box 0 1 0 1 0 1
                                                                                                                                 \Box 42
      int n;
                                                                   \Box 0 1 2 0 1 2
      double x:
                                                                   \Box 0 0 1 1 2 2
                                                                                                                                 \Box 1
   };
                                                             16. Le code suivant :
                                                                                                                                 \Box 6
   Alors pour tester l'égalité de a et de b on utilise la
                                                                  int i;
   condition:
                                                                                                                           19. Si carre est une fonction prenant en entrée un en-
                                                                  for (i = 1; i < 5; i = i + 1)
      \Box a = b
                                                                                                                               tier et renvoyant le carré de cet entier, et que n est
                                                                  {
      □ a == b
                                                                                                                               une variable entière définie et initialisée, il est correct
                                                                       printf("%d ", i);
      \square a{n, x} == b{n, x}
                                                                                                                               d'écrire :
                                                                  printf("\n");
      \square (a.n == b.n) && (a.x == b.x)
                                                                                                                                 \square int carre(2);
13. Si cette erreur apparaît à la compilation :
                                                                 affichera:
                                                                                                                                 \square n = carre(int n);
   Undefined symbols :"_prinft" ou
                                                                   \square 4 3 2 1
   référence indéfinie vers « prinft » que doit-
                                                                                                                                 \square n = carre(n):
                                                                   \Box 01234
   on chercher dans le programme?
                                                                   \square 1 2 3 4
                                                                                                                                 \square int n = carre();
      \Box une directive préprocesseur \verb"#include" manquante
                                                                   \Box 43210
     \square une faute de frappe dans un appel de fonction
                                                                                                                           20. Le code suivant :
                                                             17. Quel est le problème d'un programme comportant les
      \square un caractère interdit en C
                                                                 lignes suivantes?
                                                                                                                                int age = 18;
      □ une variable non déclarée
                                                                 while (1)
                                                                                                                                if (age < 18)
14. Laquelle des analyses suivantes ne fait pas partie des
                                                                 {
                                                                                                                                {
   étapes de la compilation :
                                                                    printf("coucou\n");
                                                                                                                                     printf("Mineur\n");
      \square analyse sémantique
                                                                                                                                }
     \square analyse harmonique
                                                                   \square il risque d'afficher bonjour à la place de coucou
                                                                                                                                else
     \square analyse lexicale
                                                                                                                                {
                                                                   \square il comporte une boucle infinie
      \square analyse syntaxique
                                                                                                                                     printf("Majeur\n");
                                                                   \square il n'affiche rien
                                                                                                                                }
15. Pour l'extrait de programme suivant :
                                                                   \square il ne compile pas
     int i = 0:
                                                             18. Le code suivant :
     int j = 0;
                                                                                                                               affichera:
     for (i = 0; i < 3; i = i + 1)
                                                                  int somme = 0;
                                                                                                                                 \square rien
                                                                  int i;
                                                                  for (i = 1; i < 4; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
                                                                                                                                 □ Mineur
              printf("%d ", i);
                                                                                                                                 □ Majeur
                                                                     somme = somme + i;
         }
                                                                                                                                 ☐ Mineur
                                                                  printf("%d", somme);
                                                                                                                                    Majeur
    printf("\n");
```

т		-1
	acence	

# Éléments d'informatique – contrôle continue

Prónom ·	Nom ·
Prenom:	Nom .
NO INIT .	
IN IINE:	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant : int a = 0;

```
int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
      scanf("%d", &n);
   Quelle est la condition cond :
     \square (n<=a) && (n<=b)
     \square (a<n) || (n>b)
     □ a<=n<=b
     \square (a<=n) && (n<=b)
2. Pour déclarer un tableau d'entiers de taille 5, on peut
  utiliser l'instruction
    ☐ int toto[taille=5]:
```

```
\square int toto[5];
\square int tab[] = 5;
☐ int[] new tableau(5):
□ char tableau[5];
```

3. Vous utilisez une boucle while quand:

□ vous n'avez pas déclaré de fonction

_			- F					_		
	vous	avez	déjà	${\rm fait}$	un	for	${\rm dans}$	le	${\bf m\hat{e}me}$	pro
	oram	me ni	rincin	<u>-1</u>						

☐ l'incrément de la variable de boucle n'est pas 1

_								P	_
	vous r	ne con	naissez	pas	le	nombre	d'itér	ations	$d\epsilon$
	la bou	ıcle à	l'avanc	e					

4. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit :

□ void afficher\_date(struct date\_s d);

☐ struct date_s afficher_date(struct date_s	d);
☐ int afficher_date(date_s d);	
□ void afficher_date(date_s d);	

```
5. Soient deux variables entières x et y initialisées à 4 et
  5 respectivement. L'affichage x=4 et y=5 est obtenu
  avec la commande :
    \square printf("x=%x et y=%y\n");
```

```
\square printf("x=%d et y=%d\n",x y);
    \square printf("x=%d et y=%d\n",x,y);
    \square printf("x=%d et y=%d\n,x,y");
6. Une variable booléenne est un variable :
    □ jamais nulle
    □ NaN (not a number, qui n'est pas un nombre)
    □ qui est vraie ou fausse
    □ à laquelle une valeur vient d'être affectée
    ☐ réelle positive
```

7. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :

```
\Box int factorielle(int x);
☐ int factorielle():
☐ int factorielle(double n);
□ struct int factorielle(int n);
```

8. Pour compiler un programme prog.c, on utilise la ligne de commande :

```
☐ gcc prog.exe -Wall -o prog.c
☐ gcc -Wall prog.exe -o prog.c
☐ gcc prog.c -o -Wall prog.exe
☐ gcc -Wall prog.c -o prog.exe
```

9. Le code suivant :

```
int i;
for (i = 1; i < 5; i = i + 1)
{
     printf("%d ", i);
printf("\n");
affichera:
 \square 1 2 3 4
```

 $\Box \ 4\ 3\ 2\ 1\ 0$  $\Box 01234$ 

```
\square 4 3 2 1
```

```
10. Si cette erreur apparaît à la compilation :
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft » que doit-
   on chercher dans le programme?
     □ une directive préprocesseur #include manquante
     \square une faute de frappe dans un appel de fonction
     □ une variable non déclarée
     □ un caractère interdit en C
```

11. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit:

```
☐ int saisie_utilisateur();
☐ saisie_utilisateur(scanf(%d));
□ void saisie_utilisateur(char c);
□ void saisie_utilisateur(int n);
```

12. Pour l'extrait de programme suivant :

```
int produit = 0;
  int serie[4] = \{2, 2, 2, 2\};
  for (i = 0; i < 4; i = i + 1)
    produit = produit * serie[i];
  printf("produit = %d", produit);
La valeur affichée est :
```

 $\Box$  16

4
0

 $\square$  8

13. Le code suivant :

 $\Box 01234$ 

 $\square$  4 3 2 1

```
int i;
 for (i = 4; i \ge 0; i = i - 1)
     printf("%d ", i);
 printf("\n");
affichera:
 \Box \ 4\ 3\ 2\ 1\ 0
 \Box 1234
```

14. Pour afficher à l'aide de printf("%d\n",tab[i]);	17. Un fichier source est:	□ char 'c';
le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :	☐ un fichier texte qui sera traduit en instructions processeur	$\square$ int char;
☐ for(i=0;i<=5;i=i+1)	□ un fichier que l'ont doit citer dans les documents	20. Soit la fonction f définie par :
$\square for(i=0;i<5;i=i+1)$	produits sur l'ordinateur	int f(int a)
☐ for(i=1;i<=5;i=i+1)	□ un document qui doit être protégé	{
☐ for(i=1;i<5;i=i+1)	□ un document de référence du système	printf("a = \n", %d);
15. Un programme en langage C doit comporter une et une seule définition de la fonction :	$\hfill\Box$ un document illisible pour les humains	if (a > 0) {
$\square$ main	18. L'écriture <u>111</u> en binaire correspond au nombre natu-	return f(a - 1) + 1;
$\Box$ init	rel:	}
$\square$ include		return 4;
$\square$ begin	□ 7	J
16. L'écriture 101 en binaire correspond au nombre natu-	□ 8	Alors l'expression f(1) prendra la valeur :
rel :	□ 111	$\Box$ 0
□ 101	19. Laquelle de ces écritures correspond à la déclaration	□ 1
$\square$ 3	d'une variable de type caractère en langage C?	
$\Box$ 4	□ char c;	□ 5
$\Box$ 5	☐ char "c";	$\Box$ 4

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un fichier source est: □ un fichier texte qui sera traduit en instructions processeur □ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un document qui doit être protégé ☐ un document illisible pour les humains  $\square$  un document de référence du système 2. Pour l'extrait de programme suivant : int somme = 0; int  $serie[4] = \{2, 4, 10, 4\};$ for (i = 0; i < 4; i = i + 1)somme = somme + serie[i]: printf("somme = %d",somme); La valeur de somme affichée est :  $\square$  16  $\square$  3  $\square$  20  $\Box$  6 3. Une variable booléenne est un variable :  $\square$  qui est vraie ou fausse □ à laquelle une valeur vient d'être affectée □ NaN (not a number, qui n'est pas un nombre) ☐ réelle positive ☐ jamais nulle 4. Soit le programme principal suivant : int main() { int a = 3; int b = 5; printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b)$ ;

return EXIT\_SUCCESS;

```
appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
5. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit :
    □ struct date_s afficher_date(struct date_s d);
    ☐ int afficher_date(date_s d);
    □ void afficher_date(struct date_s d);
    □ void afficher_date(date_s d);
6. L'écriture 111 en binaire correspond au nombre natu-
  rel:
    \square 3
    \Box 7
    \square 8
     \square 111
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
    □ char c;
    ☐ int char:
    □ char 'c';
    ☐ char "c":
8. L'écriture 101 en binaire correspond au nombre natu-
  rel:
    \Box 4
    \square 3
    \square 101
     \Box 5
```

```
9. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
     \square #define taille = 3
     ☐ #define taille = N
     \square #define N = 3
     □ #define N 3
10. Vous utilisez une boucle while quand:
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
      □ vous avez déjà fait un for dans le même pro-
        gramme principal
      \square l'incrément de la variable de boucle n'est pas 1
      □ vous n'avez pas déclaré de fonction
11. Si cette erreur apparaît à la compilation :
   Undefined symbols :"_prinft" ou
   référence indéfinie vers « prinft » que doit-
   on chercher dans le programme?
     \square une faute de frappe dans un appel de fonction
     □ un caractère interdit en C
     □ une variable non déclarée
      □ une directive préprocesseur #include manquante
12. Sur un ordinateur avec un seul processeur, habituelle-
    ment les processus sont exécutés :
      \square tous ensemble
     □ chacun son tour, après que le processus précédent
        a terminé
      □ tour à tour, un petit peu à chaque fois
      □ en parallèle, chacun dans un registre
13. Après exécution du programme :
       lecture 8 r0
       valeur 3 r1
       mult r1 r0
       valeur 1 r2
       add r2 r0
       ecriture r0 8
       stop
       5
```

```
\square la case mémoire 8 contiendra 0
      □ la case mémoire 8 contiendra 16
      □ le terminal affiche 8
      \square le bus explose
14. Sur unix (ou linux), la commande mkdir permet de :
      □ créer un fichier texte
      □ créer un répertoire
      \Box changer de répertoire courant
      \Box ouvrir un fichier texte
15. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
         return 3;
      return 4;
    }
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 4
      \Box 0
```

```
16. Si pgcd est une fonction prenant en entrée deux entiers
                                                               18. Si carre est une fonction prenant en entrée un en-
    et renvoyant un entier, il est correct d'écrire :
                                                                   tier et renvoyant le carré de cet entier, et que n est
                                                                   une variable entière définie et initialisée, il est correct
      \square int n = pgcd();
                                                                   d'écrire :
      \Box n = pgcd(int p, int q);
                                                                     \square n = carre(int n);
      \square int pgcd(2);
                                                                     \square int carre(2);
      \square n = pgcd(n, 3);
                                                                     \square n = carre(n);
17. Pour l'extrait de programme suivant :
                                                                     \square int n = carre();
     int i = 0;
                                                               19. Le type des réels en C est :
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
                                                                     \square real
                                                                     \square int
          for (j = 0; j < 2; j = j + 1)
                                                                     \square double
               printf("%d ", i);
                                                                     \square char
          }
                                                               20. Pour déclarer un tableau d'entiers de taille 5, on peut
                                                                   utiliser l'instruction
     printf("\n");
                                                                     \square int[] new tableau(5);
    qu'est ce qui sera affiché?
                                                                     ☐ int toto[taille=5];
      \Box 1 2 3 1 2
                                                                     \Box int tab[] = 5;
      \Box 0 1 0 1 0 1
                                                                     \square int toto[5];
      \Box 0 0 1 1 2 2
      \Box 0 1 2 0 1 2
                                                                     □ char tableau[5];
```

## Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
     ☐ double afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu(int char);
    □ void afficher_menu();
     ☐ int afficher menu():
2. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
    printf("\n");
  affichera:
    \Box 0123456
    \Box 0 1 2 3 4 5 6 7
    \Box 02468
     \Box 0246
3. Si pgcd est une fonction prenant en entrée deux entiers
  et renvoyant un entier, il est correct d'écrire :
     \square n = pgcd(int p, int q);
    \square n = pgcd(n, 3);
    \square int pgcd(2);
    \square int n = pgcd();
4. Le code suivant :
    int age = 18;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
        printf("Majeur\n");
    }
```

```
affichera:
     □ Mineur
     □ Mineur
        Majeur
     \square rien
     □ Majeur
5. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
        return 3;
     return 4:
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \square 0
6. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     \square analyse harmonique
     \square analyse lexicale
     □ analyse sémantique
     \square analyse syntaxique
7. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut lancer un déboggueur
     □ qu'on veut changer alétoirement de fond d'écran
     ☐ que l'on veut voir tous les avertissements
     □ qu'il faut indenter le fichier source
8. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     \Box char c;
     ☐ int char;
     ☐ char "c";
     □ char 'c';
```

```
9. Si a et b sont deux variables de type:
   struct toto_s
      int n;
      double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \square (a.n == b.n) & (a.x == b.x)
     \square a{n, x} == b{n, x}
     □ a == b
     \Box a = b
10. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     \square une accolade en trop
     ☐ un point-virgule manquant
     ☐ une accolade manquante
     □ un point-virgule en trop
11. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 8
     \square 7
     □ 111
     \square 3
12. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
    printf("\n");
   Alors l'affichage sera:
     ☐ ABCDEF
     \Box i
     □ A
```

13. Si cet avertissement apparaît à la compilation :	16. Soit la fonction f définie par :	$\square$ ses champs
warning: implicit declaration of function 'max, que doit-on chercher dans le programme?	int f(int a)	□ ses blocs
☐ une directive préprocesseur #include manquante	printf("a = \n", %d);	☐ ses cases
☐ une fonction déclarée mais non définie	if (a > 0)	19. Le code suivant :
☐ une fonction appelée avant sa déclaration	t return f(a - 1) + 1;	int i;
☐ un désaccord entre la déclaration et la définition d'une fonction	} return 4; }	<pre>for (i = 0; i &lt; 5; i = i + 1) {     printf("%d ", i);</pre>
14. Les lignes	Alors l'expression f(1) prendra la valeur :	<pre>} printf("\n");</pre>
<pre>int i; int x=0;</pre>		affichera:
for(i=0,i<5,i=i+1)	□ 1 □ 0	$\Box 01234$
t x=x+1;	□ 5	
}	17. Le code suivant :	$\square$ 4 3 2 1
<ul> <li>□ comportent une erreur qui ne sera pas détectée</li> <li>□ comportent une erreur qui sera détectée au cours de l'analyse syntaxique</li> <li>□ ne comportent aucune erreur</li> <li>□ comportent une erreur qui sera détectée au cours de l'édition de lien</li> <li>15. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :</li> <li>□ #include <stdio.h></stdio.h></li> <li>□ #include <studlib.h></studlib.h></li> <li>□ #include <studlib.h></studlib.h></li> </ul>	<pre>int i; for (i = 4; i &gt; 0; i = i - 1) {         printf("%d ", i); } printf("\n"); affichera:</pre>	20. Après exécution jusqu'à la ligne 15 du programme C:  10 int main() { 11 int x = 5; 12 int y = 3; 13 14 x = y; 15 16 17 }  □ la variable y vaut 5 □ la variable x vaut 5 et la variable y vaut 3 □ la programme affiche "Faux"
☐ #appart <stdlib.h></stdlib.h>	$\square$ ses chants	□ le programme affiche "Faux"

## Éléments d'informatique - contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 4
     \Box 0
     \Box 5
     \Box 1
2. Soit la fonction g définie par :
   int g(int a)
   ₹
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     }
     return 7:
   Alors l'expression g(0) prendra la valeur :
     \Box 5
     \Box 0
     \Box 7
3. Pour l'extrait de programme suivant :
    int i;
    int j;
    for(i=4;i>0;i=i-1)
      for(j=i;j<6;j=j+1)
        printf("*");
```

```
printf(" ");
   }
   qu'est ce qui sera affiché?
      ***** *** ***
4. Sous unix (ou linux), la commande 1s permet de :
     \square afficher le contenu d'un fichier texte
    □ compiler un programme
    □ voir des clips musicaux
    \square afficher la liste de fichiers contenus dans un
       répertoire
5. Le code suivant :
    int i;
   for (i = 4; i \ge 0; i = i - 1)
        printf("%d ", i);
   printf("\n");
   affichera:
    \Box 01234
    \Box 1 2 3 4
    \square 4 3 2 1
    \Box 4 3 2 1 0
6. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
    □ un débogueur
    □ scanf("%d", &n);
    ☐ printf("Valeur de n ? %d\n", n);
    ☐ printf("Valeur de n ? %g\n", n);
7. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
    ☐ printf("%d", factorielle(n));
    \square n = factorielle(p, q);
    ☐ int factorielle(int 2);
     \square n = factorielle();
```

```
8. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \Box 1
     \square 0
     \square 1.5
9. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     □ void afficher_date(struct date_s d);
     □ struct date_s afficher_date(struct date_s d);
     ☐ int afficher_date(date_s d);
     □ void afficher_date(date_s d);
10. Vous utilisez une boucle while quand :
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     ☐ l'incrément de la variable de boucle n'est pas 1
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
11. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     ☐ char "c";
     □ char 'c';
     ☐ int char;
     □ char c;
12. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 6
     \square 20
     \Box 16
     \square 3
```

13. Le bus système sert à :	☐ le programme affiche "Faux"	18. Un enregistrement permet de grouper plusieurs valeurs
$\hfill\Box$ Arriver à l'heure en cours	$\Box$ la variable x va ut 5 et la variable y vaut 3	dans:
$\hfill \square$ Transférer des données et intructions entre pro-	$\Box$ la variable x vaut 3	□ ses champs
cesseur et mémoire	☐ la variable y vaut 5	□ ses chants
☐ Écrire des données sur le dique dur	16. Le code suivant :	ses cases
☐ transporter les processus du tourniquet au processeur	int i;	□ ses blocs
14. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max	<pre>for (i = 4; i &gt; 0; i = i - 1) {     printf("%d ", i);</pre>	19. Le langage C est un langage
, que doit-on chercher dans le programme?	}	$\Box$ lu, écrit, parlé
$\Box$ une fonction appelée avant sa déclaration	<pre>printf("\n");</pre>	$\square$ composé
$\hfill \square$ une fonction déclarée mais non définie	affichera:	□ compilé
$\Box$ une directive préprocesseur #include manquante	$\Box \ 0\ 1\ 2\ 3$	□ interprété
$\Box$ un désaccord entre la déclaration et la définition	$\Box \ 0\ 1\ 2\ 3\ 4$	
d'une fonction	$\Box \ 4\ 3\ 2\ 1\ 0$	20. Un fichier source est:
15. Après exécution jusqu'à la ligne 15 du programme C :	$\square \ 4\ 3\ 2\ 1$	$\hfill \square$ un document de référence du système
10 int main() { 11 int x = 5;	17. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction	$\Box$ un fichier texte qui sera traduit en instructions processeur
12 int y = 3; 13 14 x = y;	☐ int toto[5]; ☐ int toto[taille=5]; ☐ toto[7]	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
15	<pre>□ int[] new tableau(5);</pre> □ int tab[] = 5;	$\Box$ un document qui doit être protégé
16 17 }	☐ char tableau[5];	$\hfill\Box$ un document illisible pour les humains

 $\square$  7

# Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
        }
    }
  qu'est ce qui sera affiché?
    \Box 0 1 2 0 1 2
     \Box 0 0 0 1 1 1
     □ 1 2 1 2 3
     \Box 0 1 0 1 0 1 0 1
2. Si pgcd est une fonction prenant en entrée deux entiers
  et renvoyant un entier, il est correct d'écrire :
     \square n = pgcd(n, 3);
     \square int pgcd(2);
     \square n = pgcd(int p, int q);
    \square int n = pgcd();
3. Soit la fonction g définie par :
  int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
     \Box 5
     \Box 0
```

```
4. Sous unix (ou linux), la commande cd permet de :
     □ changer de répertoire courant
    □ ouvir un bureau partagé (common desktop)
    ☐ récupérer un programme arrêté avec la commande
       ab
    □ détruire un fichier
     \square jouer de la musique
5. Si le code :
   struct toto_s
     int n;
     double x;
  };
   précède la fonction main(), alors on peut écrire en
  début de main() :
    \square int toto.n = 3:
    \Box toto_s struct z = {3, 0.5};
    □ struct toto_s toto;
    \square toto_s n, x;
    \Box int struct toto_s = {3, -1e10};
6. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
    \square printf("x=%d et y=%d\n",x,y);
    \square printf("x=%d et y=%d\n",x y);
    \square printf("x=%x et y=%y\n");
    \square printf("x=%d et y=%d\n,x,y");
7. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
  est l'ordre :
    \square un ordre quelconque
    □ dans lequel vous avez déclaré ces fonction
    □ alphabétique
     □ dans lequel ces fonctions sont appelées dans le
       main
```

```
8. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
          printf("%d ", i);
     printf("\n");
    affichera:
     \Box 8 6 4 2
     \square 8 2
     \Box 02468
     \Box 8 6 4 2 0
9. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
   }
   Alors l'expression f(1) prendra la valeur :
      \Box 0
     \Box 4
      \Box 1
      \Box 5
10. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
     \square n = factorielle();
     ☐ printf("%d", factorielle(n));
     \square n = factorielle(p, q);
     ☐ int factorielle(int 2);
11. Quels calculs peut-on programmer en programmation
   structurée?
      ☐ il v a des calculs programmables en langage ma-
        chine et qui ne sont pas programmables en pro-
```

grammation structurée

□ en programmation structurée on peut programmer tous les calculs programmables en langage machine □ certains programmes sont de vrais plats de spaghetti □ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine  12. Pour compiler un programme prog.c, on utilise la ligne de commande: □ gcc -Wall prog.exe -o prog.exe □ gcc prog.c -o -Wall prog.exe □ gcc prog.exe -Wall -o prog.c  13. Le code suivant: int i; for (i = 4; i > 0; i = i - 1) {     printf("\n"); affichera: □ 0 1 2 3 4 □ 4 3 2 1 □ 4 3 2 1 0 □ 0 1 2 3  14. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire:	for (i = 0; i < 2; i = i + 1) {  for (j = 0; j < 3; j = j + 1) {  printf("%d ", j); }  qu'est ce qui sera affiché?  0 1 2 3 0 1 2  0 1 2 0 1 2 3  0 1 2 0 1 2  10 0 1 1 2 2 3  16. Si cette erreur apparaît à la compilation: error: expected ';' before '}' token que doiton chercher dans le programme?  un point-virgule manquant une accolade manquant une accolade en trop un point-virgule en trop  17. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt:  scanf("%d", &n); printf("Valeur de n ? %d\n", n);	18. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :  □ struct date_s afficher_date(struct date_s □ void afficher_date(struct date_s d); □ void afficher_date(date_s d); □ int afficher_date(date_s d);  19. Si a et b sont deux variables de type :  struct toto_s {     int n;     double x;     };  Alors pour tester l'égalité de a et de b on utilise la condition :  □ (a.n == b.n) && (a.x == b.x) □ a = b □ a{n, x} == b{n, x} □ a == b  20. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses cases □ ses champs □ ses blocs □ ses chants
--	--	---

d);

# Éléments d'informatique – contrôle continue

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N° INE}: \end{array}$ 

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
1. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
  Alors l'affichage sera:
     Πi
     \square A
     ☐ ABCDEF
     □ cccccc
2. Laquelle de ces écritures correspond à la déclaration
  d'une variable de type caractère en langage C?
     ☐ int char;
     \Box char c;
     ☐ char "c";
     □ char 'c':
3. L'écriture 101 en binaire correspond au nombre natu-
  rel:
     \Box 5
     \square 3
     \square 101
     \Box 4
4. Soient deux variables entières x et y initialisées à 4 et
  5 respectivement. L'affichage x=4 et y=5 est obtenu
  avec la commande :
     \square printf("x=%x et y=%y\n");
    \square printf("x=%d et y=%d\n",x y);
    \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n,x,y");
5. Pour déclarer une variable qui sera utilisée comme va-
  riable de boucle on peut utiliser l'instruction
     □ loop i;
     \square int k;
     \square int loop n;
     \square int %d;
```

```
6. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \square 8
     \Box 4
     \square 16
     \Box 0
7. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il comporte une boucle infinie
     □ il n'affiche rien
     □ il risque d'afficher bonjour à la place de coucou
    \square il ne compile pas
8. Un registre du processeur est :
     □ une gamme de fréquence de fonctionnement du
        processeur
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     \square un composant qui contient la liste des fichiers du
        système
     □ une unité de calcul spécialisée de l'ordinateur
9. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
```

return 3:

```
}
      return 4;
    Alors l'expression f(0) prendra la valeur :
     \Box 0
     \Box 4
     \square 3
10. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
      \Box 6
     \Box 10
     \square 15
     \Box 0
11. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \Box 7
     \square 8
     \square 3
      □ 111
12. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
     printf("\n");
    affichera:
     \Box 01234567
     \square 0 2 4 6
     \Box 0123456
      \Box 02468
```

```
13. Si a et b sont deux variables de type :
                                                            15. Soit la fonction g définie par :
                                                                                                                               □ avoir défini une constante symbolique de la taille
                                                                                                                                  de cette fonction
    struct toto_s
                                                                 int g(int a)
                                                                                                                               ☐ l'avoir définie
    {
      int n;
                                                                   printf("a = \n", %d);
                                                                                                                               □ l'avoir déclarée et définie
      double x;
                                                                   if (1 > 0)
                                                                                                                         18. Si n est une variable entière pour demander sa valeur
    };
                                                                   {
                                                                                                                             à l'utilisateur, on utilise plutôt :
                                                                     return 5;
    Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                               □ un débogueur
    condition:
                                                                   return 7;
                                                                                                                               ☐ printf("Valeur de n ? %g\n", n);
      □ a == b
                                                                                                                               □ scanf("%d", &n);
     \square a{n, x} == b{n, x}
                                                                Alors l'expression g(0) prendra la valeur :
                                                                                                                               ☐ printf("Valeur de n ? %d\n", n);
     \square (a.n == b.n) && (a.x == b.x)
                                                                  \square 0
                                                                                                                         19. Dans la commande gcc, l'option -Wall signifie :
      \Box a = b
                                                                  \Box 5
                                                                                                                               □ qu'il faut lancer un déboggueur
14. Le code suivant :
                                                                   \square 7
                                                                                                                               ☐ que l'on veut voir tous les avertissements
     int i;
     for (i = 4; i > 0; i = i - 1)
                                                            16. Lorsqu'un programme utilise printf ou scanf il faut
                                                                                                                               \Box qu'on veut changer alétoirement de fond d'écran
                                                                qu'il contienne l'instruction préprocesseur :
                                                                                                                               □ qu'il faut indenter le fichier source
         printf("%d ", i);
                                                                   ☐ #include <studlib.h>
                                                                                                                         20. Vous utilisez une boucle while quand:
                                                                  ☐ #appart <stdlib.h>
     printf("\n");
                                                                                                                               \square vous ne connaissez pas le nombre d'itérations de
                                                                  ☐ #include <studio.h>
                                                                                                                                  la boucle à l'avance
    affichera:
                                                                                                                               □ vous avez déjà fait un for dans le même pro-
      \Box 01234
                                                                   ☐ #include <stdio.h>
                                                                                                                                  gramme principal
      \square 4 3 2 1
                                                            17. Avant de faire appel à une fonction il est nécessaire
                                                                                                                               ☐ l'incrément de la variable de boucle n'est pas 1
      \square 0 1 2 3
                                                                de:
                                                                                                                               \square vous n'avez pas déclaré de fonction
      \Box 4 3 2 1 0
                                                                   □ l'avoir déclarée
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Un enregistrement permet de grouper plusieurs valeurs dans:  $\square$  ses chants  $\square$  ses blocs  $\square$  ses champs  $\square$  ses cases 2. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return f(a - 1) + 1; return 4; Alors l'expression f(1) prendra la valeur :  $\Box$  0  $\Box$  5  $\Box$  4  $\Box$  1 3. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  n = carre(n);  $\square$  n = carre(int n);  $\Box$  int carre(2);  $\square$  int n = carre(); 4. Un bit est: □ un battement d'horloge processeur □ la longueur d'un mot mémoire  $\square$  un chiffre binaire (0 ou 1) ☐ l'instruction qui met fin à un programme

```
5. Après exécution jusqu'à la ligne 15 du programme C:
10
11
      int main() {
12
           int x = 5;
13
14
           x = 3 * x + 1:
15
16
    }
17
    \square la variable x vaut -\frac{1}{2}
     ☐ le programme affiche ****
     □ la variable x vaut 16
     \square le programme affiche x
6. Au début de la fonction main() on place le code :
    char i:
   for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
    printf("\n");
  Alors l'affichage sera:
     ☐ ABCDEF
     \square A
     \Box ccccc
     \Box i
7. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il risque d'afficher bonjour à la place de coucou
     □ il n'affiche rien
     \square il comporte une boucle infinie
     \square il ne compile pas
```

```
8. Le code suivant :
     int i:
     for (i = 8; i > 0; i = i - 2)
          printf("%d ", i);
     printf("\n");
    affichera:
     \Box 02468
     \square 8 6 4 2
     \square 8 2
     \Box 8 6 4 2 0
 9. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(racine(x)*racine(x));
     \square x = racine(x * x) - racine(x);
     \square x = racine(2/3):
     \square x - 1 = racine(x);
10. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
              printf("%d ", j);
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 3 0 1 2
```

 $\Box$  0 0 1 1 2 2 3

11. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	<pre></pre>	17. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :
☐ char c;		☐ int afficher_date(date_s d);
☐ int char; ☐ char 'c';	_ **** *** ***	□ void afficher_date(date_s d);
□ char "c";	45. 01	□ struct date_s afficher_date(struct date_s
12. Pour déclarer une variable qui sera utilisée comme va-	15. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou	□ void afficher_date(struct date_s d);
riable de boucle on peut utiliser l'instruction	référence indéfinie vers « prinft » que doit-	18. Avant de faire appel à une fonction il est nécessaire
☐ int %d;	on chercher dans le programme?	de:
□ loop i;	$\Box$ un caractère interdit en C	☐ l'avoir déclarée et définie
□ int k;	$\square$ une variable non déclarée	☐ l'avoir déclarée
☐ int loop n;	$\hfill\Box$ une faute de frappe dans un appel de fonction	□ avoir défini une constante symbolique de la taille de cette fonction
13. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-	☐ une directive préprocesseur #include manquante	☐ l'avoir définie
on chercher dans le programme?  □ une accolade manquante	16. Soit un programme contenant les lignes suivantes :	19. Vous avez déclaré préalablement un ensemble de fonc-
☐ un point-virgule manquant	int i = 0;	tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions
☐ un point-virgule en trop	int j = 0;	est l'ordre :
□ une accolade en trop	for (i = 0; i < 2; i = i + 1) {	$\square$ alphabétique
14. Pour l'extrait de programme suivant :	for $(j = 0; j < 3; j = j + 1)$	$\hfill\Box$ un ordre quel conque
int i;	{	$\Box$ dans lequel vous avez déclaré ces fonction
int j;	printf("%d ", i);	$\Box$ dans lequel ces fonctions sont appelées dans le
for(i=4;i>0;i=i-1)	}	main
{ for(j=i;j<6;j=j+1)		20. La virtualisation de la mémoire permet notamment de
{	qu'est ce qui sera affiché?	stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
<pre>printf("*"); }</pre>	□ 1 2 1 2 3	□ les fichiers du disque
<pre>printf(" ");</pre>	$\Box$ 0 1 0 1 0 1 0 1	□ certaines données de la mémoire de travail
}	$\Box$ 0 1 2 0 1 2	□ en temps d'accès
qu'est ce qui sera affiché?	□ 0 0 0 1 1 1	☐ des processus
4	·	

d);

1. Si le code :

struct toto s

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
int n:
     double x;
  };
  précède la fonction main(), alors on peut écrire en
  début de main() :
    □ struct toto_s toto;
    \square int struct toto_s = {3, -1e10};
    \square toto_s struct z = {3, 0.5};
    \square int toto.n = 3;
    \square toto_s n, x;
2. Vous utilisez une boucle while quand:
    □ vous n'avez pas déclaré de fonction
    □ vous avez déjà fait un for dans le même pro-
       gramme principal
    ☐ l'incrément de la variable de boucle n'est pas 1
    \square vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
3. Soit le programme principal suivant :
  int main()
   int a = 3:
    int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
```

```
\Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
4. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre :
    \square un ordre quelconque
     □ alphabétique
     □ dans lequel vous avez déclaré ces fonction
     □ dans lequel ces fonctions sont appelées dans le
        main
5. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses chants
     \square ses blocs
     □ ses cases
     \square ses champs
6. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \Box 1
    \square 1.5
     \Box 0
7. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.c -o -Wall prog.exe
8. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \square int pgcd(int x, int x);
     \square void pgcd(int x, int y);
     \square int pgcd(int x, y);
    \square int pgcd(int y, int x);
```

```
9. Au début de la fonction main() on place le code :
     char i;
     for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
    printf("\n");
    Alors l'affichage sera:
     \square A
     ☐ ABCDEF
      □ ccccc
     Πi
10. Quelle étape de la compilation vient d'échouer lors-
    qu'on a un message comme celui-ci :
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
      □ l'analyse sémantique
     ☐ l'analyse harmonique
     ☐ l'analyse des entrées clavier
     □ l'édition de liens
11. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square void exposant(double x^n);
     \square double exposant(double x, int n);
      \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
12. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n",x y);
      \square printf("x=%d et y=%d\n,x,y");
     \square printf("x=%d et y=%d\n",x,y);
13. Un bit est:
      ☐ l'instruction qui met fin à un programme
     \square la longueur d'un mot mémoire
      □ un battement d'horloge processeur
```

 $\square$  un chiffre binaire (0 ou 1)

(	dans le répertoire courant on peut utiliser la commande :
	□ yppasswd
	□ mkdir TP4
	□ new TP4
	☐ kwrite TP4
5. \$	Soit un programme contenant les lignes suivantes :
	<pre>int i = 0; int j = 0; for (i = 0; i &lt; 2; i = i + 1) {     for (j = 0; j &lt; 3; j = j + 1)     {         printf("%d ", i);     } }</pre>
(	qu'est ce qui sera affiché?
	□ 0 1 2 0 1 2
	□ 0 1 0 1 0 1 0 1
	□ 1 2 1 2 3
	□ 0 0 0 1 1 1

```
16. Pour l'extrait de programme suivant :
                                                                   \Box un composant qui contient la liste des fichiers du
                                                                      système
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
                                                             19. Pour l'extrait de programme suivant :
      for (i = 0; i < 4; i = i + 1)
                                                                  int i = 0;
                                                                  int j = 0;
         somme = somme + serie[i];
                                                                  for (i = 0; i < 3; i = i + 1)
      printf("somme = %d",somme);
                                                                       for (j = 0; j < 2; j = j + 1)
    La valeur de somme affichée est :
      \square 3
                                                                            printf("%d ", i);
     \Box 6
                                                                       }
                                                                  }
      \Box 16
                                                                  printf("\n");
      \square 20
                                                                 qu'est ce qui sera affiché?
17. Une variable booléenne est un variable :
     □ jamais nulle
                                                                   \Box 0 1 0 1 0 1
     \Box à la
quelle une valeur vient d'être affectée
                                                                   \Box 0 0 1 1 2 2
     \square qui est vraie ou fausse
                                                                   \Box 0 1 2 0 1 2
     □ réelle positive
                                                                   \Box 1 2 3 1 2
     \square NaN (not a number, qui n'est pas un nombre)
                                                             20. Dans la commande gcc, l'option -Wall signifie :
18. Un registre du processeur est :
                                                                   \square qu'il faut lancer un déboggueur
     □ une gamme de fréquence de fonctionnement du
                                                                   \Box qu'on veut changer alétoirement de fond d'écran
      \square une unité de calcul spécialisée de l'ordinateur
                                                                   □ qu'il faut indenter le fichier source
      □ une case mémoire interne au processeur qui sera
                                                                   ☐ que l'on veut voir tous les avertissements
         manipulée directement lors des calculs
```

## Éléments d'informatique – contrôle continue

 $\square$  0

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N}^{\circ} \text{ INE}: \end{array}$ 

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction g définie par :
   int g(int a)
   {
     printf("a = \n", %d);
     if (1 > 0)
        return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 5
     \square 7
     \Box 0
2. Le langage C est un langage
     □ interprété
     □ compilé
     □ composé
     □ lu, écrit, parlé
3. Après exécution jusqu'à la ligne 15 du programme C:
 10
      int main() {
 11
 12
           int x = 5;
 13
 14
           x = 3 * x + 1;
 15
 16
 17
      }
     \square le programme affiche x
     ☐ le programme affiche ****
     \square la variable x vaut 16
     \square la variable x vaut -\frac{1}{2}
```

```
4. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=1;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=0;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
5. Si a et b sont deux variables de type:
   struct toto s
   {
     int n;
     double x;
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \square (a.n == b.n) && (a.x == b.x)
     □ a == b
     \square a = b
     \square a{n, x} == b{n, x}
6. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
     \square que l'on veut voir tous les avertissements
     \square qu'on veut changer alétoirement de fond d'écran
7. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 5
     \Box 1
     \Box 4
```

```
8. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
         return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 4
      \Box 0
 9. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
    une variable entière définie et initialisée, il est correct
    d'écrire :
      \square int carre(2);
      \square int n = carre();
      \square n = carre(int n);
      \square n = carre(n);
10. Au début de la fonction main() on place le code :
     char i;
     for (i = 'A'; i \le 'F'; i = i + 1)
        printf("%c", i);
     printf("\n");
    Alors l'affichage sera:
      ☐ ABCDEF
      \Box A
      Πi
      \Box ccccc
11. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée
      □ l'avoir déclarée et définie
      \square avoir défini une constante symbolique de la taille
         de cette fonction
```

□ l'avoir définie

```
12. Après exécution jusqu'à la ligne 15 du programme C:
                                                             15. Pour déclarer une fonction factorielle qui prend en
                                                                 argument un entier et renvoie sa factorielle on écrit :
 10
       int main() {
                                                                                                                              int f(int a, int b)
                                                                   \square int factorielle(int x);
 11
            int x = 5;
                                                                   □ struct int factorielle(int n);
                                                                                                                                a = a + b;
 12
            int y = 3;
 13
                                                                   ☐ int factorielle(double n);
                                                                                                                                return a;
 14
            x = y;
                                                                   ☐ int factorielle();
 15
                                                             16. On considère deux variables booléennes A et B initia-
  16
            . . .
                                                                 lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                                                                                \Box f(a,b)=8, a=3, b=5
 17
      }
                                                                 pressions booléennes suivantes, laquelle a pour valeur
                                                                                                                               \Box f(a,b)=13, a=8, b=5
                                                                 TRUE?
     \Boxla variable y vaut 5
                                                                                                                               \Box f(3,5)=8, a=3, b=5
                                                                   □ A && B
     \square la variable x vaut 3
                                                                                                                                \Box f(a,b)=8, a=8, b=5
                                                                   \square (!A || B)
     ☐ le programme affiche "Faux"
                                                                   \square (A == TRUE) && (B == TRUE)
                                                                                                                          19. Un fichier source est:
     \square la variable x vaut 5 et la variable y vaut 3
                                                                   \square !(!A || B) == (A && !B)
                                                             17. Après exécution jusqu'à la ligne 15 du programme C:
13. Le type des réels en C est :
                                                                    int main() {
                                                               10
      \square real
                                                                         int x = 5;
                                                              11
     □ char
                                                               12
                                                                         int y;
     ☐ double
                                                               13
                                                                                                                                   processeur
                                                               14
                                                                         y = x;
      \square int
                                                              15
14. Pour l'extrait de programme suivant :
                                                               16
                                                              17
                                                                   }
      int somme = 0;
                                                                                                                                int somme = 0;
                                                                   \square le programme affiche "Faux"
      for (i = 0; i < 5; i = i + 1)
                                                                  \square la variable y vaut 5
      {
        somme = somme + i;
                                                                   \Box la variable x vaut 0
        i = i + 1; /* attention ! */
                                                                   \Boxla variable x vaut 5 et la variable y vaut 0
                                                             18. Soit le programme principal suivant :
      printf("somme = %d",somme);
                                                                 int main()
   La valeur de somme affichée est :
     \Box 10
                                                                                                                                \Box 6
                                                                  int a = 3;
                                                                  int b = 5;
     \Box 6
                                                                                                                                \square 3
                                                                  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     \square 15
                                                                                                                                \Box 16
                                                                  return EXIT_SUCCESS;
     \Box 0
                                                                                                                                \square 20
                                                                 }
```

```
appelant la fonction f ainsi définie :
   L'affichage dans le main est le suivant :
      ☐ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
      □ un document qui doit être protégé
      \square un document de référence du système
      □ un fichier texte qui sera traduit en instructions
      ☐ un document illisible pour les humains
20. Pour l'extrait de programme suivant :
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
```

т		-1
	acence	- 1

## Éléments d'informatique – contrôle continue

Prónom ·	Nom ·
Prenom:	Nom .
NO INIT .	
IN IINE:	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
   int b = 20;
   int n:
   scanf("%d", &n);
   while(cond)
      scanf("%d", &n);
  Quelle est la condition cond :
    \square (a<n) || (n>b)
    □ a<=n<=b
    \square (a<=n) && (n<=b)
    \square (n<=a) && (n<=b)
2. Pour l'extrait de programme suivant :
   int i = 0;
   int j = 0;
   for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
            printf("%d ", j);
        }
  qu'est ce qui sera affiché?
    \Box 0 0 1 1 2 2 3
    \Box 0 1 2 3 0 1 2
    \Box 0 1 2 0 1 2
    \Box 0 1 2 0 1 2 3
```

3. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :

analyse	syntaxique
analyse	lexicale
analyse	sémantique
analyse	harmonique

4. Dans la commande gcc, l'option -Wall signifie : □ qu'il faut lancer un déboggueur □ qu'on veut changer alétoirement de fond d'écran □ qu'il faut indenter le fichier source ☐ que l'on veut voir tous les avertissements 5. Le code suivant : int i: for (i = 8: i > 0: i = i - 2)printf("%d ", i); printf("\n"); affichera:  $\Box 02468$  $\Box$  8 6 4 2  $\square$  8 2  $\Box$  8 6 4 2 0 6. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n");  $\square$  il comporte une boucle infinie  $\square$  il ne compile pas □ il n'affiche rien ☐ il risque d'afficher bonjour à la place de coucou 7. Pour déclarer une fonction factorielle qui prend en

argument un entier et renvoie sa factorielle on écrit : □ struct int factorielle(int n):

□ struct int factorielle(int n
$\Box$ int factorielle(int x);
$\square$ int factorielle(double n);
☐ int factorielle();

8. Vous utilisez une boucle while quand: □ l'incrément de la variable de boucle n'est pas 1 □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction 9. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction  $\square$  int k:  $\square$  int %d; □ loop i;  $\square$  int loop n; 10. L'écriture 101 en binaire correspond au nombre naturel:  $\square$  5  $\Box$  4  $\square$  3

11. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

$\square$ n = factorielle();		
$\Box$ n = factorielle(p, q);		
☐ int factorielle(int 2);		
$\square$ printf("%d", factorielle(n));		

12. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :

$\square$ des processus	
$\Box$ en temps d'accès	
$\square$ les fichiers du disqu	ıe

 $\square$  101

□ certaines données de la mémoire de travail

```
13. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     ☐ Mineur
     \square rien
     □ Majeur
     ☐ Mineur
        Majeur
14. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      }
      return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \Box 0
15. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
```

```
for (i = 0; i < 4; i = i + 1)
         produit = produit * serie[i];
      printf("produit = %d", produit);
    La valeur affichée est :
      \Box 0
      \Box 4
      \square 8
      \Box 16
16. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt:
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
      \square for(i=0;i<5;i=i+1)
17. Sur unix (ou linux), la commande mkdir permet de :
      □ ouvrir un fichier texte
      □ créer un répertoire
      □ changer de répertoire courant
      \square créer un fichier texte
18. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
      ☐ (A == TRUE) && (B == TRUE)
      \square (!A || B)
      \square !(!A \mid | B) == (A \&\& !B)
      □ A && B
```

```
19. Soit le programme principal suivant :
   int main()
   {
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
20. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square int exposant(double n, int x);
     \square double exposant(double x, int n);
```

 $\square$  void exposant(double x^n);

 $\square$  exposant(double x, int n, int r);

## Éléments d'informatique – contrôle continue

Dránom .	NT
Prénom:	Nom:
N° INE ·	
TO INCE.	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

 $1.\ {\rm Si}$ a et b<br/> sont deux variables de type :

```
struct toto_s
{
   int n;
   double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition :

```
\square (a.n == b.n) \&\& (a.x == b.x)
```

- □ a == b
- $\square$  a = b
- $\square$  a{n, x} == b{n, x}

2. Un programme en langage C doit comporter une et une seule définition de la fonction :

- $\square$  include
- $\square$  init
- $\square$  begin
- $\square$  main

3. Pour compiler un programme prog.c, on utilise la ligne de commande :

```
\hfill\Box gcc prog.c -o -Wall prog.exe
```

- ☐ gcc prog.exe -Wall -o prog.c
- $\hfill\Box$  gcc -Wall prog.c -o prog.exe
- $\square$  gcc -Wall prog.exe -o prog.c

4. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\square x = racine(x * x) - racine(x);
```

- $\square$  x = racine(2/3);
- $\square$  x 1 = racine(x);
- $\square$  x = racine(racine(x)\*racine(x));

5. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

- void saisie\_utilisateur(int n);
- ☐ saisie\_utilisateur(scanf(%d));
- ☐ int saisie\_utilisateur();
  ☐ void saisie\_utilisateur(char c);
- 6. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
  - $\square$  n = carre(int n);
  - ☐ int carre(2); ☐ n = carre(n);
  - $\square$  int n = carre();
- 7. Quel est l'opérateur de différence en C :
  - □!
  - □ <>
  - □ !=
  - $\square \neq$

8. Après exécution jusqu'à la ligne 15 du programme C :

```
10 int main() {
11 int x = 5;
12 int y;
```

- 15 16 ...
- 17 }
  - $\Box$ la variable y vaut 5
  - $\Box\,$ la variable x vaut 5 et la variable y vaut 0
  - $\Box\,$ le programme affiche "Faux"
  - $\Box$ la variable x vaut 0

9. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :

- $\Box\,$ tour à tour, un petit peu à chaque fois
- $\Box\,$ en parallèle, chacun dans un registre
- $\Box$  to us ensemble
- $\Box\,$  chacun son tour, après que le processus précédent a terminé

10. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
```

appelant la fonction f ainsi définie :

```
int f(int a, int b)
{
   a = a + b;
   return a;
}
```

L'affichage dans le main est le suivant :

- $\Box$  f(a,b)=8, a=3, b=5
- $\Box$  f(a,b)=13, a=8, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=8, a=8, b=5

11. Le code suivant :

```
int i;
for (i = 4; i > 0; i = i - 1)
{
    printf("%d ", i);
}
printf("\n");
```

affichera:

- $\Box$  4 3 2 1 0
- $\Box 01234$
- $\square$  0 1 2 3
- $\square$  4 3 2 1

12. L'écriture <u>111</u> en binaire correspond au nombre naturel :

- $\Box$  7
- $\square$  3
- □ 111
- $\square$  8

13. L'écriture 101 en binaire correspond au nombre natu-	16. Un enregistrement permet de grouper plusieurs valeurs	□ 0 1 2 0 1 2
rel:	dans:	$\Box$ 0 1 0 1 0 1
$\Box$ 4	□ ses blocs	$\square$ 1 2 3 1 2
$\Box$ 5	$\square$ ses chants	$\Box$ 0 0 1 1 2 2
□ 101	$\square$ ses champs	19. Pour l'extrait de programme suivant :
	☐ ses cases	
14. Vous utilisez une boucle while quand :	17. Si x est une variable réelle (de type double) alors x = $3/2$ lui affecte la valeur :	<pre>int somme = 0; int serie[4] = {2, 4, 10, 4}; for (i = 0; i &lt; 4; i = i + 1)</pre>
□ vous avez déjà fait un <b>for</b> dans le même programme principal	□ 0 □ 0.5	<pre>{     somme = somme + serie[i];</pre>
$\hfill \square$ vous n'avez pas déclaré de fonction	□ 1.5	}
□ vous ne connaissez pas le nombre d'itérations de	□ 1	<pre>printf("somme = %d",somme);</pre>
la boucle à l'avance	18. Pour l'extrait de programme suivant :	La valeur de somme affichée est :
$\hfill\Box$ l'incrément de la variable de boucle n'est pas $1$	int i = 0;	
15. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	<pre>int j = 0; for (i = 0; i &lt; 3; i = i + 1) {     for (j = 0; j &lt; 2; j = j + 1)</pre>	□ 16 □ 3 □ 20
$\hfill \square$ une directive préprocesseur #include manquante	{	20. Un bit est:
$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction	<pre>printf("%d ", i); } </pre>	☐ la longueur d'un mot mémoire ☐ l'instruction qui met fin à un programme
$\hfill\Box$ une fonction appelée avant sa déclaration	<pre>printf("\n");</pre>	$\square$ un chiffre binaire (0 ou 1)
$\Box$ une fonction déclarée mais non définie	qu'est ce qui sera affiché?	$\hfill\Box$ un battement d'horloge processeur

 $\square$  0.5

 $\square$  1.5

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme? □ un désaccord entre la déclaration et la définition d'une fonction  $\square$  une fonction déclarée mais non définie  $\square$  une fonction appelée avant sa déclaration ☐ une directive préprocesseur #include manquante 2. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :  $\square$  void pgcd(int x, int y);  $\square$  int pgcd(int y, int x);  $\Box$  int pgcd(int x, int x);  $\Box$  int pgcd(int x, y); 3. Le code suivant : int i; for (i = 8; i > 0; i = i - 2)printf("%d ", i); } printf("\n"); affichera:  $\square$  8 2  $\Box$  8 6 4 2  $\Box$  8 6 4 2 0  $\Box 02468$ 4. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\Box$  0  $\Box$  1

```
5. Avant de faire appel à une fonction il est nécessaire
  de:
    \Box avoir défini une constante symbolique de la taille
        de cette fonction
     □ l'avoir définie
     □ l'avoir déclarée
     □ l'avoir déclarée et définie
6. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
  {
     printf("coucou\n");
     □ il comporte une boucle infinie
     □ il n'affiche rien
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il ne compile pas
7. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     \square init
     □ main
     \square begin
     □ include
8. Si le code:
   struct toto_s
     int n;
     double x;
  };
   précède la fonction main(), alors on peut écrire en
  début de main() :
     \square toto_s n, x;
     □ struct toto_s toto;
     \square int struct toto_s = {3, -1e10};
     \square int toto.n = 3;
```

 $\Box$  toto s struct z = {3, 0.5}:

```
9. Vous utilisez une boucle while quand:
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
      \square vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      \squarevous n'avez pas déclaré de fonction
      □ l'incrément de la variable de boucle n'est pas 1
10. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt :
      \square for(i=1;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
11. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
              printf("%d ", j);
     }
    qu'est ce qui sera affiché?
     \Box 0 1 2 3 0 1 2
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2 3
      \Box 0 1 2 0 1 2 3
12. Lorsqu'un programme utilise printf ou scanf il faut
    qu'il contienne l'instruction préprocesseur :
      ☐ #include <stdio.h>
      ☐ #include <studio.h>
```

☐ #include <studlib.h>

☐ #appart <stdlib.h>

```
13. Soit le programme principal suivant :
   int main()
   {
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
   {
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
      \Box f(a,b)=8, a=3, b=5
14. Une de ces manière de composer les blocs de pro-
   grammes ne fait pas partie des opérations de la pro-
   grammation structurée:
      \square sélectionner entre deux blocs à l'aide d'une condi-
        tion
     □ répéter un bloc tant qu'une condition est vérifée
      \square mettre les blocs en séquence les uns à la suite des
        autres
      □ retourner un bloc
```

```
15. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box j = %d
     \Box i = 5
     \Box j = 4
     \Box j = 0
16. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
     ☐ une directive préprocesseur #include manquante
     □ une fonction déclarée mais non définie
17. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     \square une accolade manquante
     \square une accolade en trop
```

```
☐ un point-virgule manquant
      □ un point-virgule en trop
18. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int n = carre();
     \square int carre(2);
     \square n = carre(n);
     \square n = carre(int n);
19. Le code suivant :
     int i:
     for (i = 0; i < 7; i = i + 2)
          printf("%d ", i);
     printf("\n");
    affichera:
     \square 0 2 4 6
     \Box 0123456
     \Box 0 1 2 3 4 5 6 7
     \Box 02468
20. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%d et y=%d\n,x,y");
     \Box printf("x=%d et y=%d\n",x,y);
     \Box printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n",x y);
```

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Après exécution du programme :
- 1 lecture 8 r0
- 2 valeur 3 r1
- 3 mult r1 r0
- 4 valeur 1 r2
- 5 add r2 r0
- 6 ecriture r0 8
- 7 stop
- 8 5
  - $\square$  le bus explose
  - □ le terminal affiche 8
  - □ la case mémoire 8 contiendra 0
  - $\Box$  la case mémoire 8 contiendra 16
- 2. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?
  - $\square !(!A || B) == (A \&\& !B)$
  - □ A && B
  - $\square$  (A == TRUE) && (B == TRUE)
  - $\square$  (!A || B)
- 3. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
  - ☐ #define N 3
  - $\square$  #define taille = N
  - $\square$  #define N = 3
  - $\square$  #define taille = 3
- 4. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
  - $\Box$  certaines données de la mémoire de travail
  - $\square$  des processus
  - $\Box$  en temps d'accès
  - □ les fichiers du disque

- 5. Au début de la fonction main() on place le code :
  - char i;
    for (i = 'A'; i <= 'F'; i = i + 1)
    {
     printf("%c", i);</pre>
  - Alors l'affichage sera :

printf("\n");

- $\square$  i
- □ cccccc
- □ A
- ☐ ABCDEF
- 6. Après exécution jusqu'à la ligne 15 du programme C :
- 10 int main() {
  11 int x = 5;
  12 int y;
  13
- 14 y = x; 15
- 16 ... 17 }
- □ la variable y vaut 5
  - $\Box$  la variable x vaut 0
  - □ le programme affiche "Faux"
  - $\square$  la variable x vaut 5 et la variable y vaut 0
- 7. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
  - $\square$  for(i=1;i<5;i=i+1)
  - $\square$  for(i=1;i<=5;i=i+1)
  - ☐ for(i=0;i<=5;i=i+1)
  - $\square$  for(i=0;i<5;i=i+1)
- 8. Soit la fonction g définie par :

```
int g(int a)
{
  printf("a = \n", %d);
  if (1 > 0)
  {
```

return 7;
}

return 5:

- Alors l'expression g(0) prendra la valeur :
  - $\Box$  5  $\Box$  0
  - □ 7
- 9. Vous utilisez une boucle while quand :
  - □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
  - $\Box$ vous n'avez pas déclaré de fonction
  - $\hfill \square$  l'incrément de la variable de boucle n'est pas 1
  - □ vous avez déjà fait un for dans le même programme principal
- 10. L'écriture <u>111</u> en binaire correspond au nombre naturel :
  - $\square$  3
  - $\Box$  7
  - □ 111
  - $\square$  8
- 11. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :
  - $\square$  int pgcd(int x, int x);
  - $\square$  int pgcd(int y, int x);
  - $\square$  int pgcd(int x, y);
  - □ void pgcd(int x, int y);
- 12. Dans la commande gcc, l'option -Wall signifie :
  - $\square$  qu'il faut indenter le fichier source
  - $\square$  qu'on veut changer alétoirement de fond d'écran
  - □ qu'il faut lancer un déboggueur
  - $\square$  que l'on veut voir tous les avertissements

```
13. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \Box 4
      \square 3
      \Box 0
14. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
      \square int %d;
      \square int loop n;
      \square loop i;
      \square int k;
15. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         {
```

```
printf("j = %d\n", j);
    qu'est ce qui sera affiché par ce printf?
      \Box j = 0
      □ j = 5
      \Box j = 4
      \Box j = %d
16. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
    une variable entière définie et initialisée, il est correct
    d'écrire :
      \square n = carre(int n);
      \square n = carre(n);
      \square int n = carre();
      \square int carre(2);
17. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \Box 5
      \Box 4
```

	□ 1
18.	Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?
	$\square$ une fonction déclarée mais non définie
	$\Box$ une directive préprocesseur #include manquante
	$\Box$ un désaccord entre la déclaration et la définition d'une fonction
	$\square$ une fonction appelée avant sa déclaration
19.	Quel est le problème d'un programme comportant les lignes suivantes?
	<pre>while (1) {    printf("coucou\n");</pre>
	}
	$\hfill\Box$ il comporte une boucle infinie
	$\Box$ il n'affiche rien
	$\Box$ il risque d'afficher bonjour à la place de coucou
	$\square$ il ne compile pas
20.	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	$\square$ analyse syntaxique
	$\square$ analyse lexicale
	$\square$ analyse sémantique
	$\square$ analyse harmonique

 $\Box$  0

т		-1
	100000	- 1
	acence	

Prénom :	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le langage C est un langage □ lu, écrit, parlé

2. Pour déclarer une procédure afficher\_date qui prend en argument un struct date\_s et affiche le contenu du struct, on écrit :

□ void afficher\_date(date\_s d); ☐ int afficher\_date(date\_s d);

□ void afficher\_date(struct date\_s d);

□ struct date\_s afficher\_date(struct date\_s d);

3. Le code suivant :

□ interprété

□ composé □ compilé

```
int i;
 for (i = 0; i < 7; i = i + 2)
     printf("%d ", i);
printf("\n");
affichera:
 \Box 0 1 2 3 4 5 6 7
```

 $\Box 0246$ 

 $\Box 0123456$ 

 $\Box 02468$ 

4. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :

```
\square int n = carre();
\square n = carre(int n);
\square n = carre(n);
\square int carre(2);
```

5. Le code suivant :

```
int i;
for (i = 1; i < 5; i = i + 1)
     printf("%d ", i);
printf("\n");
affichera:
 \Box 43210
 \square 4 3 2 1
```

 $\square$  1 2 3 4  $\Box 01234$ 

6. Si factorielle est une fonction prenant en entrée un entier et renvovant un entier, il est correct d'écrire :

```
□ printf("%d", factorielle(n));
\square n = factorielle(p, q);
```

☐ int factorielle(int 2);

 $\square$  n = factorielle():

7. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de  $x^n$  on écrit :

```
\square int exposant(double n, int x);
```

 $\square$  void exposant(double x^n);

 $\square$  double exposant(double x, int n);

 $\square$  exposant(double x, int n, int r);

8. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :

```
\square printf("x=%d et y=%d\n",x y);
```

 $\square$  printf("x=%d et y=%d\n",x,y);

 $\square$  printf("x=%d et y=%d\n,x,y");

 $\square$  printf("x=%x et y=%y\n");

```
9. Soit le programme principal suivant :
```

```
int main()
₹
 int a = 3:
 int b = 5:
 printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
 return EXIT_SUCCESS;
appelant la fonction f ainsi définie :
int f(int a, int b)
  a = a + b;
  return a;
L'affichage dans le main est le suivant :
 \Box f(3,5)=8, a=3, b=5
 \Box f(a,b)=13, a=8, b=5
 \Box f(a,b)=8, a=3, b=5
```

10. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme?

□ un caractère interdit en C

 $\Box$  f(a,b)=8, a=8, b=5

 $\square$  une variable non déclarée

☐ une directive préprocesseur #include manquante

 $\square$  une faute de frappe dans un appel de fonction

11. Au début de la fonction main() on place le code :

```
char i:
 for (i = 'A'; i \le 'F'; i = i + 1)
   printf("%c", i);
 printf("\n");
Alors l'affichage sera:
 □ ccccc
 \Box i
```

 $\Box$  A

☐ ABCDEF

```
12. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 0
     \square 15
     \Box 10
     \Box 6
13. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     □ new TP4
     ☐ kwrite TP4
     ☐ mkdir TP4
     ☐ yppasswd
14. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
    for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
             printf("%d ", i);
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 1 2 3 1 2
```

```
\Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1
     \Box 0 0 1 1 2 2
15. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3:
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 0
      \Box 4
      \square 3
16. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
     printf("\n");
    affichera:
      \square 8 2
     \square 8 6 4 2
     \Box 02468
      \Box 8 6 4 2 0
17. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
      ☐ char "c";
      ☐ char c:
      ☐ int char;
      □ char 'c';
```

```
18. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt :
      \square for(i=0;i<5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
      \square for(i=1;i<5;i=i+1)
19. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses champs
      \square ses blocs
      \square ses cases
      \square ses chants
20. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
        scanf("%d", &n);
    Quelle est la condition cond:
      □ a<=n<=b
      □ (n<=a) && (n<=b)
```

☐ (a<=n) && (n<=b)

 $\square$  (a<n) || (n>b)

## Éléments d'informatique – contrôle continue

}

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre:
  - $\Box\,$  dans lequel vous avez déclaré ces fonction
  - $\square$  dans lequel ces fonctions sont appelées dans le main
  - $\square$  alphabétique
  - $\square$  un ordre quelconque
- 2. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
   scanf("%d", &n);
}
```

Quelle est la condition cond :

```
\square (a<=n) && (n<=b)
```

□ a<=n<=b

☐ (n<=a) && (n<=b)

☐ (a<n) || (n>b)

3. Si le code :

```
struct toto_s
{
  int n;
  double x;
};
```

précède la fonction main(), alors on peut écrire en début de main() :

```
\Box toto_s struct z = {3, 0.5};
\Box int toto.n = 3;
```

 $\Box$  int struct toto\_s = {3, -1e10};

☐ struct toto\_s toto;

□ toto\_s n, x;

```
4. Soit un programme contenant les lignes suivantes :

int i = 0;
int j = 0;
for (i = 0; i < 2; i = i + 1)
{

for (j = 0; j < 3; j = j + 1)
}
```

printf("%d ", i);

qu'est ce qui sera affiché?

```
\square \ 0 \ 0 \ 0 \ 1 \ 1 \ 1
```

 $\square$  0 1 2 0 1 2

 $\Box$  1 2 1 2 3

 $\Box$  0 1 0 1 0 1 0 1

5. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :

```
\square #include <studlib.h>
```

☐ #appart <stdlib.h>

☐ #include <stdio.h>

 $\Box$  #include <studio.h>

6. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
  printf("coucou\n");
}
```

 $\Box\,$ il comporte une boucle infinie

 $\square$  il n'affiche rien

 $\Box\,$ il risque d'afficher bonjour à la place de coucou

 $\Box\,$ il ne compile pas

7. Le langage C est un langage

□ compile	É
-----------	---

 $\Box \ \ {\rm compos\acute{e}}$ 

□ interprété

 $\Box\,$ lu, écrit, parlé

8. Pour compiler un programme prog.c, on utilise la ligne de commande :

 $\square$  gcc prog.exe -Wall -o prog.c

 $\square$  gcc -Wall prog.c -o prog.exe

☐ gcc -Wall prog.exe -o prog.c ☐ gcc prog.c -o -Wall prog.exe

9. Les lignes

```
int i;
int x=0;
for(i=0,i<5,i=i+1)
{
    x=x+1;
}</pre>
```

 $\square$  ne comportent aucune erreur

 $\Box$  comportent une erreur qui ne sera pas détectée

□ comportent une erreur qui sera détectée au cours de l'analyse syntaxique

□ comportent une erreur qui sera détectée au cours de l'édition de lien

10. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?

 $\square$  une fonction appelée avant sa déclaration

 $\Box\,$  une directive préprocesseur  $\# \mathtt{include}$  man quante

☐ un désaccord entre la déclaration et la définition d'une fonction

 $\hfill \square$  une fonction déclarée mais non définie

11. Dans la commande gcc, l'option -Wall signifie :

 $\Box\,$  qu'on veut changer alétoirement de fond d'écran

 $\Box$  qu'il faut indenter le fichier source

☐ que l'on veut voir tous les avertissements

 $\Box\,$ qu'il faut lancer un déboggueur

12. L'écriture  $\underline{101}$  en binaire correspond au nombre naturel :

 $\Box$  5

 $\Box$  4

□ 101

 $\square$  3

13. Soit la fonction g définie par :	15. Le bus système sert à :	$\hfill\Box$ un point-virgule man quant
int g(int a)	☐ Arriver à l'heure en cours	$\hfill\Box$ une accolade en trop
{	$\hfill \square$ Transférer des données et intructions entre pro-	$\square$ un point-virgule en trop
printf("a = \n", %d); if (1 > 0)	cesseur et mémoire	19. Soit la fonction <b>f</b> définie par :
11 (1 > 0) {	☐ Écrire des données sur le dique dur	int f(int a)
return 5;	$\Box$ transporter les processus du tourniquet au pro-	f
}	cesseur	<pre>printf("a = \n", %d);</pre>
return 7;	16. Une variable booléenne est un variable :	if (a > 0)
, Al. 11 (2)	□ NaN (not a number, qui n'est pas un nombre)	{ 
Alors l'expression g(0) prendra la valeur :	$\square$ jamais nulle	return 3;
	$\Box$ à la quelle une valeur vient d'être affectée	return 4;
	☐ réelle positive	}
□ 5	$\square$ qui est vraie ou fausse	Alors l'expression f(0) prendra la valeur :
14. Une <i>segmentation fault</i> est une erreur qui survient lorsque:	17. Sous unix (ou linux), pour créer un répertoire TP4	$\Box$ 0
□ le programme tente d'afficher des caractères sur	dans le répertoire courant on peut utiliser la com-	$\square$ 3
une ligne qui va au delà de la largeur de la fenêtre	mande:	$\Box$ 4
du terminal	□ yppasswd	20. Avant de faire appel à une fonction il est nécessaire
$\Box$ le programme tente d'accèder à une partie de la	□ kwrite TP4	de:
mémoire qui ne lui est pas réservée	□ new TP4	☐ l'avoir déclarée et définie
□ la division du programme en zones homogènes	□ mkdir TP4	$\square$ avoir défini une constante symbolique de la taille
échoue	18. Si cette erreur apparaît à la compilation :	de cette fonction
☐ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne	error: expected ';' before '}' token que doit- on chercher dans le programme?	☐ l'avoir déclarée
peut pas être chargé par le compilateur	une accolade manquante	☐ l'avoir définie
· ·		

## Éléments d'informatique – contrôle continue

Prénom: Nom	:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :
  - $\Box$  0
  - $\square$  0.5
  - $\Box$  1
  - $\square$  1.5
- 2. Une variable booléenne est un variable :
  - □ à laquelle une valeur vient d'être affectée
  - □ réelle positive
  - $\Box$  qui est vraie ou fausse
  - □ NaN (not a number, qui n'est pas un nombre)
  - $\Box$  jamais nulle
- 3. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie :
```

L'affichage dans le main est le suivant :

 $\Box$  f(a,b)=8, a=8, b=5

int f(int a, int b)

a = a + b;

return a;

- $\square$  f(a,b)=8, a=3, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=13, a=8, b=5

- 4. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?
  - $\square$  une fonction appelée avant sa déclaration
  - un désaccord entre la déclaration et la définition d'une fonction
  - $\Box$  une directive préprocesseur #include manquante
  - $\Box\,$ une fonction déclarée mais non définie
- 5. Le code suivant :

- $\Box \ 0\ 1\ 2\ 3\ 4$
- $\square$  4 3 2 1
- $\square$  0 1 2 3
- 6. Pour l'extrait de programme suivant :

```
int somme = 0;
for (i = 0; i < 5; i = i + 1)
{
   somme = somme + i;
   i = i + 1; /* attention ! */
}
printf("somme = %d",somme);</pre>
```

La valeur de somme affichée est :

- $\Box$  0
- $\Box$  15
- □ 10
- $\Box$  6
- 7. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :

	alphabétique
--	--------------

- $\square$  dans lequel vous avez déclaré ces fonction
- □ un ordre quelconque
- $\Box$  dans lequel ces fonctions sont appelées dans le main
- 8. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
  - $\Box$ les fichiers du disque
  - $\Box\,$ en temps d'accès
  - $\Box$  des processus
  - $\Box$  certaines données de la mémoire de travail
- 9. Le code suivant :

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
```

#### affichera:

- $\square\ \, 8\ \, 6\ \, 4\ \, 2\ \, 0$
- $\Box \ 0\ 2\ 4\ 6\ 8$
- $\square$  8 6 4 2
- $\square$  8 2
- 10. Un enregistrement permet de grouper plusieurs valeurs dans :
  - $\square$  ses chants
  - $\Box$  ses champs
  - $\square$  ses blocs
  - $\square$  ses cases
- 11. Un bit est:
  - $\square$  un battement d'horloge processeur
  - $\square$  l'instruction qui met fin à un programme
  - $\square$  un chiffre binaire (0 ou 1)
  - $\square$  la longueur d'un mot mémoire

```
12. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
     else
         printf("Majeur\n");
   affichera:
      ☐ Mineur
      □ Majeur
      \square rien
      ☐ Mineur
         Majeur
13. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      □ 111
      \square 7
      \square 3
      \square 8
14. Vous utilisez une boucle while quand :
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
      \square vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
     □ vous n'avez pas déclaré de fonction
     \Box l'incrément de la variable de boucle n'est pas 1
```

```
15. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ char afficher_menu(printf("menu"));
     □ void afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
16. Un fichier source est:
     \square un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
     □ un document qui doit être protégé
     \square un document illisible pour les humains
     □ un fichier texte qui sera traduit en instructions
        processeur
     □ un document de référence du système
17. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     □ un point-virgule manquant
     \square une accolade en trop
     □ un point-virgule en trop
      \square une accolade manquante
18. Pour l'extrait de programme suivant :
     int i;
     int j;
     for(i=4;i>0;i=i-1)
       for(j=i;j<6;j=j+1)
         printf("*");
```

```
printf(" ");
   qu'est ce qui sera affiché?
      ** ** ** ** **
     ***** **** ***
         **** **** ****
19. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
        return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \Box 0
20. Sous unix (ou linux), la commande 1s permet de :
     □ afficher la liste de fichiers contenus dans un
        répertoire
     \square afficher le contenu d'un fichier texte
     □ compiler un programme
```

 $\square$  voir des clips musicaux

т		-1
	acence	

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :  $\Box$  int factorielle(int x); ☐ int factorielle(); ☐ int factorielle(double n); ☐ struct int factorielle(int n); 2. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme?  $\square$  une variable non déclarée □ un caractère interdit en C  $\square$  une directive préprocesseur #include manquante  $\square$  une faute de frappe dans un appel de fonction 3. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return f(a - 1) + 1; return 4; Alors l'expression f(1) prendra la valeur :  $\Box$  4  $\Box$  1  $\Box$  0  $\Box$  5 4. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n");

printf("Majeur\n");

4	de controle continue	1
	affichera:	
	☐ Mineur	
	☐ Mineur	
	Majeur	
	□ Majeur	
	□ rien	
5.	Le code suivant :	
	<pre>int age = 20;</pre>	
	if (age < 18)	
	<pre>{     printf("Mineur\n");</pre>	
	}	
	else	
	{	
	<pre>printf("Majeur\n"); }</pre>	
	J	
	affichera:	
	☐ Mineur	
	$\square$ Majeur	
	□ rien	
	☐ Mineur	
	Majeur	
6.	Si carre est une fonction prenant en entrée un en-	
	tier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct	
	d'écrire :	
	$\square$ n = carre(n);	
	☐ int n = carre();	
	☐ int carre(2);	
	$\square$ n = carre(int n);	
7.	Si cette erreur apparaît à la compilation :	
	erreur: conflicting types for 'max', que doit- on chercher dans le programme?	
	$\Box$ un désaccord entre la déclaration et la définition	
	d'une fonction	
	$\Box$ une directive préprocesseur $\# include \ manquante$	:
	$\square$ une fonction appelée avant sa déclaration	
	□ une fonction déclarée mais non définie	

8. Dans la commande gcc, l'option -Wall signifie :
$\hfill \square$ qu'on veut changer alétoirement de fond d'écran
$\hfill\Box$ qu'il faut lancer un déboggueur
$\hfill\Box$ que l'on veut voir tous les avertissements
$\hfill\Box$ qu'il faut indenter le fichier source
9. Un enregistrement permet de grouper plusieurs valeur dans :
$\square$ ses champs
$\square$ ses chants
□ ses cases
$\square$ ses blocs
10. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
□ 111
$\Box$ 7
$\Box$ 3
□ 8
11. Lorsqu'un programme utilise printf ou scanf il fau qu'il contienne l'instruction préprocesseur :
☐ #appart <stdlib.h></stdlib.h>
$\square$ #include <stdio.h></stdio.h>
$\square$ #include <studio.h></studio.h>
$\square$ #include <studlib.h></studlib.h>
12. Sur unix (ou linux), la commande mkdir permet de :
$\square$ ouvrir un fichier texte
$\hfill \Box$ changer de répertoire courant
$\Box$ créer un fichier texte
$\Box$ créer un répertoire
13. Le type des réels en C est :
□ char
$\Box$ int
$\square$ double
□ real

14. Si x est une variable réelle (de type double) alors $x = 3/2$ lui affecte la valeur :	17. Vous utilisez une boucle while quand :  □ vous n'avez pas déclaré de fonction	} }
$\Box$ 0	$\Box$ l'incrément de la variable de boucle n'est pas 1	qu'est ce qui sera affiché?
□ 0.5	$\hfill \square$ vous ne connaissez pas le nombre d'itérations de	$\Box$ 0 1 2 0 1 2 3
□ 1.5	la boucle à l'avance	$\Box$ 0 0 1 1 2 2 3
□ 1	□ vous avez déjà fait un for dans le même pro- gramme principal	$\Box$ 0 1 2 0 1 2
	18. Pour déclarer une procédure afficher_date qui prend	$\Box$ 0 1 2 3 0 1 2
15. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :	en argument un struct date_s et affiche le contenu	20. Soit la fonction <b>f</b> définie par :
□ n = mccarthy();	du struct, on écrit :	int f(int a)
·	☐ int afficher_date(date_s d);	<pre>1 d). printf("a = \n", %d);</pre>
□ x = mccarthy(n);	☐ struct date_s afficher_date(struct date_s ☐ void afficher_date(struct date_s d);	if (a > 0)
☐ int mccarthy(int 2);	□ void afficher_date(date_s d);	{ return 3;
$\square$ n = mccarthy(p, q);	19. Pour l'extrait de programme suivant :	}
16. Un bit est:	int i = 0;	return 4;
$\Box$ un battement d'horloge processeur	int j = 0;	41 11 1 (42)
$\Box$ la longueur d'un mot mémoire	for (i = 0; i < 2; i = i + 1)	Alors l'expression $f(0)$ prendra la valeur : $\square 0$
$\Box$ un chiffre binaire (0 ou 1)	for $(j = 0; j < 3; j = j + 1)$	□ 4
$\hfill\Box$ l'instruction qui met fin à un programme	<pre>{     printf("%d ", j);</pre>	

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

ar réponse fausse. Durée : 20 minutes.
1. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
☐ int saisie_utilisateur();
☐ saisie_utilisateur(scanf(%d));
☐ void saisie_utilisateur(int n);
☐ void saisie_utilisateur(char c);
2. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
☐ n = carre(int n);
☐ int n = carre();
$\square$ n = carre(n);
☐ int carre(2);
3. Le langage C est un langage
$\Box$ lu, écrit, parlé
$\square$ interprété
$\square$ composé
$\square$ compilé
4. Vous utilisez une boucle while quand :
$\Box$ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
$\hfill \square$ l'incrément de la variable de boucle n'est pas $1$
□ vous avez déjà fait un <b>for</b> dans le même programme principal
$\square$ vous n'avez pas déclaré de fonction
5. Le bus système sert à :
$\Box$ transporter les processus du tourniquet au processeur
$\square$ Arriver à l'heure en cours
$\Box$ Transférer des données et intructions entre processeur et mémoire
☐ Écrire des données sur le dique dur

```
6. Sous unix (ou linux), la commande 1s permet de :
     \square afficher la liste de fichiers contenus dans un
        répertoire
     □ voir des clips musicaux
     □ compiler un programme
    \square afficher le contenu d'un fichier texte
7. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 1 2 3 4
    \square 4 3 2 1
8. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
```

 $\Box$  f(a,b)=13, a=8, b=5

9.	Une variable booléenne est un variable :
	□ réelle positive
	□ NaN (not a number, qui n'est pas un nombre)
	$\square$ jamais nulle
	$\Box$ à la quelle une valeur vient d'être affectée
	$\Box$ qui est vraie ou fausse
10.	Le code suivant :
	int i;
	for (i = 4; i >= 0; i = i - 1) {
	printf("%d ", i);
	}
	<pre>printf("\n");</pre>
	affichera:
11.	Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :
	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition
	$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres
	$\Box$ retourner un bloc
	$\Box$ répéter un bloc tant qu'une condition est vérifée
12.	Quel est le problème d'un programme comportant les lignes suivantes?
	while (1) {
	<pre>printf("coucou\n"); }</pre>
	$\Box$ il risque d'afficher bonjour à la place de coucou
	$\Box$ il comporte une boucle infinie
	$\Box$ il n'affiche rien
	☐ il ne compile pas

```
13. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit:
      □ struct date_s afficher_date(struct date_s | d);
     □ void afficher_date(struct date_s d);
      □ void afficher_date(date_s d);
     ☐ int afficher_date(date_s d);
14. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      \square 8
     \Box 111
      \square 7
     \square 3
15. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
     \Box 4
     \Box 0
     \square 8
      \Box 16
```

```
16. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
              printf("%d ", i);
         }
     }
    printf("\n");
    qu'est ce qui sera affiché?
     \Box 0 0 1 1 2 2
     \Box 0 1 2 0 1 2
     \Box 1 2 3 1 2
     \Box 0 1 0 1 0 1
17. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ int char:
     □ char 'c':
     ☐ char "c";
      \Box char c;
18. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
```

```
scanf("%d", &n);
    Quelle est la condition cond :
      \square (a<=n) && (n<=b)
      \square (n<=a) && (n<=b)
      \Box (a<n) || (n>b)
      □ a<=n<=b
19. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3:
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \Box 4
      \square 3
      \Box 0
20. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
      ☐ une fonction appelée avant sa déclaration
      \square une fonction déclarée mais non définie
     □ une directive préprocesseur #include manquante
     \square un désaccord entre la déclaration et la définition
```

d'une fonction

# Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si pgcd est une fonction prenant en entrée deux entiers
  et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
     \square n = pgcd(n, 3);
     \square n = pgcd(int p, int q);
     \square int pgcd(2);
2. Soit la fonction g définie par :
  int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
     \square 0
     \square 7
     \Box 5
3. Si le code :
  struct toto_s
  {
     int n:
     double x;
  précède la fonction main(), alors on peut écrire en
  début de main() :
     \square int struct toto_s = {3, -1e10};
     □ struct toto_s toto;
     \square int toto.n = 3;
     \Box toto_s struct z = {3, 0.5};
```

 $\square$  toto\_s n, x;

```
4. Les lignes
   int i;
   int x=0;
   for(i=0,i<5,i=i+1)
     x=x+1;
     \square ne comportent aucune erreur
    □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
    \Box comportent une erreur qui sera détectée au cours
        de l'édition de lien
     □ comportent une erreur qui ne sera pas détectée
5. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     \square une accolade en trop
     \square une accolade manquante
     □ un point-virgule en trop
     ☐ un point-virgule manquant
6. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int i = 0;
    for (i = 0; i < 0; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box i = 0
     \Box j = %d
     \Box j = 4
     \Box i = 5
```

```
7. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 4
     \square 8
     \Box 0
     \Box 16
 8. Un enregistrement permet de grouper plusieurs valeurs
    dans:
     \square ses blocs
     □ ses cases
     □ ses chants
      \square ses champs
 9. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
    Quelle est la condition cond:
     \square (a<n) || (n>b)
     □ a<=n<=b
      ☐ (a<=n) && (n<=b)
      □ (n<=a) && (n<=b)
10. Le type des réels en C est :
     \square int
     □ double
      □ char
```

□ real

11. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :	if (a > 0) {	17. Soier 5 res
☐ char afficher_menu(printf("menu"));	return f(a - 1) + 1;	avec
□ void afficher_menu();	}	
☐ int afficher_menu(int char);	return 4;	
☐ double afficher_menu();	Alors l'expression f(1) prendra la valeur :	
☐ int afficher_menu();		
12. Pour déclarer une procédure afficher_date qui prend		18. Le la
en argument un struct date_s et affiche le contenu	□ 5	
du struct, on écrit :	□ 0	
<pre>□ void afficher_date(date_s d);</pre> □ int afficher_date(date_s d);	15. Au début de la fonction main() on place le code :	
□ void afficher_date(struct date_s d);	char b = 'A';	
struct date_s afficher_date(struct date_s)	h = h + 2	19. Pour
13. Quel est le problème d'un programme comportant les lignes suivantes?	Alors l'affichage sera :	le pl pass
while (1)	□ A	
{	□В	
<pre>printf("coucou\n");</pre>	□b	
}	□ С	
☐ il risque d'afficher bonjour à la place de coucou	16. Un fichier source est:	20. Sur
$\square$ il ne compile pas	□ un document qui doit être protégé	men
$\square$ il comporte une boucle infinie	$\square$ un document illisible pour les humains	
□ il n'affiche rien	$\hfill \square$ un document de référence du système	
14. Soit la fonction f définie par :	☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur	
<pre>int f(int a) {    printf("a = \n", %d);</pre>	☐ un fichier texte qui sera traduit en instructions processeur	

17. Soient deux variables entières x et 5 respectivement. L'affichage x=4 avec la commande :	•
$\Box$ printf("x=%x et y=%y\n")	;
$\Box$ printf("x=%d et y=%d\n",	x,y);
$\Box$ printf("x=%d et y=%d\n",	х у);
$\Box$ printf("x=%d et y=%d\n,x	,y");
18. Le langage C est un langage	
$\square$ lu, écrit, parlé	
$\Box$ interprété	
$\square$ composé	
$\square$ compilé	
19. Pour déclarer une fonction pgcd que le plus grand diviseur commun de passés en arguments on écrit :	
$\square$ void pgcd(int x, int y);	
$\Box$ int pgcd(int y, int x);	
$\Box$ int pgcd(int x, int x);	
$\Box$ int pgcd(int x, y);	
20. Sur un ordinateur avec un seul pr ment les processus sont exécutés :	
$\hfill\Box$ en parallèle, chacun dans un	registre
$\Box$ tous ensemble	
$\Box$ tour à tour, un petit peu à c	haque fois
☐ chacun son tour, après que le a terminé	processus précédent

т	•	-1
	acence	

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution du programme :
     lecture 8 r0
      valeur 3 r1
     mult r1 r0
     valeur 1 r2
     add r2 r0
     ecriture r0 8
      stop
     5
    \Box le terminal affiche 8
    □ la case mémoire 8 contiendra 16
    \square le bus explose
    □ la case mémoire 8 contiendra 0
2. Soit la fonction f définie par :
  int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
  Alors l'expression f(0) prendra la valeur :
    \Box 0
    \Box 4
    \square 3
3. Pour l'extrait de programme suivant :
     int produit = 1;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
  La valeur affichée est :
```

```
\Box 0
     \square 16
     \Box 4
     \square 8
4. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera :
     \Box C
     \square A
     □b
     □В
5. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre :
     □ dans lequel vous avez déclaré ces fonction
     □ dans lequel ces fonctions sont appelées dans le
        main
     □ alphabétique
     \square un ordre quelconque
6. Une variable booléenne est un variable :
     ☐ réelle positive
     □ jamais nulle
     \square qui est vraie ou fausse
    □ NaN (not a number, qui n'est pas un nombre)
     □ à laquelle une valeur vient d'être affectée
7. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     □ new TP4
     ☐ kwrite TP4
     ☐ mkdir TP4
     ☐ yppasswd
```

```
8. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
      ☐ char "c";
     □ char 'c';
     ☐ int char:
      □ char c:
 9. Un registre du processeur est :
      \square un composant qui contient la liste des fichiers du
        système
      □ une gamme de fréquence de fonctionnement du
        processeur
      □ une unité de calcul spécialisée de l'ordinateur
      □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
10. Les lignes
   int i;
   int x=0:
   for(i=0, i<5, i=i+1)
   {
      x=x+1;
      \square ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
      □ comportent une erreur qui ne sera pas détectée
      □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
11. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
      □ printf("%d", factorielle(n));
     \square n = factorielle(p, q);
     ☐ int factorielle(int 2);
      \square n = factorielle();
```

```
12. Soit la fonction g définie par :
                                                              15. Le code suivant :
                                                                                                                                 affichera:
    int g(int a)
                                                                                                                                   \Box \ 4\ 3\ 2\ 1\ 0
                                                                    int age = 18;
    {
                                                                    if (age < 18)
                                                                                                                                   \square 4 3 2 1
      printf("a = \n", %d);
                                                                    {
                                                                                                                                   \Box 01234
      if (1 > 0)
                                                                        printf("Mineur\n");
                                                                                                                                   \square 1 2 3 4
         return 5;
                                                                    else
                                                                                                                             18. Si cet avertissement apparaît à la compilation :
                                                                    {
                                                                                                                                 warning: implicit declaration of function 'max'
      return 7;
                                                                        printf("Majeur\n");
                                                                                                                                 , que doit-on chercher dans le programme?
    Alors l'expression g(0) prendra la valeur :
                                                                                                                                   □ une fonction déclarée mais non définie
      \square 7
                                                                   affichera:
                                                                                                                                   ☐ une fonction appelée avant sa déclaration
      \Box 5
                                                                    \square rien
                                                                                                                                   □ une directive préprocesseur #include manquante
      \Box 0
                                                                    □ Majeur
                                                                                                                                   \square un désaccord entre la déclaration et la définition
13. Dans la commande gcc, l'option -Wall signifie :
                                                                                                                                      d'une fonction
                                                                     ☐ Mineur
      □ qu'il faut indenter le fichier source
                                                                     ☐ Mineur
                                                                                                                             19. Si x est une variable réelle (de type double) alors
      ☐ que l'on veut voir tous les avertissements
                                                                                                                                 x = 3/2 lui affecte la valeur :
                                                                        Majeur
      □ qu'il faut lancer un déboggueur
                                                                                                                                   \square 1.5
                                                              16. Laquelle des analyses suivantes ne fait pas partie des
      □ qu'on veut changer alétoirement de fond d'écran
                                                                  étapes de la compilation :
                                                                                                                                   \Box 1
14. Si a et b sont deux variables de type :
                                                                    \square analyse sémantique
                                                                                                                                   \square 0.5
    struct toto_s
                                                                     \square analyse syntaxique
                                                                                                                                   \Box 0
    {
                                                                    \square analyse harmonique
      int n;
                                                                                                                             20. La virtualisation de la mémoire permet notamment de
      double x;
                                                                     \square analyse lexicale
                                                                                                                                 stocker des portions inactives de la mémoire de travail
    };
                                                                                                                                 sur le disque dur. Mais on perd :
                                                              17. Le code suivant :
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
                                                                    int i;
                                                                                                                                   □ certaines données de la mémoire de travail
                                                                    for (i = 1; i < 5; i = i + 1)
      \square (a.n == b.n) && (a.x == b.x)
                                                                                                                                   □ les fichiers du disque
      □ a == b
                                                                                                                                   \square en temps d'accès
                                                                        printf("%d ", i);
      \square a{n, x} == b{n, x}
                                                                                                                                   \square des processus
      \square a = b
                                                                   printf("\n");
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- Un fichier source est :
   □ un fichier texte qui sera traduit en instructions processeur
  - $\Box\,$ un document qui doit être protégé
  - ☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
  - $\square\,$  un document de référence du système
  - $\Box$  un document illisible pour les humains
- 2. Un programme en langage C doit comporter une et une seule définition de la fonction :
  - □ begin
  - $\square$  init
  - $\square$  main
  - $\square$  include
- 3. Soit la fonction g définie par :

```
int g(int a)
{
  printf("a = \n", %d);
  if (1 > 0)
  {
    return 5;
  }
  return 7;
```

Alors l'expression g(0) prendra la valeur :

- $\Box$  5
- $\Box$  7
- $\Box$  0
- 4. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 3; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...</pre>
```

- ☐ int char;
  ☐ char "c";
  ☐ char 'c';
  ☐ char c;
- 6. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :
  - □ n = carre(n);
    □ int carre(2);
    □ int n = carre();
    □ n = carre(int n);
- 7. Le code suivant :

 $\Box 01234$ 

 $\Box \ 4\ 3\ 2\ 1\ 0$ 

```
8. Le type des réels en C est :
      □ char
      \square real
      □ double
      \Box int
 9. Une variable booléenne est un variable :
      □ NaN (not a number, qui n'est pas un nombre)
      \square qui est vraie ou fausse
      ☐ jamais nulle
      \Box à la
quelle une valeur vient d'être affectée
      ☐ réelle positive
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      ☐ il risque d'afficher bonjour à la place de coucou
      \square il ne compile pas
      \square il comporte une boucle infinie
      □ il n'affiche rien
11. Si a et b sont deux variables de type :
    struct toto_s
      int n;
      double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
      \square (a.n == b.n) & (a.x == b.x)
```

□ a == b

 $\Box$  a = b

 $\square$  a{n, x} == b{n, x}

□ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur □ la division du programme en zones homogènes échoue □ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal □ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée  13. Sous unix (ou linux), la commande cd permet de : □ changer de répertoire courant □ jouer de la musique □ récupérer un programme arrêté avec la commande ab □ ouvir un bureau partagé (common desktop) □ détruire un fichier  14. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max' , que doit-on chercher dans le programme? □ une fonction appelée avant sa déclaration □ une fonction déclarée mais non définie □ une directive préprocesseur #include manquante	préalable, on utilise plutôt:    for(i=0;i<5;i=i+1)	☐ int afficher_date(date_s d); ☐ void afficher_date(struct date_s d);  19. Le code suivant:  int i; for (i = 4; i >= 0; i = i - 1) {     printf("%d ", i); } printf("\n");  affichera: ☐ 4 3 2 1 ☐ 4 3 2 1 0 ☐ 0 1 2 3 4 ☐ 1 2 3 4  20. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeuTRUE? ☐ (!A    B) ☐ (A == TRUE) && (B == TRUE) ☐ !(!A    B) == (A && !B)
☐ un désaccord entre la déclaration et la définition d'une fonction	du struct, on écrit :  □ struct date_s afficher_date(struct date_s	

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. L'écriture  $\underline{111}$  en binaire correspond au nombre naturel :
  - $\square$  3
  - □ 7
  - □ 8 □ 111
- 2. Pour l'extrait de programme suivant :

```
int i = 0;
int j = 0;
for (i = 0; i < 3; i = i + 1)
{
    for (j = 0; j < 2; j = j + 1)
    {
        printf("%d ", i);
    }
}
printf("\n");</pre>
```

- qu'est ce qui sera affiché?
  - □ 1 2 3 1 2
  - $\square$  0 1 2 0 1 2
  - $\square$  0 1 0 1 0 1
- $\Box$  0 0 1 1 2 2
- 3. Si a et b sont deux variables de type :

```
struct toto_s
{
   int n;
   double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition :

□ a = b
 □ a == b
 □ a{n, x} == b{n, x}
 □ (a.n == b.n) && (a.x == b.x)

- 4. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
  - ☐ #appart <stdlib.h>
  - ☐ #include <studio.h>
  - ☐ #include <stdio.h>
    ☐ #include <studlib.h>
- 5. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\square x = racine(x * x) - racine(x);
```

- $\square$  x 1 = racine(x);
- $\square$  x = racine(racine(x)\*racine(x));
- $\square$  x = racine(2/3);
- 6. Le type des réels en C est :
  - $\square$  real
  - $\square$  char
  - $\square$  int
  - □ double
- 7. Si cette erreur apparaît à la compilation :

erreur: conflicting types for 'max', que doiton chercher dans le programme?

- $\Box$  une fonction appelée avant sa déclaration
- ☐ un désaccord entre la déclaration et la définition d'une fonction
- □ une fonction déclarée mais non définie
- $\hfill \square$  une directive préprocesseur  $\#\mbox{include}$  man quante
- 8. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  ${\tt n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  ${\tt n}$ . Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
    scanf("%d", &n);
```

- Quelle est la condition cond :
- $\square$  (a<n) || (n>b)
- $\square$  (n<=a) && (n<=b)
- □ a<=n<=b
- $\square$  (a<=n) && (n<=b)
- 9. Soit la fonction f définie par :

```
int f(int a)
{
  printf("a = \n", %d);
  if (a > 0)
  {
    return f(a - 1) + 1;
  }
  return 4;
```

Alors l'expression f(1) prendra la valeur :

- $\Box$  0
- $\Box$  1
- $\Box$  5
- $\Box$  4
- 10. Quel est l'opérateur de différence en C :
  - □!
  - □ ! =
  - □ <>
  - $\square \neq$
- 11. Le code suivant :

```
int i;
for (i = 1; i < 5; i = i + 1)
{
    printf("%d ", i);
}
printf("\n");</pre>
```

- affichera:
- $\square$  4 3 2 1
- $\Box$  0 1 2 3 4
- $\square \ 1\ 2\ 3\ 4$
- $\Box$  4 3 2 1 0

12. Vous utilisez une boucle while quand :	15. Un registre du processeur est :	$\Box$ il ne compile pas
$\Box$ l'incrément de la variable de boucle n'est pas $1$	$\Box$ une unité de calcul spécialisée de l'ordinateur	$\Box$ il risque d'afficher bonjour à la place de coucou
$\square$ vous n'avez pas déclaré de fonction	$\Box$ un composant qui contient la liste des fichiers du	☐ il comporte une boucle infinie
□ vous avez déjà fait un for dans le même programme principal	système  une gamme de fréquence de fonctionnement du processeur	□ il n'affiche rien
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	une case mémoire interne au processeur qui sera manipulée directement lors des calculs	19. Un bit est:
13. Le code suivant :	16. Si cette erreur apparaît à la compilation :	$\Box$ la longueur d'un mot mémoire
int i;	error: expected ';' before '}' token que doit-	$\Box$ un battement d'horloge processeur
for $(i = 8; i > 0; i = i - 2)$	on chercher dans le programme?	$\hfill\Box$ l'instruction qui met fin à un programme
{     printf("%d ", i);	<ul><li>□ un point-virgule manquant</li><li>□ une accolade en trop</li></ul>	$\Box$ un chiffre binaire (0 ou 1)
}	□ un point-virgule en trop	20. Après exécution jusqu'à la ligne 14 du programme C
<pre>printf("\n");</pre>	une accolade manquante	10 int main() {
affichera:	17. Un enregistrement permet de grouper plusieurs valeurs	10 Int main() $\chi$ 11 int x = 5;
$\square$ 8 2	dans:	12
$\Box \ 8 \ 6 \ 4 \ 2 \ 0$	□ ses blocs	13 printf(" x = %d\n", 2);
$\Box \ 0\ 2\ 4\ 6\ 8$	$\square$ ses champs	14
$\square$ 8 6 4 2	□ ses cases	15 16 }
14. Laquelle de ces écritures correspond à la déclaration	$\square$ ses chants	10 )
d'une variable de type caractère en langage C?	18. Quel est le problème d'un programme comportant les	$\square$ le terminal affiche $5$
□ char 'c';	lignes suivantes?	$\Box$ le terminal affiche x = 5
☐ char "c";	while (1)	□ le terminal affiche "Faux"
□ char c;	<pre>1     printf("coucou\n");</pre>	
$\square$ int char;	}	$\Box$ le terminal affiche x = 2

Prénom:	Nom:	
N° INE :		

Licence 1 Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si cette erreur apparaît à la compilation : Undefined symbols :"\_prinft" ou référence indéfinie vers « prinft » que doiton chercher dans le programme?  $\square$  une faute de frappe dans un appel de fonction □ une variable non déclarée ☐ une directive préprocesseur #include manquante □ un caractère interdit en C 2. L'écriture 111 en binaire correspond au nombre naturel:  $\square$  8  $\Box$  7  $\square$  111  $\square$  3 3. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : □ dans lequel ces fonctions sont appelées dans le main  $\square$  un ordre quelconque □ alphabétique

□ dans lequel vous avez déclaré ces fonction 4. Vous utilisez une boucle while quand: ☐ l'incrément de la variable de boucle n'est pas 1 □ vous n'avez pas déclaré de fonction □ vous avez déjà fait un for dans le même programme principal  $\square$  vous ne connaissez pas le nombre d'itérations de

5. Laquelle de ces écritures correspond à la déclaration

d'une variable de type caractère en langage C?

la boucle à l'avance

□ char 'c';

☐ int char;

☐ char "c":

☐ char c:

6. Quels calculs peut-on programmer en programmation structurée? □ certains programmes sont de vrais plats de spaghetti  $\square$  il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine ☐ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée □ en programmation structurée on peut programmer tous les calculs programmables en langage machine 7. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\square$  3  $\Box$  4  $\square$  0 8. Sous unix (ou linux), la commande cd permet de : ☐ récupérer un programme arrêté avec la commande ab □ détruire un fichier □ ouvir un bureau partagé (common desktop) □ jouer de la musique □ changer de répertoire courant 9. Le code suivant : int i: for (i = 0; i < 7; i = i + 2){ printf("%d ", i);

printf("\n");

```
affichera:
     \Box 0246
     \Box 0123456
     \Box 02468
     \Box 0 1 2 3 4 5 6 7
10. Après exécution jusqu'à la ligne 14 du programme C:
 10
      int main() {
 11
            int x = 5;
 12
 13
            printf(" x = %d\n", 2);
 14
 15
 16
      }
      \square le terminal affiche 5
     \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
      \square le terminal affiche x = 5
11. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
     \square un désaccord entre la déclaration et la définition
         d'une fonction
     \square une fonction déclarée mais non définie
     ☐ une directive préprocesseur #include manquante
      \square une fonction appelée avant sa déclaration
12. Un bit est:
     \square un chiffre binaire (0 ou 1)
      ☐ l'instruction qui met fin à un programme
      □ un battement d'horloge processeur
      □ la longueur d'un mot mémoire
13. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle(double n):
     \square int factorielle(int x);
     ☐ int factorielle();
      □ struct int factorielle(int n):
```

```
14. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 7
     \Box 5
     \Box 0
15. Le code suivant :
    int somme = 0;
    int i;
    for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
    printf("%d", somme);
```

affichera :	□ 8
$\Box$ 42	$\Box$ 16
□ 1	$\Box$ 4
□ 6	$\Box$ 0
	18. Le type des réels en C est :
16. Pour déclarer un tableau d'entiers de taille 5, on peut	$\square$ double
utiliser l'instruction	$\square$ real
☐ int tab[] = 5;	$\Box$ int
☐ char tableau[5];	□ char
☐ int[] new tableau(5);	19. Quel est l'opérateur de différence en C :
☐ int toto[taille=5];	$\square \neq$
☐ int toto[5];	□ !=
17. Pour l'extrait de programme suivant :	□!
<pre>int produit = 0;</pre>	$\Box \Leftrightarrow$
int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)	20. Un enregistrement permet de grouper plusieurs valeur dans :
<pre>{     produit = produit * serie[i];</pre>	$\square$ ses champs
produit - produit * Serie[i], }	$\square$ ses blocs
<pre>printf("produit = %d", produit);</pre>	$\square$ ses chants
La valeur affichée est :	□ ses cases

т	•	-1
	acence	

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Pour déclarer une procédure afficher\_date qui prend en argument un struct date s et affiche le contenu du struct, on écrit :

□ void afficher\_date(struct date\_s d); ☐ int afficher\_date(date\_s d); □ void afficher\_date(date\_s d);

2. Pour l'extrait de programme suivant :

```
int produit = 0;
int serie[4] = \{2, 2, 2, 2\};
for (i = 0; i < 4; i = i + 1)
  produit = produit * serie[i];
printf("produit = %d", produit);
```

La valeur affichée est :

 $\square$  8

 $\Box$  0

 $\Box$  16

 $\Box$  4

3. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :

 $\square$  0.5

 $\Box$  0

 $\square$  1.5

 $\Box$  1

4. L'écriture 111 en binaire correspond au nombre naturel:

□ 111

 $\square$  7

 $\square$  8

 $\square$  3

5. Un registre du processeur est :

un composant qui contient la liste des fichiers of système	lυ
□ une case mémoire interne au processeur qui se manipulée directement lors des calculs	ra
$\Box$ une unité de calcul spécialisée de l'ordinateur	
□ une gamme de fréquence de fonctionnement or processeur	lu

□ struct date\_s afficher\_date(struct date\_s |d);6. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

$\square$ char c	;
------------------	---

☐ int char: □ char 'c':

☐ char "c";

7. Le code suivant :

```
int age = 20;
if (age < 18)
     printf("Mineur\n");
printf("Majeur\n");
affichera:
```

□ Majeur

□ Mineur

□ Mineur

Majeur

 $\square$  rien

8. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :

 $\square$  printf("x=%d et y=%d\n,x,y");

 $\square$  printf("x=%d et y=%d\n",x y);

 $\square$  printf("x=%x et y=%y\n");

9. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :

strı	ıct	int	factorielle(int	n);
int	fac	ctor	ielle();	

 $\square$  int factorielle(int x):

☐ int factorielle(double n);

10. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :

 $\square$  for(i=0;i<5;i=i+1)

☐ for(i=1;i<5;i=i+1)

 $\square$  for(i=1;i<=5;i=i+1)

 $\square$  for(i=0;i<=5;i=i+1)

11. Pour compiler un programme prog.c, on utilise la ligne de commande :

☐ gcc -Wall prog.c -o prog.exe

☐ gcc prog.c -o -Wall prog.exe

☐ gcc prog.exe -Wall -o prog.c

☐ gcc -Wall prog.exe -o prog.c

12. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

□ void saisie\_utilisateur(char c);

☐ saisie\_utilisateur(scanf(%d));

☐ int saisie\_utilisateur();

□ void saisie\_utilisateur(int n);

13. L'écriture 101 en binaire correspond au nombre naturel:

 $\Box$  5

 $\square$  101

 $\square$  3  $\Box$  4

14. Un fichier source est:

□ un fichier texte qui sera traduit en instructions processeur

 $\square$  un fichier que l'ont doit citer dans les documents produits sur l'ordinateur

 $\square$  un document illisible pour les humains

□ un document qui doit être protégé

□ un document de référence du système

5. Si racine est une fonction prenant en entrée un réel	17. Une variable booléenne est un variable :	19. On considère deux variables booléennes A et B initia-	
et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect	☐ réelle positive	lisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur	
d'écrire :	$\Box$ qui est v raie ou fausse	TRUE?	
$\square$ x = racine(x * x) - racine(x);	$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)	□ (!A    B)	
$\square$ x - 1 = racine(x);	$\Box$ à la quelle une valeur vient d'être affectée	$\square$ (A == TRUE) && (B == TRUE)	
<pre>     x = racine(racine(x)*racine(x));</pre>	$\square$ jamais nulle	□ A && B	
$\square$ x = racine(2/3);	18. Si carre est une fonction prenant en entrée un en-	$\square !(!A    B) == (A \&\& !B)$	
6. Vous utilisez une boucle while quand :	tier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct	20. Quelle étape de la compilation vient d'échouer lors- qu'on a un message comme celui-ci : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »	
□ vous avez déjà fait un for dans le même programme principal	d'écrire :		
$\hfill \square$ vous n'avez pas déclaré de fonction	$\square$ n = carre(n);	$\square$ l'analyse sémantique	
□ vous ne connaissez pas le nombre d'itérations de	$\square$ int n = carre();	$\square$ l'analyse des entrées clavier	
la boucle à l'avance	☐ int carre(2);	☐ l'édition de liens	
$\Box$ l'incrément de la variable de boucle n'est pas $1$	□ n = carre(int n);	$\Box$ l'analyse harmonique	
'	'		

## Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction f définie par :
   int f(int a)
   ₹
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \Box 0
2. Après exécution jusqu'à la ligne 14 du programme C:
 10
      int main() {
 11
           int x = 5;
 12
           printf(" x = %d\n", 2);
 13
 14
 15
 16
     }
     □ le terminal affiche "Faux"
     \square le terminal affiche x = 5
     □ le terminal affiche 5
     \square le terminal affiche x = 2
3. Soit le programme principal suivant :
   int main()
    int a = 3:
    int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
```

```
int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
4. Si factorielle est une fonction prenant en entrée un
  entier et renvoyant un entier, il est correct d'écrire :
    □ printf("%d", factorielle(n));
    \square n = factorielle();
    ☐ int factorielle(int 2);
    \square n = factorielle(p, q);
5. Quels calculs peut-on programmer en programmation
  structurée?
    □ en programmation structurée on peut program-
       mer tous les calculs programmables en langage
       machine
    ☐ il y a des calculs programmables en langage ma-
       chine et qui ne sont pas programmables en pro-
       grammation structurée
    \square il y a des calculs programmables en programma-
       tion structurée qui ne sont pas programmables en
       langage machine
    □ certains programmes sont de vrais plats de spa-
       ghetti
6. Le code suivant :
   int age = 20;
   if (age < 18)
   {
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
```

```
affichera:
     □ rien
     □ Majeur
     □ Mineur
     □ Mineur
        Majeur
 7. Au début de la fonction main() on place le code :
     char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
    printf("\n");
    Alors l'affichage sera:
     □ cccccc
     □ A
     ☐ ABCDEF
     □i
 8. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
     □ dans lequel vous avez déclaré ces fonction
     □ alphabétique
     □ un ordre quelconque
     □ dans lequel ces fonctions sont appelées dans le
        main
9. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     □ créer un fichier texte
     □ changer de répertoire courant
     □ ouvrir un fichier texte
10. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     \Box char c;
     ☐ char "c";
     □ char 'c';
      ☐ int char;
```

```
14. Si racine est une fonction prenant en entrée un réel
11. Le code suivant :
                                                                                                                          18. Pour déclarer un tableau d'entiers de taille 5, on peut
                                                                 et renvoyant la racine carrée de cet réel, et que x est
                                                                                                                              utiliser l'instruction
     int i:
                                                                 une variable réelle définie et initialisée, il est incorrect
     for (i = 4: i > 0: i = i - 1)
                                                                                                                                \square int[] new tableau(5);
                                                                 d'écrire :
                                                                                                                                ☐ int toto[taille=5];
         printf("%d ", i);
                                                                   \square x = racine(x * x) - racine(x);
                                                                                                                                \square int toto[5]:
                                                                   \square x = racine(racine(x)*racine(x)):
                                                                                                                                □ char tableau[5];
     printf("\n");
                                                                   \square x = racine(2/3);
                                                                                                                                \square int tab[] = 5;
    affichera:
                                                                   \square x - 1 = racine(x);
      \Box 01234
                                                                                                                          19. Pour l'extrait de programme suivant :
      \Box 43210
                                                             15. Quelle étape de la compilation vient d'échouer lors-
                                                                                                                                int produit = 1;
                                                                 qu'on a un message comme celui-ci :
      \square 4 3 2 1
                                                                                                                                int serie[4] = \{2, 2, 2, 2\};
                                                                 Undefined symbols :"_prinft" ou
      \Box 0123
                                                                                                                                for (i = 0; i < 4; i = i + 1)
                                                                 référence indéfinie vers « prinft »
12. Si cette erreur apparaît à la compilation :
                                                                                                                                   produit = produit * serie[i];
                                                                   □ l'édition de liens
    erreur: conflicting types for 'max', que doit-
    on chercher dans le programme?
                                                                   \square l'analyse des entrées clavier
                                                                                                                                printf("produit = %d", produit);
      \square un désaccord entre la déclaration et la définition
                                                                   ☐ l'analyse sémantique
         d'une fonction
                                                                                                                              La valeur affichée est :
                                                                   \square l'analyse harmonique
      ☐ une fonction appelée avant sa déclaration
                                                                                                                                \Box 16
      \square une fonction déclarée mais non définie
                                                             16. Si cet avertissement apparaît à la compilation :
                                                                                                                                \Box 0
                                                                 warning: implicit declaration of function 'max'
      □ une directive préprocesseur #include manquante
                                                                                                                                \Box 4
                                                                 , que doit-on chercher dans le programme?
13. Pour l'extrait de programme suivant :
                                                                                                                                \square 8
                                                                   ☐ une fonction appelée avant sa déclaration
      int somme = 0;
                                                                                                                          20. Quel est le problème d'un programme comportant les
      for (i = 0; i < 5; i = i + 1)
                                                                   □ un désaccord entre la déclaration et la définition
                                                                                                                              lignes suivantes?
                                                                      d'une fonction
        somme = somme + i;
                                                                                                                              while (1)
                                                                   ☐ une directive préprocesseur #include manquante
        i = i + 1; /* attention ! */
                                                                   □ une fonction déclarée mais non définie
                                                                                                                                printf("coucou\n");
      printf("somme = %d",somme);
                                                             17. Le type des réels en C est :
    La valeur de somme affichée est :
                                                                                                                                □ il n'affiche rien
                                                                   \square char
      \Box 10
                                                                                                                                \square il comporte une boucle infinie
                                                                   ☐ double
      \Box 0
                                                                                                                                ☐ il risque d'afficher bonjour à la place de coucou
                                                                   □ int
      \Box 6
                                                                                                                                \square il ne compile pas
                                                                   \square real
      \square 15
```

 ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ıce	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

1	reponse lausse. Duree: 20 minutes.
l.	Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :
	☐ struct int factorielle(int n);
	☐ int factorielle(int x);
	☐ int factorielle(double n);
	☐ int factorielle();
2.	Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
	☐ int char;
	□ char "c";
	□ char c;
	☐ char 'c';
3.	Un fichier source est :
	$\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\Box$ un document illisible pour les humains
	$\Box$ un document qui doit être protégé
	$\square$ un document de référence du système
1.	Après exécution du programme :
4.	l lecture 8 r0 valeur 3 r1 mult r1 r0 valeur 1 r2 add r2 r0 ceriture r0 8 stop 8 5
	$\Box$ la case mémoire 8 contiendra 16
	$\square$ le terminal affiche 8
	$\square$ le bus explose
	□ la case mémoire 8 contiendra 0

```
5. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses blocs
     \square ses chants
     \square ses cases
6. Une variable booléenne est un variable :
     □ NaN (not a number, qui n'est pas un nombre)
    □ qui est vraie ou fausse
     ☐ jamais nulle
     □ à laquelle une valeur vient d'être affectée
     ☐ réelle positive
7. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ une fonction déclarée mais non définie
     ☐ une directive préprocesseur #include manquante
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
8. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
   Alors l'affichage sera:
     \sqcap C
     ПЪ
     □ A
     □В
9. Un registre du processeur est :
     □ une unité de calcul spécialisée de l'ordinateur
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     □ un composant qui contient la liste des fichiers du
        système
     □ une gamme de fréquence de fonctionnement du
        processeur
```

```
10. Le code suivant :
    int age = 20;
    if (age < 18)
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
   affichera:
     □ Majeur
     □ Mineur
     □ rien
     ☐ Mineur
        Majeur
11. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit:
     □ void saisie_utilisateur(char c);
     ☐ int saisie_utilisateur();
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
12. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
    int a = 0;
    int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
       scanf("%d", &n);
   Quelle est la condition cond :
     \square (a<n) || (n>b)
     □ a<=n<=b
     □ (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
```

13. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-	☐ le terminal affiche 5 ☐ le terminal affiche "Faux"	$\Box a\{n, x\} == b\{n, x\}$
on chercher dans le programme?  ☐ un point-virgule manquant	16. Vous utilisez une boucle while quand:	$\Box$ a = b $\Box$ (a.n == b.n) && (a.x == b.x)
$\Box$ une accolade manquante	☐ l'incrément de la variable de boucle n'est pas 1	□ a == b
☐ un point-virgule en trop	□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	19. Afin de représenter la taille d'un tableau, définir un constante symbolique N valant 3.
une accolade en trop	$\hfill \square$ vous n'avez pas déclaré de fonction	#define taille = 3
14. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la com-	□ vous avez déjà fait un <b>for</b> dans le même pro- gramme principal	$\square$ #define N 3
mande: □ kwrite TP4	17. Le bus système sert à :	☐ #define N = 3
☐ mkdir TP4	☐ transporter les processus du tourniquet au processeur	☐ #define taille = N  20. Le code suivant :
$\square$ yppasswd	☐ Arriver à l'heure en cours	int i;
□ new TP4	☐ Écrire des données sur le dique dur	for (i = 8; i > 0; i = i - 2)
15. Après exécution jusqu'à la ligne 14 du programme C : 10 int main() {	☐ Transférer des données et intructions entre processeur et mémoire	<pre>t     printf("%d ", i);</pre>
int $x = 5$ ;	18. Si a et b sont deux variables de type :	<pre>} printf("\n");</pre>
12 13 printf(" x = %d\n", 2);	struct toto_s {	affichera:
14 15	<pre>int n; double x;</pre>	$\square$ 8 6 4 2
16 }	};	
$\Box \text{ le terminal affiche } \mathbf{x} = 2$	Alors pour tester l'égalité de a et de b on utilise la	
$\square$ le terminal affiche $x = 5$	condition:	

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Sous unix (ou linux), la commande 1s permet de : □ compiler un programme □ voir des clips musicaux  $\square$  afficher la liste de fichiers contenus dans un répertoire  $\square$  afficher le contenu d'un fichier texte 2. Au début de la fonction main() on place le code : char i; for  $(i = 'A'; i \le 'F'; i = i + 1)$ printf("%c", i); printf("\n"); Alors l'affichage sera:  $\square$  A □ cccccc Пі ☐ ABCDEF 3. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n"); printf("Majeur\n"); affichera: □ Mineur Majeur □ Majeur □ rien ☐ Mineur

```
4. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
     □ dans lequel ces fonctions sont appelées dans le
       main
     □ dans lequel vous avez déclaré ces fonction
     □ un ordre quelconque
     □ alphabétique
5. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
6. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ une fonction déclarée mais non définie
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
     ☐ une directive préprocesseur #include manquante
7. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     \square il ne compile pas
     □ il n'affiche rien
     \square il comporte une boucle infinie
     ☐ il risque d'afficher bonjour à la place de coucou
8. Le langage C est un langage
     □ composé
    □ lu, écrit, parlé
     □ compilé
```

□ interprété

```
9. Le code suivant :
     int age = 18;
     if (age < 18)
     {
          printf("Mineur\n");
     else
     {
          printf("Majeur\n");
     }
    affichera:
      ☐ Mineur
      □ Mineur
        Majeur
      □ Majeur
      □ rien
10. L'ordonnancement par tourniquet permet :
      ☐ d'entretenir l'illusion que les processus tournent
         en parallèle
      □ d'afficher des ronds colorés à l'écran
      ☐ de ne pas perdre de temps avec la commutation
         de contexte
      \square de doubler la mémoire disponible
11. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(x * x) - racine(x):
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
      \square x = racine(2/3);
12. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20:
```

int n:

```
scanf("%d", &n);
                                                             15. Le code suivant :
     while(cond)
                                                                  int i:
                                                                  for (i = 4; i > 0; i = i - 1)
       scanf("%d", &n);
                                                                       printf("%d ", i);
    Quelle est la condition cond:
      \square (a<n) || (n>b)
                                                                  printf("\n");
      \square (a<=n) && (n<=b)
                                                                 affichera:
      \square (n<=a) && (n<=b)
      □ a<=n<=b
                                                                   \Box 01234
13. Soit le programme principal suivant :
                                                                   \square 0 1 2 3
   int main()
                                                                   \square 4 3 2 1
   {
     int a = 3;
                                                                   \Box \ 4\ 3\ 2\ 1\ 0
     int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                             16. Le code suivant :
     return EXIT_SUCCESS;
                                                                  int i;
                                                                  for (i = 1; i < 5; i = i + 1)
   appelant la fonction f ainsi définie :
   int f(int a, int b)
                                                                       printf("%d ", i);
      a = a + b;
                                                                  printf("\n");
      return a;
                                                                 affichera:
   L'affichage dans le main est le suivant :
                                                                   \square 4 3 2 1
      \Box f(a,b)=13, a=8, b=5
                                                                   \Box 01234
     \Box f(a,b)=8, a=8, b=5
                                                                   \square 1 2 3 4
     \Box f(a,b)=8, a=3, b=5
                                                                   \Box \ 4\ 3\ 2\ 1\ 0
      \Box f(3,5)=8, a=3, b=5
14. Si cet avertissement apparaît à la compilation :
                                                             17. L'écriture 111 en binaire correspond au nombre natu-
   warning: implicit declaration of function 'max
                                                                 rel:
   , que doit-on chercher dans le programme ?
                                                                   \square 111
      ☐ une fonction appelée avant sa déclaration
     □ une fonction déclarée mais non définie
                                                                   \square 8
     \square un désaccord entre la déclaration et la définition
                                                                   \square 3
         d'une fonction
                                                                   \Box 7
     ☐ une directive préprocesseur #include manquante
```

18. Quels calculs peut-on programmer en programmation structurée? □ en programmation structurée on peut programmer tous les calculs programmables en langage machine  $\square$  certains programmes sont de vrais plats de spaghetti  $\square$  il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine ☐ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée 19. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0){ return f(a - 1) + 1; return 4; Alors l'expression f(1) prendra la valeur :  $\Box$  0  $\Box$  4  $\Box$  5  $\Box$  1 20. Vous utilisez une boucle while quand: □ l'incrément de la variable de boucle n'est pas 1 □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction f définie par :
  int f(int a)
  {
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1:
     return 4;
  Alors l'expression f(1) prendra la valeur :
     \Box 4
     \Box 0
     \Box 1
     \square 5
2. Un programme en langage C doit comporter une et une
  seule définition de la fonction :
     \Box init
     \square include
     \square main
     □ begin
3. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
  affichera:
     ☐ Mineur
        Majeur
     □ Majeur
     \square rien
```

□ Mineur

```
4. Vous utilisez une boucle while quand :
    ☐ l'incrément de la variable de boucle n'est pas 1
    □ vous avez déjà fait un for dans le même pro-
       gramme principal
    \square vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
    □ vous n'avez pas déclaré de fonction
5. Quelle étape de la compilation vient d'échouer lors-
  qu'on a un message comme celui-ci :
  Undefined symbols : "_prinft" ou
  référence indéfinie vers « prinft »
    □ l'édition de liens
    ☐ l'analyse sémantique
    ☐ l'analyse des entrées clavier
    ☐ l'analyse harmonique
6. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise:
    ☐ int afficher_menu(int char);
    □ void afficher_menu();
    ☐ double afficher_menu();
    ☐ int afficher_menu();
    ☐ char afficher_menu(printf("menu"));
7. Si le code :
  struct toto_s
     int n;
     double x;
  };
  précède la fonction main(), alors on peut écrire en
  début de main() :
    □ struct toto_s toto;
    \Box toto_s n, x;
    \Box toto_s struct z = {3, 0.5};
    \Box int toto.n = 3;
```

 $\Box$  int struct toto\_s = {3, -1e10};

```
8. Une segmentation fault est une erreur qui survient
   lorsque:
      □ le programme tente d'afficher des caractères sur
         une ligne qui va au delà de la largeur de la fenêtre
         du terminal
      □ le programme source a été enregistré sur le disque
         dur en plusieurs morceaux et l'un d'entre eux ne
         peut pas être chargé par le compilateur
      \square le programme tente d'accèder à une partie de la
         mémoire qui ne lui est pas réservée
      □ la division du programme en zones homogènes
         échoue
 9. Soit la fonction g définie par :
   int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5:
      }
      return 7;
   Alors l'expression g(0) prendra la valeur :
      \Box 5
      \Box 7
      \Box 0
10. Le code suivant :
     int i:
     for (i = 4; i > 0; i = i - 1)
          printf("%d ", i);
    printf("\n");
    affichera:
      \Box 01234
      \Box \ 4\ 3\ 2\ 1\ 0
      \square 4 3 2 1
      \square 0 1 2 3
```

```
18. Pour déclarer une fonction saisie_utilisateur qui
11. Après exécution jusqu'à la ligne 15 du programme C:
                                                                 □ en parallèle, chacun dans un registre
                                                                                                                           demande à l'utilisateur d'entrer un entier au clavier et
                                                                 □ tour à tour, un petit peu à chaque fois
       int main() {
                                                                                                                           renvoie cet entier on écrit :
                                                                 \square tous ensemble
 11
           int x = 5;
 12
           int y;
                                                                                                                             ☐ int saisie_utilisateur();
                                                           15. Pour afficher à l'aide de printf("%d\n",tab[i]);
 13
                                                               le contenu d'un tableau de 5 entiers initialisé au
                                                                                                                             □ void saisie_utilisateur(char c);
 14
           y = x;
                                                               préalable, on utilise plutôt :
                                                                                                                             ☐ saisie utilisateur(scanf(%d)):
 15
                                                                 \square for(i=1;i<5;i=i+1)
 16
                                                                                                                             □ void saisie_utilisateur(int n);
                                                                 \square for(i=1;i<=5;i=i+1)
 17
                                                                 \Box for(i=0;i<5;i=i+1)
                                                                                                                       19. Soit un programme contenant les lignes suivantes :
     \square la variable x vaut 0
                                                                 \square for(i=0;i<=5;i=i+1)
     □ le programme affiche "Faux"
                                                                                                                            int i = 0;
                                                           16. Soit le programme principal suivant :
     \square la variable y vaut 5
                                                                                                                            int j = 0;
                                                                int main()
                                                                                                                            for (i = 0; i < 2; i = i + 1)
     \square la variable x vaut 5 et la variable y vaut 0
                                                               {
                                                                int a = 3;
12. Si cet avertissement apparaît à la compilation :
                                                                                                                                 for (j = 0; j < 3; j = j + 1)
                                                                int b = 5;
   warning: implicit declaration of function 'max'
                                                                printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                                     printf("%d ", i);
   , que doit-on chercher dans le programme?
                                                                return EXIT_SUCCESS;
                                                                                                                                 }
     ☐ une directive préprocesseur #include manquante
                                                               }
                                                                                                                            }
     □ une fonction déclarée mais non définie
                                                               appelant la fonction f ainsi définie :
     ☐ une fonction appelée avant sa déclaration
                                                               int f(int a, int b)
                                                                                                                           qu'est ce qui sera affiché?
     □ un désaccord entre la déclaration et la définition
                                                                  a = a + b;
        d'une fonction
                                                                                                                             \Box 0 1 0 1 0 1 0 1
                                                                 return a;
13. Pour déclarer une procédure afficher_date qui prend
                                                                                                                             \Box 0 1 2 0 1 2
                                                               }
   en argument un struct date_s et affiche le contenu
                                                               L'affichage dans le main est le suivant :
                                                                                                                             \Box 1 2 1 2 3
   du struct, on écrit :
                                                                 \Box f(a,b)=8, a=3, b=5
                                                                                                                             \Box 0 0 0 1 1 1
     ☐ int afficher_date(date_s d);
                                                                 \Box f(a,b)=13, a=8, b=5
     □ void afficher_date(struct date_s d);
                                                                                                                       20. Si x est une variable réelle (de type double) alors
                                                                 \Box f(a,b)=8, a=8, b=5
                                                                                                                           x = 3/2 lui affecte la valeur :
     □ void afficher_date(date_s d);
                                                                 \Box f(3,5)=8, a=3, b=5
     \square struct date_s afficher_date(struct date_s |d) i7. Sur unix (ou linux), la commande mkdir permet de :
                                                                                                                             \square 0.5
14. Sur un ordinateur avec un seul processeur, habituelle-
                                                                 □ créer un fichier texte
                                                                                                                             \square 1.5
   ment les processus sont exécutés :
                                                                 □ créer un répertoire
                                                                                                                             \Box 0
                                                                 □ changer de répertoire courant
      □ chacun son tour, après que le processus précédent
                                                                                                                             \Box 1
        a terminé
                                                                 □ ouvrir un fichier texte
```

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0,5 points par réponse fausse. Durée: 20 minutes.

1. Si cet avertissement apparaît à la compilation: warning: implicit declaration of function 'man, que doit-on chercher dans le programme?

Si cet avertissement apparaît à la compilation :

warning: implicit declaration of function 'max',

que doit-on chercher dans le programme?

une directive préprocesseur #include manquante

une fonction déclarée mais non définie

un désaccord entre la déclaration et la définition
d'une fonction

une fonction appelée avant sa déclaration

2. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?une fonction appelée avant sa déclaration

☐ une fonction appelee avant sa declaration ☐ une fonction déclarée mais non définie

□ une directive préprocesseur #include manquante
 □ un désaccord entre la déclaration et la définition d'une fonction

3. Quel est le problème d'un programme comportant les lignes suivantes?

while (1)
{
 printf("coucou\n");
}

□ il risque d'afficher bonjour à

 $\Box\,$ il risque d'afficher bonjour à la place de coucou

 $\hfill\Box$  il comporte une boucle infinie

 $\Box$ il n'affiche rien

 $\Box\,$ il ne compile pas

4. Pour l'extrait de programme suivant :

```
int i;
int j;
for(i=4;i>0;i=i-1)
{
    for(j=i;j<6;j=j+1)
    {
       printf("*");
    }
    printf(" ");
}</pre>
```

qu'est ce qui sera affiché? \*\*\*\* \*\*\*\* \*\*\*\* 5. Une variable booléenne est un variable : □ qui est vraie ou fausse □ à laquelle une valeur vient d'être affectée ☐ réelle positive □ NaN (not a number, qui n'est pas un nombre)  $\square$  jamais nulle 6. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire : ☐ printf("%d", factorielle(n));  $\square$  n = factorielle(); ☐ int factorielle(int 2);  $\square$  n = factorielle(p, q); 7. Quels calculs peut-on programmer en programmation structurée?  $\square$  il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée □ en programmation structurée on peut programmer tous les calculs programmables en langage machine  $\square$  il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine

□ certains programmes sont de vrais plats de spaghetti

8. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :

□ int pgcd(int x, y);
□ int pgcd(int x, int x);
□ void pgcd(int x, int y);
□ int pgcd(int y, int x);

9. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :

1
0
1.5
0.5

10. Le bus système sert à :

transporter les processus du tourniquet au processeur
Écrire des données sur le dique dur
Transférer des données et intructions entre processeur et mémoire
Arriver à l'heure en cours

int i;
for (i = 4; i > 0; i = i - 1)
{
 printf("%d ", i);

}
printf("\n");

 $\Box$  4 3 2 1  $\Box$  4 3 2 1 0

affichera:

11. Le code suivant :

12. Après exécution jusqu'à la ligne 15 du programme C :

10 ...
11 int main() {
12 int x = 5;
13
14 x = 3 \* x + 1;
15
16 ...
17 }

 $\Box\,$ le programme affiche \*\*\*\*

 $\Box$ la variable x vaut 16

□ la variable x vaut  $-\frac{1}{2}$  □ le programme affiche x

13. Sur unix (ou linux), la commande mkdir permet de :	<pre>printf("Mineur\n");</pre>	$\square$ 3
$\square$ créer un répertoire	}	
$\square$ ouvrir un fichier texte	else s	$\square$ 4
$\Box$ créer un fichier texte	<pre>printf("Majeur\n");</pre>	18. Un enregistrement permet de grouper plusieurs valeurs
$\Box$ changer de répertoire courant	}	dans:
14. L'écriture $\underline{101}$ en binaire correspond au nombre natu-		□ ses chants
rel:	affichera:	☐ ses cases
$\Box$ 5	☐ Mineur	□ ses champs
$\Box$ 4	Majeur	□ ses blocs
$\square$ 3	☐ Mineur	19. Lorsqu'un programme utilise printf ou scanf il faut
□ 101	□ rien	qu'il contienne l'instruction préprocesseur :
15. Avant de faire appel à une fonction il est nécessaire	☐ Majeur	☐ #include <studio.h></studio.h>
de :	17. Soit la fonction f définie par :	#include <stdio.h></stdio.h>
□ avoir défini une constante symbolique de la taille de cette fonction	int f(int a)	#appart <stdlib.h></stdlib.h>
□ l'avoir déclarée et définie	<pre>1     printf("a = \n", %d);</pre>	#include <studlib.h></studlib.h>
☐ l'avoir définie	if (a > 0)	
☐ l'avoir déclarée	{	20. Un bit est:
16. Le code suivant :	return 3;	☐ la longueur d'un mot mémoire
	return 4;	$\square$ un battement d'horloge processeur
int age = 18; if (age < 18)	}	$\square$ l'instruction qui met fin à un programme
{	Alors l'expression f(0) prendra la valeur :	$\square$ un chiffre binaire (0 ou 1)
	•	•

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 2; j = j + 1)
             printf("%d ", i);
        }
    }
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1
     \Box 0 0 1 1 2 2
     \Box 1 2 3 1 2
2. Un registre du processeur est :
     □ un composant qui contient la liste des fichiers du
        système
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     □ une gamme de fréquence de fonctionnement du
        processeur
     □ une unité de calcul spécialisée de l'ordinateur
3. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \square 8
     \square 3
     \Box 7
4. Le code suivant :
    int age = 18;
    if (age < 18)
        printf("Mineur\n"):
```

```
else
    {
        printf("Majeur\n");
    }
   affichera:
     □ Mineur
     □ rien
     □ Mineur
        Majeur
     □ Majeur
5. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ une directive préprocesseur #include manquante
    ☐ une fonction appelée avant sa déclaration
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     \square une fonction déclarée mais non définie
6. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square void exposant(double x^n);
    \square int exposant(double n, int x);
     \square double exposant(double x, int n);
    \square exposant(double x, int n, int r);
7. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2:
    printf("%c\n", b);
   Alors l'affichage sera :
     □ A
     \Box C
     □ b
     ПВ
```

```
8. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
    avec la commande :
      \square printf("x=%d et y=%d\n,x,y");
     \Box printf("x=%x et y=%y\n");
     \Box printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n",x y);
 9. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      □ une directive préprocesseur #include manquante
      \square une fonction appelée avant sa déclaration
      □ une fonction déclarée mais non définie
      \square un désaccord entre la déclaration et la définition
         d'une fonction
10. Un fichier source est:
      □ un document illisible pour les humains
      □ un fichier texte qui sera traduit en instructions
         processeur
      □ un document de référence du système
     □ un document qui doit être protégé
      □ un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
11. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square int pgcd(2);
     \square n = pgcd(n, 3);
     \square int n = pgcd();
     \square n = pgcd(int p, int q);
12. Un bit est:
      \square un chiffre binaire (0 ou 1)
      □ un battement d'horloge processeur
      ☐ l'instruction qui met fin à un programme
      □ la longueur d'un mot mémoire
```

```
15. Vous avez déclaré préalablement un ensemble de fonc-
13. Soit le programme principal suivant :
                                                                                                                             ☐ int saisie_utilisateur();
                                                               tions utilisées par votre programme principal. L'ordre
   int main()
                                                                                                                             □ void saisie_utilisateur(char c);
                                                               dans lequel vous devez maintenant définir ces fonctions
   {
                                                               est l'ordre :
                                                                                                                             ☐ saisie_utilisateur(scanf(%d));
    int a = 3;
                                                                 □ un ordre quelconque
     int b = 5;
                                                                                                                       19. Laquelle de ces écritures correspond à la déclaration
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                 □ alphabétique
                                                                                                                           d'une variable de type caractère en langage C?
    return EXIT_SUCCESS;
                                                                 □ dans lequel ces fonctions sont appelées dans le
                                                                                                                             ☐ char "c";
                                                                    main
   appelant la fonction f ainsi définie :
                                                                 \square dans lequel vous avez déclaré ces fonction
                                                                                                                             ☐ int char;
   int f(int a, int b)
                                                           16. Avant de faire appel à une fonction il est nécessaire
                                                                                                                             ☐ char c:
   {
                                                               de:
      a = a + b;
                                                                                                                             □ char 'c';
                                                                 □ l'avoir déclarée
      return a;
                                                                 ☐ l'avoir déclarée et définie
                                                                                                                       20. Le code suivant :
                                                                 □ avoir défini une constante symbolique de la taille
   L'affichage dans le main est le suivant :
                                                                                                                            int i;
                                                                    de cette fonction
     \Box f(a,b)=8, a=8, b=5
                                                                                                                            for (i = 0; i < 5; i = i + 1)
                                                                 ☐ l'avoir définie
     \Box f(a,b)=13, a=8, b=5
                                                           17. Lorsqu'un programme utilise printf ou scanf il faut
     \Box f(3,5)=8, a=3, b=5
                                                                                                                                 printf("%d ", i);
                                                               qu'il contienne l'instruction préprocesseur :
     \Box f(a,b)=8, a=3, b=5
                                                                                                                            printf("\n");
                                                                 ☐ #include <studio.h>
14. Si carre est une fonction prenant en entrée un en-
                                                                 ☐ #include <studlib.h>
   tier et renvoyant le carré de cet entier, et que n est
                                                                                                                           affichera:
   une variable entière définie et initialisée, il est correct
                                                                 ☐ #appart <stdlib.h>
   d'écrire :
                                                                                                                             \Box 0123
                                                                 ☐ #include <stdio.h>
     \square int carre(2);
                                                                                                                             \square 4 3 2 1
                                                           18. Pour déclarer une fonction saisie_utilisateur qui
     \square int n = carre();
                                                               demande à l'utilisateur d'entrer un entier au clavier et
                                                                                                                             \Box 01234
     \square n = carre(int n);
                                                               renvoie cet entier on écrit :
                                                                                                                             \Box 43210
     \square n = carre(n);
                                                                 □ void saisie_utilisateur(int n);
```

т	•	-1
	acence	

17

}

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

- 1. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction
  - ☐ char tableau[5];
    ☐ int[] new tableau(5);
  - $\Box$  int tab[] = 5;
  - $\square$  int toto[taille=5];
  - $\square$  int toto[5];
- 2. Après exécution jusqu'à la ligne 15 du programme C :

```
10 ...
11 int main() {
12 int x = 5;
13
14 x = 3 * x + 1;
15
16 ...
```

- □ la variable x vaut 16
- $\Box$  le programme affiche x
- $\square$  la variable x vaut  $-\frac{1}{2}$
- $\square$  le programme affiche \*\*\*\*
- 3. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?
  - $\Box$  un désaccord entre la déclaration et la définition d'une fonction
  - $\Box\,$ une fonction appelée avant sa déclaration
  - $\Box\,$  une directive préprocesseur  $\# {\tt include}$  man quante
  - $\Box$  une fonction déclarée mais non définie
- 4. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :

```
□ printf("Valeur de n ? %d\n", n);□ scanf("%d", &n);□ un débogueur
```

```
\Box printf("Valeur de n ? %g\n", n);
```

```
5. Le langage C est un langage
    □ lu, écrit, parlé
    □ composé
    □ interprété
    □ compilé
6. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
       somme = somme + i;
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
  La valeur de somme affichée est :
    \square 15
    \square 10
    \Box 0
    \Box 6
7. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 2; j = j + 1)
             printf("%d ", i);
        }
    }
    printf("\n");
   qu'est ce qui sera affiché?
    \Box 0 1 2 0 1 2
    \Box 0 1 0 1 0 1
```

 $\Box$  1 2 3 1 2

 $\Box$  0 0 1 1 2 2

```
8. Soit le programme principal suivant :
   int main()
     int a = 3;
     int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a. int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square int exposant(double n, int x);
     \square exposant(double x, int n, int r);
     \square double exposant(double x, int n);
     □ void exposant(double x^n);
10. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle(double n):
     ☐ struct int factorielle(int n);
     ☐ int factorielle();
     \Box int factorielle(int x);
11. Sous unix (ou linux), la commande cd permet de :
     □ ouvir un bureau partagé (common desktop)
     □ jouer de la musique
     □ changer de répertoire courant
     \square détruire un fichier
     ☐ récupérer un programme arrêté avec la commande
        ab
```

```
12. Pour l'extrait de programme suivant :
                                                                    \Box 0
                                                                                                                                   \square 3
                                                                    \square 7
                                                                                                                                   \Box 4
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
                                                                    \Box 5
                                                                                                                                   \Box 0
      for (i = 0; i < 4; i = i + 1)
                                                              15. Sur un ordinateur avec un seul processeur, habituelle-
                                                                                                                            18. Le code suivant :
                                                                  ment les processus sont exécutés :
         somme = somme + serie[i];
                                                                                                                                  int i;
                                                                    □ en parallèle, chacun dans un registre
                                                                                                                                  for (i = 4; i \ge 0; i = i - 1)
      printf("somme = %d",somme);
                                                                    □ tour à tour, un petit peu à chaque fois
                                                                                                                                      printf("%d ", i);
   La valeur de somme affichée est :
                                                                    \square tous ensemble
                                                                                                                                  }
      \square 3
                                                                    □ chacun son tour, après que le processus précédent
                                                                                                                                 printf("\n");
                                                                       a terminé
      \square 20
                                                                                                                                affichera:
                                                              16. Vous avez déclaré préalablement un ensemble de fonc-
      \Box 16
                                                                                                                                  \Box \ 4\ 3\ 2\ 1\ 0
                                                                  tions utilisées par votre programme principal. L'ordre
      \Box 6
                                                                  dans lequel vous devez maintenant définir ces fonctions
                                                                                                                                  \square 1 2 3 4
13. Si factorielle est une fonction prenant en entrée un
                                                                  est l'ordre :
                                                                                                                                  \Box 01234
   entier et renvoyant un entier, il est correct d'écrire :
                                                                    \square dans lequel vous avez déclaré ces fonction
                                                                                                                                  \square 4 3 2 1
      \square n = factorielle():
                                                                    □ un ordre quelconque
                                                                                                                            19. Pour afficher à l'aide de printf("%d\n",tab[i]);
      ☐ int factorielle(int 2);
                                                                    □ dans lequel ces fonctions sont appelées dans le
                                                                                                                                le contenu d'un tableau de 5 entiers initialisé au
      \square n = factorielle(p, q);
                                                                                                                                préalable, on utilise plutôt :
     ☐ printf("%d", factorielle(n));
                                                                    □ alphabétique
                                                                                                                                  \square for(i=0;i<=5;i=i+1)
14. Soit la fonction g définie par :
                                                              17. Soit la fonction f définie par :
                                                                                                                                  \square for(i=0;i<5;i=i+1)
   int g(int a)
                                                                  int f(int a)
                                                                                                                                  \square for(i=1;i<5;i=i+1)
   {
                                                                  {
                                                                                                                                  \square for(i=1;i<=5;i=i+1)
      printf("a = \n", %d);
                                                                    printf("a = \n", %d);
                                                                                                                            20. Si x est une variable réelle (de type double) alors
      if (1 > 0)
                                                                    if (a > 0)
                                                                                                                                x = 3/2 lui affecte la valeur :
        return 5;
                                                                       return 3;
                                                                                                                                  \Box 0
                                                                                                                                  \square 0.5
      return 7;
                                                                    return 4;
                                                                                                                                   \Box 1
                                                                                                                                   \square 1.5
   Alors l'expression g(0) prendra la valeur :
                                                                  Alors l'expression f(0) prendra la valeur :
```

□ A

☐ ABCDEF

# Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Sous unix (ou linux), la commande cd permet de : □ changer de répertoire courant □ détruire un fichier ☐ récupérer un programme arrêté avec la commande ab □ jouer de la musique □ ouvir un bureau partagé (common desktop) 2. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C? □ char 'c'; □ char "c": ☐ int char; ☐ char c: 3. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme?  $\square$  une fonction appelée avant sa déclaration □ une fonction déclarée mais non définie un désaccord entre la déclaration et la définition d'une fonction ☐ une directive préprocesseur #include manquante 4. Au début de la fonction main() on place le code : char i; for  $(i = 'A'; i \le 'F'; i = i + 1)$ printf("%c", i); printf("\n"); Alors l'affichage sera:  $\Box$  i 

```
5. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
       somme = somme + i:
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 6
    \square 15
    \Box 10
    \Box 0
6. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
    ☐ int factorielle();
    \Box int factorielle(int x);
    ☐ int factorielle(double n):
     □ struct int factorielle(int n);
7. Si a et b sont deux variables de type:
   struct toto_s
  {
     int n;
     double x;
   Alors pour tester l'égalité de a et de b on utilise la
  condition:
    \square (a.n == b.n) \&\& (a.x == b.x)
    □ a == b
    \square a\{n, x\} == b\{n, x\}
    \Box a = b
8. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
```

return 5:

```
}
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 5
      \Box 0
      \square 7
 9. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
      \square n = factorielle(p, q);
     ☐ int factorielle(int 2);
     ☐ printf("%d", factorielle(n));
      \square n = factorielle();
10. Le bus système sert à :
      ☐ Écrire des données sur le dique dur
      ☐ Transférer des données et intructions entre pro-
         cesseur et mémoire
      □ transporter les processus du tourniquet au pro-
         cesseur
      ☐ Arriver à l'heure en cours
11. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \Box 5
      \Box 0
      \Box 1
      \Box 4
```

```
12. Les lignes
   int i;
   int x=0;
   for(i=0,i<5,i=i+1)
      x=x+1;
      \square comportent une erreur qui ne sera pas détectée
      □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
     \Box ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
13. L'ordonnancement par tourniquet permet :
      □ de doubler la mémoire disponible
      ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
      \square de ne pas perdre de temps avec la commutation
        de contexte
      □ d'afficher des ronds colorés à l'écran
14. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
```

```
}
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box i = 0
     \Box j = %d
     □ j = 5
     \Box j = 4
15. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
16. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     ☐ int saisie_utilisateur();
     □ void saisie_utilisateur(char c);
     ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
17. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %d\n", n);
     □ printf("Valeur de n ? %g\n", n);
     \square un débogueur
     □ scanf("%d", &n);
```

```
18. Après exécution jusqu'à la ligne 14 du programme C:
 10
       int main() {
 11
            int x = 5;
 12
 13
            printf(" x = %d\n", 2);
 14
 15
 16
      }
     \square le terminal affiche x = 5
      □ le terminal affiche "Faux"
     \square le terminal affiche x = 2
     \square le terminal affiche 5
19. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     \square une accolade manquante
     ☐ un point-virgule manquant
     □ un point-virgule en trop
      \square une accolade en trop
20. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     ☐ for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
```

 $\square$  for(i=0;i<=5;i=i+1)

 $\square$  for(i=1;i<=5;i=i+1)

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit le programme principal suivant :
  int main()
  {
    int a = 3;
    int b = 5:
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  }
  L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
2. Un registre du processeur est :
     \Box une unité de calcul spécialisée de l'ordinateur
     □ une case mémoire interne au processeur qui sera
       manipulée directement lors des calculs
     \Box une gamme de fréquence de fonctionnement du
       processeur
     \square un composant qui contient la liste des fichiers du
       système
3. Si cette erreur apparaît à la compilation :
  error: expected ';' before '}' token que doit-
  on chercher dans le programme?
     \square une accolade manquante
     \square un point-virgule en trop
     □ un point-virgule manquant
     \square une accolade en trop
```

```
4. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \Box 7
     \square 8
     \square 111
5. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     \square il ne compile pas
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il comporte une boucle infinie
     \square il n'affiche rien
6. Le code suivant:
    int i;
    for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 0 1 2 3
     \Box 43210
     \Box 01234
     \square 4 3 2 1
7. Le code suivant :
    int i;
    for (i = 4; i \ge 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 4 3 2 1
     \Box 01234
     \square 1 2 3 4
     \Box \ 4\ 3\ 2\ 1\ 0
```

```
8. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square exposant(double x, int n, int r);
     \square int exposant(double n, int x);
     \square double exposant(double x, int n);
     \square void exposant(double x^n);
 9. Le code suivant :
     int age = 15;
     if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
   affichera:
     □ Majeur
     ☐ Mineur
        Majeur
     □ Mineur
     □ rien
10. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ int char;
     ☐ char "c";
     □ char c;
     □ char 'c';
11. Le langage C est un langage
     □ composé
     □ compilé
     □ interprété
     □ lu, écrit, parlé
```

erreur: conflicting types for 'max', que doit- on chercher dans le programme?	□ 6 □ 16	$\Box$ voir des clips musicaux
☐ une directive préprocesseur <b>#include</b> manquante	$\Box$ 16	
		$\square$ afficher le contenu d'un fichier texte
	$\square$ 20	□ compiler un programme
☐ un désaccord entre la déclaration et la définition d'une fonction	$\square$ 3	
	15. Le code suivant :	□ afficher la liste de fichiers contenus dans un répertoire
□ une fonction déclarée mais non définie	int age = 18;	1
☐ une fonction appelée avant sa déclaration	if (age < 18)	18. Une variable booléenne est un variable :
13. Soit la fonction g définie par :	<pre>{     printf("Mineur\n");</pre>	☐ réelle positive
int g(int a)	princi("Mineur\n");	$\Box$ qui est vraie ou fausse
<pre>printf("a = \n", %d);</pre>	else	□ NaN (not a number, qui n'est pas un nombre)
if (1 > 0)	{	□ à laquelle une valeur vient d'être affectée
{	<pre>printf("Majeur\n");</pre>	□ jamais nulle
return 5;	}	v
}	affichera:	19. Pour compiler un programme prog.c, on utilise la ligne de commande :
return 7;	amenera.  □ Mineur	
Alors l'expression g(0) prendra la valeur :	Majeur	☐ gcc prog.c -o -Wall prog.exe
□ 5	☐ Mineur	$\square$ gcc -Wall prog.c -o prog.exe
$\square$ 7	□ rien	$\square$ gcc -Wall prog.exe -o prog.c
	□ Majeur	$\square$ gcc prog.exe -Wall -o prog.c
14. Pour l'extrait de programme suivant :	16. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est	20. Avant de faire appel à une fonction il est nécessaire de :
<pre>int somme = 0; int serie[4] = {2, 4, 10, 4}; for (i = 0; i &lt; 4; i = i + 1)</pre>	une variable réelle définie et initialisée, il est incorrect d'écrire :	□ avoir défini une constante symbolique de la taille de cette fonction
{		☐ l'avoir définie
<pre>somme = somme + serie[i];</pre>	$\square x = racine(x * x) - racine(x);$	☐ l'avoir déclarée et définie
}	$\square x - 1 = racine(x);$	
<pre>printf("somme = %d",somme);</pre>	$\Box$ x = racine(racine(x)*racine(x));	☐ l'avoir déclarée

acence	

#### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

```
□ char c;
□ int char;
□ char "c";
```

□ char 'c';

2. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
□ x - 1 = racine(x);
□ x = racine(2/3);
□ x = racine(x * x) - racine(x);
□ x = racine(racine(x)*racine(x));
3. Le code suivant:
   int i;
   for (i = 4: i >= 0: i = i - 1)
```

4. Après exécution jusqu'à la ligne 15 du programme C :

```
10 int main() {
11 int x = 5;
12 int y;
13
14 y = x;
15
16 ...
17 }
```

```
\square la variable x vaut 0
     □ le programme affiche "Faux"
    \square la variable x vaut 5 et la variable y vaut 0
5. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
   qu'est ce qui sera affiché?
     \Box 0 1 2 3 0 1 2
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 0 1 2
     \Box 0 1 2 0 1 2 3
6. Une de ces manière de composer les blocs de pro-
   grammes ne fait pas partie des opérations de la pro-
   grammation structurée :
     □ répéter un bloc tant qu'une condition est vérifée
    \square mettre les blocs en séquence les uns à la suite des
        autres
     □ retourner un bloc
     □ sélectionner entre deux blocs à l'aide d'une condi-
        tion
7. Après exécution du programme :
      lecture 8 r0
      valeur 3 r1
      mult r1 r0
      valeur 1 r2
      add r2 r0
      ecriture r0 8
      stop
      5
     \square le bus explose
```

□ le terminal affiche 8

□ la case mémoire 8 contiendra 16

□ la case mémoire 8 contiendra 0

```
8. Vous utilisez une boucle while quand:
      □ l'incrément de la variable de boucle n'est pas 1
      \Box vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      □ vous n'avez pas déclaré de fonction
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
 9. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse syntaxique
      \square analyse lexicale
      \square analyse harmonique
      □ analyse sémantique
10. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square n = mccarthy();
      ☐ int mccarthy(int 2);
      \square x = mccarthy(n);
      \square n = mccarthy(p, q);
11. Quel est l'opérateur de différence en C :
      □!=
      \Box !
      \square \neq
      \sqcap \iff
12. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ l'avoir définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      ☐ l'avoir déclarée
```

13. Après exécution jusqu'à la ligne 15 du programme $\mathcal{C}$ :	$\Box$ 4	□ (!A    B)
10 int main() {	$\square$ 3	☐ !(!A    B) == (A && !B)
11 int x = 5;	15. Quel est le problème d'un programme comportant les	□ A && B
12 int y = 3; 13 14 x = y;	lignes suivantes? while (1)	18. Si $n$ est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :
15	{	□ scanf("%d", &n);
16 17 }	<pre>printf("coucou\n"); }</pre>	$\Box$ un débogueur
☐ la variable y vaut 5 ☐ la variable x vaut 3	☐ il comporte une boucle infinie☐ il risque d'afficher bonjour à la place de coucou	<pre>□ printf("Valeur de n ? %g\n", n);</pre> <pre>□ printf("Valeur de n ? %d\n", n);</pre>
☐ le programme affiche "Faux"	☐ il n'affiche rien	19. Un bit est:
☐ la variable x vaut 5 et la variable y vaut 3	☐ il ne compile pas	$\Box$ un battement d'horloge processeur
<pre>14. Soit la fonction f définie par :    int f(int a)    {       printf("a = \n", %d);       if (a &gt; 0)</pre>	16. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :  □ n = pgcd(n, 3); □ int pgcd(2); □ n = pgcd(int p, int q);	☐ l'instruction qui met fin à un programme ☐ la longueur d'un mot mémoire ☐ un chiffre binaire (0 ou 1)  20. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit-
return 3;	☐ int n = pgcd();	on chercher dans le programme?
<pre>} return 4; }</pre>	17. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur	☐ une directive préprocesseur #include manquante ☐ une variable non déclarée
Alors l'expression f(0) prendra la valeur :	TRUE?	une faute de frappe dans un appel de fonction
$\square$ 0	☐ (A == TRUE) && (B == TRUE)	$\Box$ un caractère interdit en C

## Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
    }
    else
        printf("Majeur\n");
   }
  affichera:
     ☐ Mineur
       Majeur
    ☐ Mineur
    \square rien
    □ Majeur
2. Si cet avertissement apparaît à la compilation :
  warning: implicit declaration of function 'max'
  , que doit-on chercher dans le programme?
     □ une fonction déclarée mais non définie
    □ un désaccord entre la déclaration et la définition
       d'une fonction
    ☐ une fonction appelée avant sa déclaration
    ☐ une directive préprocesseur #include manquante
3. Au début de la fonction main() on place le code :
    char b = 'A';
    b = b + 2;
    printf("%c\n", b);
  Alors l'affichage sera:
    \Box C
    □В
    □ b
    \square A
```

```
4. Soit un programme contenant les lignes suivantes :
   int i = 0;
   int j = 0;
   for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
        }
   }
  qu'est ce qui sera affiché?
    \Box 0 0 0 1 1 1
    \Box 0 1 0 1 0 1 0 1
    □ 1 2 1 2 3
    \Box 0 1 2 0 1 2
5. Pour déclarer un tableau d'entiers de taille 5, on peut
  utiliser l'instruction
    \Box int tab[] = 5;
    \square int toto[5];
    \square int[] new tableau(5);
    □ char tableau[5];
    ☐ int toto[taille=5];
6. Pour déclarer une fonction pgcd qui calcule et renvoie
  le plus grand diviseur commun de deux entiers positifs
  passés en arguments on écrit :
    \square int pgcd(int x, int x);
    \square int pgcd(int x, y);
    \square void pgcd(int x, int y);
    \square int pgcd(int y, int x);
7. Pour compiler un programme prog.c, on utilise la
  ligne de commande :
    ☐ gcc -Wall prog.c -o prog.exe
    ☐ gcc prog.exe -Wall -o prog.c
```

☐ gcc prog.c -o -Wall prog.exe

☐ gcc -Wall prog.exe -o prog.c

```
8. Au début de la fonction main() on place le code :
     char i:
     for (i = 'A'; i \le 'F'; i = i + 1)
       printf("%c", i);
     printf("\n");
    Alors l'affichage sera:
     \Box ccccc
     □ A
      Πi
      ☐ ABCDEF
 9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square void exposant(double x^n);
     \square int exposant(double n, int x);
      \square exposant(double x, int n, int r);
      \square double exposant(double x, int n);
10. Après exécution jusqu'à la ligne 14 du programme C:
       int main() {
 10
 11
            int x = 5;
 12
 13
            printf(" x = %d\n", 2);
 14
 15
 16
      □ le terminal affiche 5
      □ le terminal affiche "Faux"
      \square le terminal affiche x = 5
      \square le terminal affiche x = 2
11. Quel est le problème d'un programme comportant les
    lignes suivantes?
   while (1)
      printf("coucou\n");
```

$\square$ il risque d'afficher bonjour à la place de coucou $\square$ il comporte une boucle infinie	<ul><li>14. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :</li><li>□ int afficher_menu(int char);</li></ul>	18. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
$\square$ il ne compile pas	☐ int afficher_menu();	☐ #include <studlib.h></studlib.h>
□ il n'affiche rien  12. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect	□ void afficher_menu(); □ double afficher_menu(); □ char afficher_menu(printf("menu"));  15. Un enregistrement permet de grouper plusieurs valeurs dans:	<ul> <li>☐ #include <stdio.h></stdio.h></li> <li>☐ #appart <stdlib.h></stdlib.h></li> <li>19. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail</li> </ul>
d'écrire: $\square x - 1 = racine(x);$	□ ses blocs □ ses chants	sur le disque dur. Mais on perd :  ☐ en temps d'accès
$\Box$ x = racine(x * x) - racine(x);	□ ses cases	□ certaines données de la mémoire de travail
$\Box$ x = racine(2/3);	□ ses champs	$\square$ des processus
<pre>     x = racine(racine(x)*racine(x)); </pre>	16. Dans la commande gcc, l'option -Wall signifie :  ☐ qu'il faut indenter le fichier source	$\Box$ les fichiers du disque
13. Le bus système sert à :	$\Box$ qu'on veut changer alétoirement de fond d'écran	20. Si cette erreur apparaît à la compilation :
$\Box$ Écrire des données sur le dique dur	☐ qu'il faut lancer un déboggueur ☐ que l'on veut voir tous les avertissements	erreur: conflicting types for 'max', que doit- on chercher dans le programme?
☐ transporter les processus du tourniquet au processeur	17. Le langage C est un langage	une directive préprocesseur <b>#include</b> manquante
☐ Transférer des données et intructions entre pro-	□ composé □ interprété	☐ un désaccord entre la déclaration et la définition d'une fonction
cesseur et mémoire	□ compilé	$\square$ une fonction appelée avant sa déclaration
☐ Arriver à l'heure en cours	$\square$ lu, écrit, parlé	$\Box$ une fonction déclarée mais non définie

т		-1
	100000	- 1
	acence	

#### Éléments d'informatique – contrôle continue

 $\square$  5

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

□ Mineur

□ Majeur

 $\square$  rien

2. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
    scanf("%d", &n);
}

Quelle est la condition cond:
    a <= n <= b
    (n <= a) && (n <= b)
    (a < n) || (n > b)
    (a <= n) && (n <= b)</pre>
```

3. Sur un ordinateur avec un seul processeur, habituelle-

ment les processus sont exécutés :

 $\square$  tous ensemble

```
□ tour à tour, un petit peu à chaque fois
     □ chacun son tour, après que le processus précédent
        a terminé
     □ en parallèle, chacun dans un registre
4. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x - 1 = racine(x):
    \square x = racine(x * x) - racine(x);
    \square x = racine(2/3);
     \square x = racine(racine(x)*racine(x)):
5. Le langage C est un langage
     □ composé
     □ compilé
     □ lu, écrit, parlé
     □ interprété
6. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
   , que doit-on chercher dans le programme?
     □ une directive préprocesseur #include manquante
     □ une fonction déclarée mais non définie
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     \square une fonction appelée avant sa déclaration
7. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \square 7
     \Box 0
```

```
8. Vous utilisez une boucle while quand:
      □ vous n'avez pas déclaré de fonction
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
      \square l'incrément de la variable de boucle n'est pas 1
      □ vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
 9. Un fichier source est:
      □ un fichier texte qui sera traduit en instructions
         processeur
      \square un document de référence du système
      □ un document qui doit être protégé
      \square un document illisible pour les humains
      \Box un fichier que l'ont doit citer dans les documents
         produits sur l'ordinateur
10. Le bus système sert à :
      ☐ Transférer des données et intructions entre pro-
         cesseur et mémoire
      □ transporter les processus du tourniquet au pro-
      ☐ Écrire des données sur le dique dur
      ☐ Arriver à l'heure en cours
11. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
         return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
     \Box 4
      \square 3
      \square 0
```

12. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :	15. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :	
<pre>□ n = factorielle(p, q);</pre> □ n = factorielle();	☐ #include <studio.h> ☐ #include <stdio.h></stdio.h></studio.h>	<ul><li>□ 5</li><li>18. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction</li></ul>
<ul> <li>□ int factorielle(int 2);</li> <li>□ printf("%d", factorielle(n));</li> <li>13. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?</li> <li>□ char c;</li> <li>□ char "c";</li> <li>□ char 'c';</li> <li>□ char 'c';</li> </ul>	<pre></pre>	<ul> <li>□ char tableau[5];</li> <li>□ int toto[taille=5];</li> <li>□ int tab[] = 5;</li> <li>□ int toto[5];</li> <li>□ int[] new tableau(5);</li> <li>19. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et</li> </ul>
☐ int char;  14. Le code suivant :     int i;     for (i = 8; i > 0; i = i - 2)     {         printf("%d ", i);     }     printf("\n");     affichera :     □ 8 2     □ 8 6 4 2 0     □ 0 2 4 6 8     □ 8 6 4 2	<pre>□ n = mccarthy();  17. Soit la fonction f définie par :     int f(int a)     {         printf("a = \n", %d);         if (a &gt; 0)         {             return f(a - 1) + 1;         }         return 4;     }  Alors l'expression f(1) prendra la valeur :     □ 4     □ 1</pre>	renvoie cet entier on écrit :  int saisie_utilisateur();  void saisie_utilisateur(char c);  saisie_utilisateur(scanf(%d));  void saisie_utilisateur(int n);  20. L'écriture 101 en binaire correspond au nombre naturel:  3  5  4

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\square$  3  $\Box$  4  $\Box$  0 2. Soit le programme principal suivant : int main() int a = 3; int b = 5; printf("f(a,b)=%d, a=%d,  $b=%d\n$ ", f(a,b), a,b); return EXIT\_SUCCESS; appelant la fonction f ainsi définie : int f(int a, int b) { a = a + b; return a; L'affichage dans le main est le suivant :  $\Box$  f(3,5)=8, a=3, b=5  $\Box$  f(a,b)=13, a=8, b=5  $\Box$  f(a,b)=8, a=8, b=5  $\Box$  f(a,b)=8, a=3, b=5 3. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt : □ scanf("%d", &n); ☐ printf("Valeur de n ? %d\n", n);  $\square$  un débogueur

□ printf("Valeur de n ? %g\n", n);

```
4. Sous unix (ou linux), la commande 1s permet de :
     \square afficher la liste de fichiers contenus dans un
        répertoire
     □ voir des clips musicaux
     □ compiler un programme
     \square afficher le contenu d'un fichier texte
5. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 16
     \Box 4
     \square 8
     \square 0
6. Vous utilisez une boucle while quand :
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     ☐ l'incrément de la variable de boucle n'est pas 1
7. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
    \square n = carre(n);
    \square int carre(2):
    \square int n = carre();
     \square n = carre(int n):
```

```
8. Soit un programme contenant les lignes suivantes :
    int i = 0:
    int j = 0;
    for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box j = 0
     \Box j = 4
     \Box j = 5
     \Box j = %d
9. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     □ void afficher_date(struct date_s d);
     ☐ int afficher date(date s d):
     □ void afficher_date(date_s d);
     □ struct date_s afficher_date(struct date_s d);
10. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
     \square #define taille = 3
     \square #define N = 3
     □ #define N 3
     ☐ #define taille = N
11. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
```

```
{
             printf("%d ", j);
         }
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2 3
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 3 0 1 2
     \square 0 1 2 0 1 2
12. Le code suivant :
     int age = 18;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     \square rien
     \square Mineur
        Majeur
     \square Majeur
     \square Mineur
```

13. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :  □ sélectionner entre deux blocs à l'aide d'une condition □ répéter un bloc tant qu'une condition est vérifée □ mettre les blocs en séquence les uns à la suite des autres □ retourner un bloc	<ul> <li>17. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme?  ☐ une directive préprocesseur #include manquante ☐ une fonction appelée avant sa déclaration ☐ un désaccord entre la déclaration et la définition d'une fonction ☐ une fonction déclarée mais non définie</li> </ul>
<ul><li>14. Un enregistrement permet de grouper plusieurs valeurs dans :</li><li>□ ses champs</li></ul>	18. Pour déclarer une fonction exposant qui prend en argument un réel $x$ et un entier positif $n$ et renvoie la valeur de $x^n$ on écrit :
□ ses cases	$\square$ exposant(double x, int n, int r);
$\square$ ses blocs	$\square$ void exposant(double x^n);
$\square$ ses chants	$\square$ double exposant(double x, int n);
<ul> <li>15. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :  □ n = pgcd(n, 3); □ n = pgcd(int p, int q); □ int pgcd(2); □ int n = pgcd();</li> <li>16. Un fichier source est : □ un document illisible pour les humains □ un document de référence du système</li> <li>□ un document qui doit être protégé</li> </ul>	☐ int exposant(double n, int x);  19. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?  ☐ int char; ☐ char 'c'; ☐ char c; ☐ char "c";  20. Le langage C est un langage
<ul> <li>□ un document qui doit être protégé</li> <li>□ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur</li> <li>□ un fichier texte qui sera traduit en instructions processeur</li> </ul>	□ interprété □ lu, écrit, parlé □ composé □ compilé

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Vous utilisez une boucle while quand: ☐ l'incrément de la variable de boucle n'est pas 1 □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance □ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction 2. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : ☐ int factorielle(); □ struct int factorielle(int n); ☐ int factorielle(double n):  $\square$  int factorielle(int x); 3. Pour compiler un programme prog.c, on utilise la ligne de commande : ☐ gcc prog.c -o -Wall prog.exe ☐ gcc prog.exe -Wall -o prog.c ☐ gcc -Wall prog.exe -o prog.c ☐ gcc -Wall prog.c -o prog.exe 4. Sur unix (ou linux), la commande mkdir permet de : □ créer un fichier texte □ créer un répertoire  $\square$  ouvrir un fichier texte  $\Box\,$  changer de répertoire courant 5. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1)for (j = 0; j < 5; j = j + 1)}  $printf("j = %d\n", j);$ 

```
qu'est ce qui sera affiché par ce printf?
    \Box j = 5
     \Box j = 0
     \Box j = %d
     \Box j = 4
6. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
     {
       produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 4
     \square 8
     \Box 0
     \square 16
7. Après exécution jusqu'à la ligne 15 du programme C:
      int main() {
 10
 11
           int x = 5:
 12
           int y;
 13
 14
           y = x;
 15
 16
 17
      }
     \square la variable x vaut 5 et la variable y vaut 0
     \square la variable x vaut 0
     \square la variable v vaut 5
     ☐ le programme affiche "Faux"
8. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max
   , que doit-on chercher dans le programme?
     \square une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
     ☐ une directive préprocesseur #include manquante
     \square un désaccord entre la déclaration et la définition
        d'une fonction
```

	9.	Un enregistrement permet de grouper plusieurs valeurs dans :
		□ ses champs
		ses cases
		ses chants
		ses blocs
	10	
	10.	L'écriture <u>111</u> en binaire correspond au nombre naturel :
		□ 8
		□ 111
		$\square$ 3
		□ 7
	11.	Avant de faire appel à une fonction il est nécessaire
		de:
		$\hfill \square$ avoir défini une constante symbolique de la taille de cette fonction
		□ l'avoir déclarée
		$\Box$ l'avoir déclarée et définie
		□ l'avoir définie
	12.	Si $\operatorname{pgcd}$ est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
		☐ int pgcd(2);
		$\square$ n = pgcd(n, 3);
		$\square$ int n = pgcd();
		$\square$ n = pgcd(int p, int q);
	13.	Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C ?
		☐ int char;
		□ char "c";
		□ char 'c';
		□ char c;
,	14.	Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
		☐ for(i=0;i<5;i=i+1)
		☐ for(i=0;i<=5;i=i+1)
		☐ for(i=1;i<=5;i=i+1)
		☐ for(i=1:i<5:i=i+1)

```
15. Si x est une variable réelle (de type double) alors
                                                                    printf("a = \n", %d);
                                                                                                                                }
    x = 3/2 lui affecte la valeur :
                                                                    if (a > 0)
                                                                                                                                else
                                                                    {
                                                                                                                                {
      \square 1.5
                                                                      return f(a - 1) + 1;
                                                                                                                                     printf("Majeur\n");
      \square 0.5
      \Box 0
                                                                    return 4;
      \Box 1
                                                                                                                               affichera:
                                                                 Alors l'expression f(1) prendra la valeur :
16. Le code suivant :
                                                                                                                                 □ Majeur
                                                                   \Box 1
     int age = 20;
                                                                                                                                 ☐ Mineur
                                                                   \Box 4
     if (age < 18)
                                                                                                                                    Majeur
     {
                                                                   \Box 0
                                                                                                                                 ☐ Mineur
         printf("Mineur\n");
                                                                   \Box 5
                                                                                                                                 \square rien
                                                             18. L'écriture 101 en binaire correspond au nombre natu-
                                                                                                                           20. Quel est le problème d'un programme comportant les
     printf("Majeur\n");
                                                                 rel:
                                                                                                                               lignes suivantes?
    affichera:
                                                                   \Box 5
                                                                                                                               while (1)
      ☐ Mineur
                                                                   \Box 101
                                                                                                                               {
      □ Majeur
                                                                                                                                 printf("coucou\n");
                                                                   \square 3
      ☐ Mineur
                                                                   \Box 4
         Majeur
                                                                                                                                 \square il n'affiche rien
                                                             19. Le code suivant :
      \square rien
                                                                                                                                 \square il comporte une boucle infinie
                                                                  int age = 15;
17. Soit la fonction f définie par :
                                                                  if (age < 18)
                                                                                                                                 \square il risque d'afficher bonjour à la place de coucou
    int f(int a)
                                                                  {
                                                                                                                                 \square il ne compile pas
                                                                       printf("Mineur\n");
    {
```

 $\square$  0.5

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le bus système sert à : ☐ Écrire des données sur le dique dur □ Transférer des données et intructions entre processeur et mémoire ☐ Arriver à l'heure en cours □ transporter les processus du tourniquet au processeur 2. Avant de faire appel à une fonction il est nécessaire de: □ l'avoir définie □ l'avoir déclarée  $\square$  avoir défini une constante symbolique de la taille de cette fonction □ l'avoir déclarée et définie 3. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <studio.h> ☐ #include <stdio.h> ☐ #include <studlib.h> ☐ #appart <stdlib.h> 4. Après la déclaration : int mccarthy(int n);, il est correct d'écrire:  $\square$  n = mccarthy(p, q);  $\square$  int mccarthy(int 2);  $\square$  x = mccarthy(n);  $\square$  n = mccarthy(); 5. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\square$  1.5  $\Box$  0  $\Box$  1

```
6. Un fichier source est:
    \square un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
    □ un document illisible pour les humains
    □ un document de référence du système
    □ un document qui doit être protégé
    □ un fichier texte qui sera traduit en instructions
        processeur
7. Le code suivant :
    int i:
    for (i = 4; i \ge 0; i = i - 1)
        printf("%d ", i);
   printf("\n");
   affichera:
    \Box 01234
    \Box \ 4\ 3\ 2\ 1\ 0
    \square 4 3 2 1
    \square 1 2 3 4
8. Pour afficher à l'aide de printf("%d\n",tab[i]);
  le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
    \square for(i=0;i<5;i=i+1)
    \square for(i=0;i<=5;i=i+1)
    \square for(i=1;i<=5;i=i+1)
    ☐ for(i=1;i<5;i=i+1)
9. Sur unix (ou linux), la commande mkdir permet de :
     □ ouvrir un fichier texte
    □ changer de répertoire courant
    □ créer un répertoire
     \Boxcréer un fichier texte
```

```
10. Pour l'extrait de programme suivant :
     int i = 0:
     int j = 0;
    for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
              printf("%d ", i);
         }
    }
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 0 1 1 2 2
     \Box 0 1 2 0 1 2
     \square 1 2 3 1 2
     \Box 0 1 0 1 0 1
11. Pour compiler un programme prog.c, on utilise la
    ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
12. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
      printf("coucou\n");
     \square il ne compile pas
     \square il comporte une boucle infinie
     □ il risque d'afficher bonjour à la place de coucou
     □ il n'affiche rien
13. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
    printf("\n");
```

affichera:	affichera:	19. Soit la fonction f définie par :
$\Box \ 0\ 1\ 2\ 3\ 4$	☐ Mineur	int f(int a)
$\Box \ 4\ 3\ 2\ 1\ 0$	☐ Majeur	<pre>int f(int a) f</pre>
$\square  4 3 2 1$	□ rien	printf("a = \n", %d);
$\Box$ 1 2 3 4	☐ Mineur	if (a > 0)
14. Si carre est une fonction prenant en entrée un en-	Majeur	{
tier et renvoyant le carré de cet entier, et que n est	17. Si a et b sont deux variables de type :	return f(a - 1) + 1;
une variable entière définie et initialisée, il est correct	struct toto_s	}
d'écrire :	{	return 4;
☐ int n = carre();	int n;	1
$\square$ n = carre(int n);	double x;	Alors l'expression f(1) prendra la valeur :
$\square$ n = carre(n);	};	$\Box$ 4
☐ int carre(2);	Alors pour tester l'égalité de a et de b on utilise la condition :	
15. Un enregistrement permet de grouper plusieurs valeurs	condition:  □ a == b	
dans:	□ a b □ a{n, x} == b{n, x}	$\square$ 0
□ ses cases	$\Box \text{ (a.n == b.n) \&\& (a.x == b.x)}$	$\square$ 5
□ ses blocs		
□ ses chants	□ a = b	20. Si cette erreur apparaît à la compilation :
□ ses champs	18. Vous avez déclaré préalablement un ensemble de fonc- tions utilisées par votre programme principal. L'ordre	erreur: conflicting types for 'max', que doit-
-	dans lequel vous devez maintenant définir ces fonctions	on chercher dans le programme?
16. Le code suivant :	est l'ordre :	$\Box$ un désaccord entre la déclaration et la définition
int age = 20;	□ un ordre quelconque	d'une fonction
if (age < 18)	☐ dans lequel vous avez déclaré ces fonction	$\hfill \square$ une fonction déclarée mais non définie
<pre>i printf("Mineur\n");</pre>	□ alphabétique	☐ une directive préprocesseur #include manquante
}	☐ dans lequel ces fonctions sont appelées dans le	• •
<pre>printf("Majeur\n");</pre>	main	$\square$ une fonction appelée avant sa déclaration

19.	Soit la fonction <b>f</b> définie par :
	<pre>int f(int a) {   printf("a = \n", %d);   if (a &gt; 0)   {     return f(a - 1) + 1;   }   return 4; }</pre>
	Alors l'expression f(1) prendra la valeur :  4  1  0  5
20.	Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit on chercher dans le programme?
	<ul> <li>□ un désaccord entre la déclaration et la définition d'une fonction</li> <li>□ une fonction déclarée mais non définie</li> </ul>

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Un fichier source est: □ un document qui doit être protégé □ un fichier texte qui sera traduit en instructions processeur  $\square$  un document illisible pour les humains □ un document de référence du système ☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur 2. Pour l'extrait de programme suivant : int somme = 0; for (i = 0; i < 5; i = i + 1)somme = somme + i;i = i + 1; /\* attention ! \*/ printf("somme = %d",somme); La valeur de somme affichée est :  $\Box$  6  $\Box$  0  $\square$  10  $\square$  15 3. Le langage C est un langage □ interprété □ composé □ lu, écrit, parlé □ compilé 4. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :  $\square$  x = racine(x \* x) - racine(x):  $\square$  x - 1 = racine(x):

 $\square$  x = racine(racine(x)\*racine(x)):

 $\square$  x = racine(2/3);

```
5. On considère deux variables booléennes A et B initia-
  lisées à TRUE et FALSE respectivement. Parmi les ex-
  pressions booléennes suivantes, laquelle a pour valeur
  TRUE?
    \square !(!A | | B) == (A && !B)
    \square (A == TRUE) && (B == TRUE)
    □ A && B
    \square (!A | | B)
6. Une variable booléenne est un variable :
    □ NaN (not a number, qui n'est pas un nombre)
    □ jamais nulle
    \square qui est vraie ou fausse
    ☐ réelle positive
    □ à laquelle une valeur vient d'être affectée
7. Pour déclarer une procédure afficher_date qui prend
  en argument un struct date_s et affiche le contenu
  du struct, on écrit:
    ☐ int afficher_date(date_s d);
    □ void afficher_date(struct date_s d);
    □ struct date_s afficher_date(struct date_s d);
    □ void afficher_date(date_s d);
8. Le type des réels en C est :
    □ real
    □ char
    □ int.
    ☐ double
9. Un registre du processeur est :
    □ une case mémoire interne au processeur qui sera
       manipulée directement lors des calculs
    □ une unité de calcul spécialisée de l'ordinateur
    \square un composant qui contient la liste des fichiers du
       système
    □ une gamme de fréquence de fonctionnement du
```

processeur

```
10. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      □ include
      \square init
      □ main
      □ begin
11. Sur unix (ou linux), la commande mkdir permet de :
      \(\sigma\) ouvrir un fichier texte
      □ créer un répertoire
      □ créer un fichier texte
      □ changer de répertoire courant
12. Si a et b sont deux variables de type :
    struct toto_s
    {
      int n:
      double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
      \Box a = b
      \square a{n, x} == b{n, x}
      □ a == b
      \square (a.n == b.n) && (a.x == b.x)
13. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
 11
            int x = 5;
 12
            int y = 3;
 13
 14
            x = y;
  15
  16
             . . .
  17
      \square la variable x vaut 5 et la variable y vaut 3
      \square la variable x vaut 3
      ☐ le programme affiche "Faux"
      \square la variable v vaut 5
```

```
14. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
   Alors l'affichage sera :
      \square A
     \Box C
     □ b
     □В
15. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %d\n", n);
     ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
     \square un débogueur
16. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
 11
           int x = 5;
 12
           int y;
 13
 14
           y = x;
 15
 16
            . . .
 17
      }
```

```
\square la variable y vaut 5
     \square la variable x vaut 5 et la variable y vaut 0
     \Box la variable x vaut 0
     □ le programme affiche "Faux"
17. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
      ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.c -o -Wall prog.exe
      ☐ gcc prog.exe -Wall -o prog.c
18. Le code suivant :
     int i;
     for (i = 1; i < 5; i = i + 1)
     {
         printf("%d ", i);
    printf("\n");
    affichera:
      \Box 4 3 2 1 0
     \square 1 2 3 4
     \Box 01234
      \square 4 3 2 1
```

```
19. Soit la fonction g définie par :
    int g(int a)
    {
        printf("a = \n", %d);
        if (1 > 0)
        {
            return 5;
        }
        return 7;
    }

Alors l'expression g(0) prendra la valeur :
        □ 7
        □ 5
        □ 0

20. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :
        □ int factorielle(int 2);
        □ n = factorielle(p, q);
```

☐ printf("%d", factorielle(n));

 $\square$  n = factorielle();

 $\Box$  0

## Éléments d'informatique – contrôle continue

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?  $\square$  une fonction déclarée mais non définie □ une directive préprocesseur #include manquante ☐ une fonction appelée avant sa déclaration  $\square$  un désaccord entre la déclaration et la définition d'une fonction 2. Soit la fonction g définie par : int g(int a)  $printf("a = \n", %d);$ if (1 > 0)return 5: } return 7; Alors l'expression g(0) prendra la valeur :  $\Box$  7  $\Box$  0  $\square$  5 3. Pour l'extrait de programme suivant : int produit = 1; int  $serie[4] = \{2, 2, 2, 2\};$ for (i = 0; i < 4; i = i + 1)produit = produit \* serie[i]; printf("produit = %d", produit); La valeur affichée est :  $\square$  8  $\Box$  16  $\Box$  4

```
4. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
     \square (!A | | B)
     □ A && B
     \square (A == TRUE) && (B == TRUE)
     \square !(!A | | B) == (A && !B)
5. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \Box 1
     \square 0.5
     \square 1.5
     \Box 0
6. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
     □ en parallèle, chacun dans un registre
     □ chacun son tour, après que le processus précédent
        a terminé
     □ tour à tour, un petit peu à chaque fois
     \square tous ensemble
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     \square ses blocs
     □ ses chants
     \square ses champs
8. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int i = 0;
    for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
    printf("j = %d\n", j);
```

```
qu'est ce qui sera affiché par ce printf?
     \Box j = %d
     \Box i = 4
     \Box i = 5
     \Box i = 0
 9. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
10. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     □ void afficher_menu();
     ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
     ☐ char afficher_menu(printf("menu"));
     □ double afficher_menu();
11. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
     □ printf("%d", factorielle(n));
     \square n = factorielle();
     \square n = factorielle(p, q);
     ☐ int factorielle(int 2);
12. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3;
      return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
     \Box 0
```

13. Si cet avertissement apparaît à la compilation :	16. Sur unix (ou linux), la commande mkdir permet de :	<pre>printf("Mineur\n");</pre>	
warning: implicit declaration of function 'max	' □ créer un répertoire	}_	
, que doit-on chercher dans le programme?	$\square$ changer de répertoire courant	else	
une directive préprocesseur #include manquante	$\square$ créer un fichier texte	printf("Majeur\n");	
<ul> <li>□ une fonction déclarée mais non définie</li> <li>□ un désaccord entre la déclaration et la définition</li> </ul>	$\square$ ouvrir un fichier texte	}	
d'une fonction	17. Un bit est:		
$\hfill\Box$ une fonction appelée avant sa déclaration	$\square$ la longueur d'un mot mémoire	affichera:	
14. Pour déclarer une variable qui sera utilisée comme va-	$\square$ un chiffre binaire (0 ou 1)	□ Mineur	
riable de boucle on peut utiliser l'instruction	$\Box$ l'instruction qui met fin à un programme	Majeur	
□ loop i;	$\Box$ un battement d'horloge processeur	□ Majeur	
☐ int loop n; ☐ int k; ☐ int %;	18. Le langage C est un langage	□ Mineur	
	□ interprété	□ rien	
☐ int %d; 15. Vous utilisez une boucle while quand:	□ lu, écrit, parlé	20. Après la déclaration : int mccarthy(int n);, il est	
vous avez déjà fait un for dans le même pro-	$\square$ composé	correct d'écrire :	
gramme principal	□ compilé	$\square$ n = mccarthy();	
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	19. Le code suivant :	$\square x = mccarthy(n);$	
☐ l'incrément de la variable de boucle n'est pas 1	int age = 20;	☐ int mccarthy(int 2);	
□ vous n'avez pas déclaré de fonction	if (age < 18) {	$\square$ n = mccarthy(p, q);	
		•	

## Éléments d'informatique - contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise:
     ☐ int afficher menu():
     ☐ int afficher_menu(int char);
     □ void afficher_menu();
    ☐ double afficher_menu();
     ☐ char afficher_menu(printf("menu"));
2. Soit la fonction f définie par :
  int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     }
     return 4;
  Alors l'expression f(1) prendra la valeur :
     \Box 5
    \Box 0
     \Box 4
    \Box 1
3. Un fichier source est:
     \square un document de référence du système
    ☐ un fichier que l'ont doit citer dans les documents
       produits sur l'ordinateur
     □ un fichier texte qui sera traduit en instructions
       processeur
     □ un document qui doit être protégé
     □ un document illisible pour les humains
4. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
```

for (j = 0; j < 5; j = j + 1)

```
{
         }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box i = %d
     \Box j = 0
     \Box j = 5
     \Box i = 4
5. Le code suivant :
    for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 02468
     \square 8 2
     \Box 8 6 4 2
     \square 8 6 4 2 0
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     □ ses chants
     □ ses cases
     \square ses blocs
7. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     \square il comporte une boucle infinie
     \square il ne compile pas
     ☐ il risque d'afficher bonjour à la place de coucou
     □ il n'affiche rien
```

```
8. Pour déclarer une variable qui sera utilisée comme va-
    riable de boucle on peut utiliser l'instruction
      \square int k;
      \square loop i;
     \square int %d;
      \square int loop n;
 9. Si le code :
    struct toto s
      int n;
      double x;
   précède la fonction main(), alors on peut écrire en
   début de main() :
     \square int toto.n = 3;
     ☐ struct toto_s toto;
     \Box int struct toto_s = {3, -1e10};
      \square toto_s n, x;
      \square toto_s struct z = {3, 0.5};
10. Quelle étape de la compilation vient d'échouer lors-
    qu'on a un message comme celui-ci :
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
      □ l'analyse sémantique
      ☐ l'analyse harmonique
      □ l'analyse des entrées clavier
      □ l'édition de liens
11. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int carre(2);
     \square int n = carre();
      \square n = carre(int n):
```

 $\square$  n = carre(n):

12.	Pour compiler un programme prog.c, on utilise la ligne de commande :
	☐ gcc prog.c -o -Wall prog.exe
	☐ gcc -Wall prog.c -o prog.exe
	☐ gcc -Wall prog.exe -o prog.c
	☐ gcc prog.exe -Wall -o prog.c
13.	Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
	☐ saisie_utilisateur(scanf(%d));
	☐ void saisie_utilisateur(int n);
	☐ void saisie_utilisateur(char c);
	☐ int saisie_utilisateur();
14.	Un bit est:
	$\Box$ la longueur d'un mot mémoire
	$\hfill\Box$ l'instruction qui met fin à un programme
	$\Box$ un battement d'horloge processeur
	$\Box$ un chiffre binaire (0 ou 1)
15.	Soit la fonction f définie par :
	<pre>int f(int a) {</pre>
	<pre>printf("a = \n", %d); if (a &gt; 0) {</pre>
	return 3;
	} return 4;
	}
	Alors l'expression f(0) prendra la valeur :
	The state of the s

```
\square 3
      \Box 4
      \Box 0
16. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
     \Box int pgcd(int y, int x);
     \Box int pgcd(int x, y);
     \Box int pgcd(int x, int x);
     \square void pgcd(int x, int y);
17. Après exécution jusqu'à la ligne 14 du programme C :
       int main() {
  10
            int x = 5;
  11
  12
  13
            printf(" x = %d\n", 2);
  14
  15
  16
       }
      □ le terminal affiche 5
      \square le terminal affiche x = 5
      □ le terminal affiche "Faux"
      \square le terminal affiche x = 2
18. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      □ une directive préprocesseur #include manquante
     □ une fonction déclarée mais non définie
     \square un désaccord entre la déclaration et la définition
         d'une fonction
     ☐ une fonction appelée avant sa déclaration
```

```
19. Soit le programme principal suivant :
   int main()
     int a = 3;
     int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=13, a=8, b=5
20. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
     ☐ for(i=1;i<=5;i=i+1)
```

☐ for(i=0;i<=5;i=i+1)

## Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \square 0
     \square 3
     \Box 4
2. Après exécution jusqu'à la ligne 14 du programme C:
      int main() {
           int x = 5;
 11
 12
 13
           printf(" x = %d\n", 2);
 14
 15
 16
     \square le terminal affiche x = 5
     \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
     \square le terminal affiche 5
3. Si le code :
   struct toto s
     int n:
     double x;
   };
   précède la fonction main(), alors on peut écrire en
   début de main() :
     \square int struct toto_s = {3, -1e10};
     □ struct toto s toto:
```

```
\square int toto.n = 3:
    \Box toto_s struct z = {3, 0.5};
    \square toto_s n, x;
4. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
    □ struct int factorielle(int n);
    ☐ int factorielle(double n);
    \Box int factorielle(int x);
    ☐ int factorielle();
5. Pour l'extrait de programme suivant :
     int somme = 0;
     int serie[4] = \{2, 4, 10, 4\};
     for (i = 0; i < 4; i = i + 1)
       somme = somme + serie[i];
     printf("somme = %d",somme);
  La valeur de somme affichée est :
    \square 16
    \square 3
    \Box 6
    \square 20
6. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
   }
  qu'est ce qui sera affiché?
    \Box 0 1 2 3 0 1 2
    \Box 0 1 2 0 1 2
    \Box 0 0 1 1 2 2 3
     \Box 0 1 2 0 1 2 3
```

```
7. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise:
    ☐ int afficher_menu(int char);
    ☐ int afficher_menu();
    ☐ double afficher_menu();
    □ void afficher_menu();
    ☐ char afficher_menu(printf("menu"));
8. Soit la fonction g définie par :
   int g(int a)
  {
    printf("a = \n", %d);
    if (1 > 0)
       return 5;
    return 7;
   Alors l'expression g(0) prendra la valeur :
    \Box 0
    \Box 5
    \Box 7
9. Soit un programme contenant les lignes suivantes :
   int i = 0;
   int j = 0;
   for (i = 0; i < 0; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
        }
   }
   printf("j = %d\n", j);
  qu'est ce qui sera affiché?
    \Box i = 4
    \Box j = %d
    \Box j = 5
    \Box j = 0
```

10. Pour déclarer un tableau d'entiers de taille 5, on peut	14. Un fichier source est:	18. Vous utilisez une boucle while quand :
utiliser l'instruction	$\square$ un document illisible pour les humains	□ vous ne connaissez pas le nombre d'itérations de
☐ char tableau[5];	□ un document de référence du système	la boucle à l'avance
☐ int toto[taille=5];	un document qui doit être protégé	$\Box$ vous avez déjà fait un for dans le même pro-
☐ int toto[5];	☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur	gramme principal
☐ int[] new tableau(5);	un fichier texte qui sera traduit en instructions	$\Box$ vous n'avez pas déclaré de fonction
☐ int tab[] = 5;	processeur  15. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre	$\hfill \square$ l'incrément de la variable de boucle n'est pas $1$
11. Un bit est:		tions utilisées par votre programme principal. L'ordre
☐ la longueur d'un mot mémoire	dans lequel vous devez maintenant définir ces fonctions est l'ordre :	□ ses champs
$\Box$ l'instruction qui met fin à un programme	□ dans lequel ces fonctions sont appelées dans le	ses chants
$\square$ un chiffre binaire (0 ou 1)	main	
$\Box$ un battement d'horloge processeur	$\Box$ dans lequel vous avez déclaré ces fonction	☐ ses cases
12. Si factorielle est une fonction prenant en entrée un	<ul> <li>□ un ordre quelconque</li> <li>□ alphabétique</li> <li>16. Dans la commande gcc, l'option -Wall signifie :</li> <li>□ qu'on veut changer alétoirement de fond d'écran</li> </ul>	□ ses blocs
entier et renvoyant un entier, il est correct d'écrire :		20. Au début de la fonction main() on place le code :
☐ n = factorielle();		char i;
☐ int factorielle(int 2);	□ qu'il faut indenter le fichier source	for (i = 'A'; i <= 'F'; i = i + 1)
☐ printf("%d", factorielle(n));	□ qu'il faut lancer un déboggueur □ que l'on veut voir tous les avertissements	<pre>{     printf("%c", i);</pre>
☐ n = factorielle(p, q);		}
13. Lorsqu'un programme utilise printf ou scanf il faut	17. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou	<pre>printf("\n");</pre>
qu'il contienne l'instruction préprocesseur :	référence indéfinie vers « prinft » que doit-	Alors l'affichage sera :
☐ #include <studio.h></studio.h>	on chercher dans le programme?	□ i
☐ #include <stdio.h></stdio.h>	<ul> <li>□ un caractère interdit en C</li> <li>□ une variable non déclarée</li> <li>□ une faute de frappe dans un appel de fonction</li> </ul>	□ сссссс
☐ #appart <stdlib.h></stdlib.h>		□ A
☐ #include <studlib.h></studlib.h>	une directive préprocesseur #include manquante	☐ ABCDEF
·		

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si a et b sont deux variables de type :
   struct toto_s
   {
     int n;
     double x;
   };
   Alors pour tester l'égalité de a et de b on utilise la
   condition:
     \Box (a.n == b.n) && (a.x == b.x)
     □ a == b
    \square a{n, x} == b{n, x}
     \Box a = b
2. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit:
     □ void afficher_date(date_s d);
     ☐ int afficher_date(date_s d);
     □ void afficher_date(struct date_s d);
    □ struct date_s afficher_date(struct date_s | d); 8. Le code suivant :
3. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \square 1.5
     \Box 0
     \Box 1
4. Une de ces manière de composer les blocs de pro-
   grammes ne fait pas partie des opérations de la pro-
   grammation structurée :
     □ répéter un bloc tant qu'une condition est vérifée
     \square mettre les blocs en séquence les uns à la suite des
        autres
     \square sélectionner entre deux blocs à l'aide d'une condi-
        tion
     \square retourner un bloc
```

```
5. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
    \square x = racine(2/3);
    \square x = racine(racine(x)*racine(x));
    \square x = racine(x * x) - racine(x);
    \square x - 1 = racine(x);
6. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un fichier texte
     □ changer de répertoire courant
     □ ouvrir un fichier texte
     □ créer un répertoire
7. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
  on chercher dans le programme?
     ☐ un point-virgule manquant
     □ un point-virgule en trop
     \square une accolade en trop
     \square une accolade manquante
    int age = 15;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
     □ Mineur
        Majeur
     □ Majeur
     ☐ Mineur
     □ rien
```

```
9. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
      □ dans lequel vous avez déclaré ces fonction
      □ un ordre quelconque
10. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il comporte une boucle infinie
      ☐ il risque d'afficher bonjour à la place de coucou
      □ il n'affiche rien
      \square il ne compile pas
11. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \square 0
      \Box 4
12. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
      □ printf("%d", factorielle(n));
      \square n = factorielle():
      \square n = factorielle(p, q);
```

☐ int factorielle(int 2);

```
13. Soit la fonction f définie par :
                                                            16. Pour l'extrait de programme suivant :
    int f(int a)
                                                                  int produit = 0;
                                                                  int serie[4] = \{2, 2, 2, 2\};
    {
                                                                  for (i = 0; i < 4; i = i + 1)
      printf("a = \n", %d);
      if (a > 0)
                                                                     produit = produit * serie[i];
        return f(a - 1) + 1;
                                                                                                                             b = b + 2;
                                                                  printf("produit = %d", produit);
      return 4;
                                                                La valeur affichée est :
                                                                  \Box 16
    Alors l'expression f(1) prendra la valeur :
                                                                  \Box 0
                                                                                                                              \Box C
      \Box 4
                                                                  \square 8
                                                                                                                              □ b
      \Box 0
                                                                  \Box 4
                                                                                                                              □ A
      \Box 1
                                                            17. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                              □В
      \Box 5
                                                              10
14. Pour déclarer une fonction exposant qui prend en ar-
                                                              11
                                                                   int main() {
    gument un réel x et un entier positif n et renvoie la
                                                              12
                                                                        int x = 5;
                                                                                                                            int g(int a)
    valeur de x^n on écrit :
                                                              13
                                                                                                                            {
                                                              14
                                                                        x = 3 * x + 1;
      \square double exposant(double x, int n);
                                                              15
                                                                                                                              if (1 > 0)
      \square exposant(double x, int n, int r);
                                                              16
                                                                        . . .
      \square int exposant(double n, int x);
                                                              17
                                                                   }
                                                                                                                                 return 5;
      \square void exposant(double x^n);
                                                                  \square le programme affiche x
                                                                                                                              return 7;
15. Pour compiler un programme prog.c, on utilise la
                                                                  □ la variable x vaut 16
    ligne de commande :
                                                                  \square la variable x vaut -\frac{1}{2}
                                                                                                                            Alors l'expression g(0) prendra la valeur :
      ☐ gcc -Wall prog.c -o prog.exe
                                                                  \square le programme affiche ****
                                                                                                                              \Box 0
      ☐ gcc prog.exe -Wall -o prog.c
                                                            18. Lorsqu'un programme utilise printf ou scanf il faut
     ☐ gcc prog.c -o -Wall prog.exe
                                                                                                                              \Box 5
                                                                qu'il contienne l'instruction préprocesseur :
      ☐ gcc -Wall prog.exe -o prog.c
                                                                                                                              \Box 7
                                                                  ☐ #include <studio.h>
```

```
☐ #include <studlib.h>
     ☐ #include <stdio.h>
     ☐ #appart <stdlib.h>
19. Au début de la fonction main() on place le code :
     char b = 'A';
    printf("%c\n", b);
   Alors l'affichage sera :
20. Soit la fonction g définie par :
     printf("a = \n", %d);
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

9. Pour compiler un programme prog.c, on utilise la

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Le langage C est un langage □ compilé □ lu, écrit, parlé □ interprété □ composé 2. Soit la fonction f définie par : int f(int a)  $printf("a = \n", %d);$ if (a > 0)return 3; return 4; Alors l'expression f(0) prendra la valeur :  $\Box$  4  $\Box$  0  $\square$  3 3. Soit la fonction g définie par : int g(int a)  $printf("a = \n", %d);$ if (1 > 0)return 5; return 7; Alors l'expression g(0) prendra la valeur :  $\Box$  0  $\square$  7  $\Box$  5

```
4. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square n = carre(n):
     \square int carre(2);
     \square int n = carre();
     \square n = carre(int n);
5. Les lignes
   int i;
   int x=0;
   for(i=0,i<5,i=i+1)
     x=x+1;
     □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
     □ comportent une erreur qui ne sera pas détectée
     \square ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
6. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
     \square !(!A \mid | B) == (A \&\& !B)
     \square (A == TRUE) && (B == TRUE)
     \square (!A || B)
     □ A && B
7. Le type des réels en C est :
     \square int
     □ double
     □ char
     \square real
8. Dans la commande gcc, l'option -Wall signifie :
     □ qu'on veut changer alétoirement de fond d'écran
     ☐ que l'on veut voir tous les avertissements
     □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
```

```
ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
10. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
     \square int pgcd(int x, int x);
     \Box int pgcd(int x, y);
     \square void pgcd(int x, int y);
     \Box int pgcd(int y, int x);
11. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i:
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 0
     \square 10
     \Box 6
      \square 15
12. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      □ un désaccord entre la déclaration et la définition
        d'une fonction
      □ une fonction déclarée mais non définie
     \Box une fonction appelée avant sa déclaration
      ☐ une directive préprocesseur #include manquante
```

13. Au début de la fonction main() on place le code :	$\Box$ une directive préprocesseur $\verb"#include"$ manquante	19. Le code suivant :
<pre>char i; for (i = 'A'; i &lt;= 'F'; i = i + 1) {     printf("%c", i); } printf("\n");</pre>	<ul> <li>□ une variable non déclarée</li> <li>□ une faute de frappe dans un appel de fonction</li> <li>□ un caractère interdit en C</li> <li>16. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :</li> <li>□ #include <studio.h></studio.h></li> </ul>	<pre>int i; for (i = 0; i &lt; 5; i = i + 1) {     printf("%d ", i); } printf("\n");</pre>
Alors l'affichage sera :	☐ #appart <stdlib.h></stdlib.h>	affichera:
□ cccccc □ A	☐ #include <stdio.h> ☐ #include <studlib.h></studlib.h></stdio.h>	$\Box \ 0 \ 1 \ 2 \ 3$
□i	17. Une variable booléenne est un variable :	$\Box \ 0\ 1\ 2\ 3\ 4$
□ ABCDEF	$\Box$ jamais nulle	$\square \ 4\ 3\ 2\ 1$
<ul><li>14. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?</li><li>□ int char;</li></ul>	<ul> <li>□ réelle positive</li> <li>□ à laquelle une valeur vient d'être affectée</li> <li>□ qui est vraie ou fausse</li> </ul>	$\hfill\Box$ 4 3 2 1 0
□ char 'c';	$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)	$\Box$ jouer de la musique
☐ char "c";	18. Si factorielle est une fonction prenant en entrée un	$\Box$ ouvir un bureau partagé (common desktop)
□ char c;	entier et renvoyant un entier, il est correct d'écrire :  □ int factorielle(int 2);	☐ détruire un fichier
15. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit-	<pre>□ n = factorielle(p, q);</pre> □ n = factorielle();	□ récupérer un programme arrêté avec la commande ab
on chercher dans le programme?	☐ printf("%d", factorielle(n));	$\Box$ changer de répertoire courant

т		-1
	100000	- 1
	acence	

# Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Le type des réels en C est : □ int. □ real ☐ double □ char 2. Un fichier source est: □ un document illisible pour les humains  $\square$  un fichier texte qui sera traduit en instructions processeur □ un document qui doit être protégé □ un document de référence du système ☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur 3. Vous utilisez une boucle while quand: □ vous n'avez pas déclaré de fonction □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance ☐ l'incrément de la variable de boucle n'est pas 1 □ vous avez déjà fait un for dans le même programme principal 4. Pour déclarer une procédure afficher\_menu sans argument et qui ne renvoie rien on utilise: ☐ double afficher\_menu(); ☐ int afficher\_menu(); ☐ int afficher\_menu(int char); ☐ char afficher\_menu(printf("menu")); □ void afficher\_menu(); 5. Le code suivant : int age = 20; if (age < 18) { printf("Mineur\n"); } else printf("Majeur\n"); }

```
affichera:
     □ rien
    ☐ Mineur
       Majeur
    □ Mineur
     □ Majeur
6. Le code suivant :
    int age = 18:
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
    □ Majeur
    \square rien
    □ Mineur
       Majeur
     □ Mineur
7. Sous unix (ou linux), la commande 1s permet de :
     □ afficher la liste de fichiers contenus dans un
       répertoire
    □ voir des clips musicaux
    \square afficher le contenu d'un fichier texte
     □ compiler un programme
8. Si cette erreur apparaît à la compilation :
  error: expected ';' before '}' token que doit-
  on chercher dans le programme?
    □ un point-virgule en trop
    ☐ une accolade manquante
     \square une accolade en trop
     ☐ un point-virgule manquant
```

```
9. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre:
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
      \square un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
10. Après exécution jusqu'à la ligne 15 du programme C:
  10
  11
        int main() {
  12
            int x = 5;
  13
  14
            x = 3 * x + 1;
  15
  16
  17
       }
      \square le programme affiche ****
      \square le programme affiche x
      \square la variable x vaut -\frac{1}{2}
      \square la variable x vaut 16
11. Si le code:
    struct toto s
      int n:
      double x;
    };
    précède la fonction main(), alors on peut écrire en
    début de main():
      \square toto_s n, x;
      \square toto_s struct z = {3, 0.5};
      \Box int struct toto_s = {3, -1e10};
      □ struct toto_s toto;
      \square int toto.n = 3;
```

```
12. Soit un programme contenant les lignes suivantes :
                                                             14. Si a et b sont deux variables de type :
                                                                                                                                 ☐ une directive préprocesseur #include manquante
                                                                                                                                 \square une fonction déclarée mais non définie
                                                                 struct toto_s
     int i = 0;
                                                                                                                                 \square un désaccord entre la déclaration et la définition
     int j = 0;
                                                                    int n;
     for (i = 0; i < 2; i = i + 1)
                                                                                                                                    d'une fonction
                                                                    double x;
                                                                                                                           17. L'écriture 111 en binaire correspond au nombre natu-
                                                                 };
         for (j = 0; j < 3; j = j + 1)
                                                                                                                               rel:
                                                                 Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                                 \Box 7
              printf("%d ", i);
                                                                 condition:
                                                                                                                                 \square 8
                                                                   \square (a.n == b.n) && (a.x == b.x)
     }
                                                                                                                                 \square 3
                                                                   \square a\{n, x\} == b\{n, x\}
                                                                   □ a == b
                                                                                                                                 □ 111
   qu'est ce qui sera affiché?
                                                                    \Box a = b
                                                                                                                           18. Un enregistrement permet de grouper plusieurs valeurs
      \Box 0 0 0 1 1 1
                                                             15. Soit la fonction f définie par :
                                                                                                                               dans:
      \Box 1 2 1 2 3
                                                                                                                                 □ ses cases
                                                                 int f(int a)
                                                                                                                                 \square ses chants
      \Box 0 1 2 0 1 2
                                                                    printf("a = \n", %d);
                                                                                                                                 \square ses champs
      \Box 0 1 0 1 0 1 0 1
                                                                    if (a > 0)
                                                                                                                                 \square ses blocs
13. Le code suivant :
                                                                      return f(a - 1) + 1;
                                                                                                                           19. Dans la commande gcc, l'option -Wall signifie :
     int i;
                                                                                                                                 □ qu'il faut indenter le fichier source
     for (i = 1; i < 5; i = i + 1)
                                                                    return 4:
                                                                                                                                 □ qu'on veut changer alétoirement de fond d'écran
         printf("%d ", i);
                                                                 Alors l'expression f(1) prendra la valeur :
                                                                                                                                 ☐ que l'on veut voir tous les avertissements
                                                                   \Box 0
                                                                                                                                 □ qu'il faut lancer un déboggueur
    printf("\n");
                                                                   \Box 4
                                                                                                                           20. Si factorielle est une fonction prenant en entrée un
   affichera:
                                                                   \Box 5
                                                                                                                               entier et renvoyant un entier, il est correct d'écrire :
      \Box 1 2 3 4
                                                                    \Box 1
                                                                                                                                 ☐ printf("%d", factorielle(n));
                                                             16. Si cette erreur apparaît à la compilation :
      \Box 01234
                                                                                                                                 \square n = factorielle(p, q);
                                                                 erreur: conflicting types for 'max', que doit-
      \square 4 3 2 1
                                                                                                                                 ☐ int factorielle(int 2);
                                                                 on chercher dans le programme?
      \Box \ 4\ 3\ 2\ 1\ 0
                                                                                                                                 \square n = factorielle();
                                                                   ☐ une fonction appelée avant sa déclaration
```

## Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Après exécution jusqu'à la ligne 15 du programme C :

```
10  int main() {
11    int x = 5;
12    int y = 3;
13
14    x = y;
15
16    ...
17  }
```

- $\square$  la variable x vaut 5 et la variable y vaut 3
- $\Box$ le programme affiche "Faux"
- $\Box$ la variable y vaut 5
- 2. Une segmentation fault est une erreur qui survient lorsque :
  - □ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur
  - □ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal
  - □ la division du programme en zones homogènes échoue
  - □ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée
- 3. Sous unix (ou linux), la commande cd permet de :
  - $\Box$  détruire un fichier
  - $\Box$ jouer de la musique
  - □ ouvir un bureau partagé (common desktop)
  - $\hfill \square$ récupérer un programme arrêté avec la commande ab
  - $\square$  changer de répertoire courant

- 4. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :
  - $\Box$  int pgcd(int x, int x);
  - $\square$  void pgcd(int x, int y);
  - □ int pgcd(int y, int x);
    □ int pgcd(int x, y);
- 5. Le code suivant :

```
int i;
for (i = 0; i < 5; i = i + 1)
{
    printf("%d ", i);
}
printf("\n");</pre>
```

- affichera:
- $\Box 01234$
- $\Box \ 4\ 3\ 2\ 1\ 0$
- $\square$  0 1 2 3
- $\square$  4 3 2 1
- 6. Après exécution du programme :
  - l lecture 8 r0
  - 2 valeur 3 r1
  - 3 mult r1 r0
  - 4 valeur 1 r2
  - 5 add r2 r0
  - 6 ecriture r0 8
- 7 stop
- 8 5
  - $\Box\,$ la case mémoire 8 contiendra 0
  - $\Box$  le bus explose
- $\square$  le terminal affiche 8
- $\square$  la case mémoire 8 contiendra 16
- 7. Soit la fonction g définie par :

```
int g(int a)
{
  printf("a = \n", %d);
  if (1 > 0)
  {
```

- ifs |
- Alors l'expression g(0) prendra la valeur :
- $\Box$  5
- □ 7
- $\Box$  0
- 8. Quel est le problème d'un programme comportant les lignes suivantes?
  - while (1)
    {
     printf("coucou\n");
    }

return 5:

return 7;

- $\Box$ il ne compile pas
- $\square$  il n'affiche rien
- $\Box\:$ il comporte une boucle infinie
- $\Box$ il risque d'afficher bonjour à la place de coucou
- 9. Un fichier source est:
  - $\Box$  un document de référence du système
  - □ un document qui doit être protégé
  - $\square$  un document illisible pour les humains
  - $\Box$  un fichier texte qui sera traduit en instructions processeur
  - $\hfill \square$  un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
- 10. Le code suivant :

```
int age = 20;
if (age < 18)
{
    printf("Mineur\n");
}
printf("Majeur\n");</pre>
```

- affichera:
- ☐ Mineur Majeur
- □ Mineur
- $\square$  rien
- $\square$  Majeur

11. Dans la commande gcc, l'option -Wall signifie :  ☐ qu'il faut indenter le fichier source ☐ que l'on veut voir tous les avertissements ☐ qu'on veut changer alétoirement de fond d'écran ☐ qu'il faut lancer un déboggueur	<pre>14. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :</pre>	17. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit- on chercher dans le programme?  ☐ une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration ☐ un désaccord entre la déclaration et la définition
<pre>□ qu'il faut lancer un déboggueur</pre> 12. Soit le programme principal suivant : int main() {   int a = 3;   int b = 5;   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);   return EXIT_SUCCESS; } appelant la fonction f ainsi définie : int f(int a, int b) {   a = a + b;   return a; } L'affichage dans le main est le suivant : □ f(a,b)=8, a=8, b=5 □ f(3,5)=8, a=3, b=5 □ f(a,b)=13, a=8, b=5  □ f(a,b)=13, a=8, b=5  13. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses blocs	<pre> □ int factorielle(); □ struct int factorielle(int n);  15. Après la déclaration : int mccarthy(int n);, il est correct d'écrire : □ n = mccarthy(); □ x = mccarthy(n); □ n = mccarthy(p, q); □ int mccarthy(int 2);  16. Soit un programme contenant les lignes suivantes :     int i = 0;     int j = 0;     for (i = 0; i &lt; 0; i = i + 1)     {         for (j = 0; j &lt; 5; j = j + 1)         {          }     }     printf("j = %d\n", j);      }  qu'est ce qui sera affiché? □ j = 4 </pre>	
□ ses champs □ ses cases □ ses chants	$ \Box j = 0 $ $ \Box j = 5 $ $ \Box j = %d $	<pre>□ Int saisie_utilisateur(); □ void saisie_utilisateur(int n); □ saisie_utilisateur(scanf(%d)); □ void saisie_utilisateur(char c);</pre>

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	
N° INE :	

Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un bit est:  $\square$  un battement d'horloge processeur  $\square$  la longueur d'un mot mémoire  $\square$  un chiffre binaire (0 ou 1) ☐ l'instruction qui met fin à un programme 2. Sous unix (ou linux), la commande 1s permet de :  $\square$  afficher le contenu d'un fichier texte  $\square$  compiler un programme  $\square$  voir des clips musicaux  $\square$  afficher la liste de fichiers contenus dans un répertoire 3. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #appart <stdlib.h> ☐ #include <studlib.h> ☐ #include <stdio.h> ☐ #include <studio.h> 4. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C? ☐ char "c":  $\Box$  char c;  $\square$  int char; □ char 'c'; 5. Un registre du processeur est : □ une unité de calcul spécialisée de l'ordinateur □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs  $\square$  une gamme de fréquence de fonctionnement du processeur  $\square$  un composant qui contient la liste des fichiers du système

```
6. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il comporte une boucle infinie
    \square il ne compile pas
    □ il n'affiche rien
    ☐ il risque d'afficher bonjour à la place de coucou
7. L'écriture 111 en binaire correspond au nombre natu-
  rel:
    \square 8
    \square 7
    \square 3
    □ 111
8. Soit la fonction g définie par :
  int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
  Alors l'expression g(0) prendra la valeur :
    \square 7
    \Box 5
    \Box 0
9. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
    □ scanf("%d", &n);
    □ printf("Valeur de n ? %d\n", n);
    □ printf("Valeur de n ? %g\n", n);
```

 $\square$  un débogueur

```
10. Avant de faire appel à une fonction il est nécessaire
   de:
      □ l'avoir définie
      □ l'avoir déclarée
      □ avoir défini une constante symbolique de la taille
        de cette fonction
      □ l'avoir déclarée et définie
11. Pour l'extrait de programme suivant :
     int i = 0;
     int i = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
              printf("%d ", j);
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 3 0 1 2
     \Box 0 1 2 0 1 2 3
12. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 16
     \square 20
      \Box 6
      \square 3
```

13.	Un enregistrement permet de grouper plusieurs valeurs	16. Le code suivant :
	dans:	int i;
	□ ses cases	for (i = 0; i
	$\square$ ses blocs	{
	$\square$ ses chants	printf("%d
	$\square$ ses champs	}
14.	Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :	<pre>printf("\n"); affichera:</pre>
	☐ int factorielle(int 2);	$\Box \ 0\ 1\ 2\ 3\ 4\ 5\ 6$
	□ n = factorielle();	$\Box 0123456$
	☐ printf("%d", factorielle(n));	$\square 02468$
	$\square$ n = factorielle(p, q);	$\square 0246$
15.	Soit la fonction f définie par :	
	<pre>int f(int a)</pre>	17. Le code suivant :
	{	int i;
	<pre>printf("a = \n", %d);</pre>	for (i = 1; i
	if (a > 0)	{
	{	printf("%d
	return 3;	}
	}	<pre>printf("\n");</pre>
	return 4;	affichera:
	}	$\square \ 4\ 3\ 2\ 1$
	Alors l'expression f(0) prendra la valeur :	$\square$ 1 2 3 4
		_
	$\square$ 3	$\square \ 4\ 3\ 2\ 1\ 0$
	$\Box$ 4	$\Box 01234$

```
< 7; i = i + 2)
(d ", i);
5 7
< 5; i = i + 1)
¼d ", i);
```

18. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :
$\hfill \square$ répéter un bloc tant qu'une condition est vérifée
$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition
$\hfill\Box$ retourner un bloc
$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres
19. Pour compiler un programme prog.c, on utilise la ligne de commande :
$\square$ gcc -Wall prog.exe -o prog.c
$\square$ gcc prog.exe -Wall -o prog.c
$\square$ gcc prog.c -o -Wall prog.exe
$\square$ gcc -Wall prog.c -o prog.exe
20. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit on chercher dans le programme?
$\square$ un désaccord entre la déclaration et la définition d'une fonction
$\square$ une fonction déclarée mais non définie
$\hfill \square$ une directive préprocesseur $\#\mbox{include}$ man quante
$\square$ une fonction appelée avant sa déclaration

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Licence 1 Barème : 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un fichier source est: □ un document illisible pour les humains □ un fichier texte qui sera traduit en instructions processeur ☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur  $\Box\,$ un document qui doit être protégé □ un document de référence du système 2. Soit la fonction g définie par : int g(int a) {  $printf("a = \n", %d);$ if (1 > 0)return 5; return 7; Alors l'expression g(0) prendra la valeur :  $\square$  7

- 3. Sous unix (ou linux), la commande cd permet de :
  - $\Box\,$ jouer de la musique
  - $\hfill \square$ récupérer un programme arrêté avec la commande ab
  - $\Box\,$  changer de répertoire courant
  - $\Box$  détruire un fichier
  - $\Box$ ouvir un bureau partagé (common desktop)
- 4. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :

```
□ n = mccarthy();
□ n = mccarthy(p, q);
□ int mccarthy(int 2);
□ x = mccarthy(n);
```

5. Au début de la fonction main() on place le code : char i;

```
for (i = 'A'; i <= 'F'; i = i + 1)
{
   printf("%c", i);
}</pre>
```

Alors l'affichage sera :

printf("\n");

- $\square$  A
- $\square$  ccccc
- ☐ ABCDEF
- $\square$  i

6. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

- $\square$  char c;
- □ char "c";
- ☐ int char;
- □ char 'c';

7. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :

- $\hfill\Box$  struct int factorielle(int n);
- $\square$  int factorielle(double n);
- $\Box$  int factorielle(int x);
- $\square$  int factorielle();

8. Si *n* est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :

- $\Box$  un débogueur
- □ scanf("%d", &n);
- $\Box$  printf("Valeur de n ? %d\n", n);
- ☐ printf("Valeur de n ? %g\n", n);

9. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

- $\square$  void saisie\_utilisateur(int n);
- ☐ int saisie\_utilisateur();
- $\hfill\Box$  void saisie\_utilisateur(char c);
- ☐ saisie\_utilisateur(scanf(%d));

 $10. \ \, \text{Si}$  cette erreur apparaı̂t à la compilation :

erreur: conflicting types for 'max', que doiton chercher dans le programme?

- $\square$  une fonction appelée avant sa déclaration
- □ une directive préprocesseur #include manquante
   □ une fonction déclarée mais non définie
- ☐ un désaccord entre la déclaration et la définition d'une fonction
- 11. Si a et b sont deux variables de type :

```
struct toto_s
{
  int n;
  double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition :

- □ a == b
- $\square$  a{n, x} == b{n, x}
- $\square$  a = b

}

 $\square$  (a.n == b.n) && (a.x == b.x)

12. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :

- $\Box$  printf("x=%x et y=%y\n");
- $\Box$  printf("x=%d et y=%d\n",x,y);
- $\Box$  printf("x=%d et y=%d\n",x y);
- $\square$  printf("x=%d et y=%d\n,x,y");
- $13.\ {\rm Soit}$  un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);
...</pre>
```

```
qu'est ce qui sera affiché?
      \Box j = 4
     \Box j = 0
      \Box j = 5
     \Box j = %d
14. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
    appelant la fonction f ainsi définie :
    int f(int a, int b)
    {
      a = a + b;
      return a;
    L'affichage dans le main est le suivant :
      \Box f(a,b)=8, a=8, b=5
      \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
15. Quel est l'opérateur de différence en C :
      □ <>
      □!
      □ !=
      \square \neq
```

```
16. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il ne compile pas
      □ il n'affiche rien
      \square il comporte une boucle infinie
      □ il risque d'afficher bonjour à la place de coucou
17. Sur unix (ou linux), la commande mkdir permet de :
      \square créer un répertoire
      \square changer de répertoire courant
      □ ouvrir un fichier texte
      □ créer un fichier texte
18. Après exécution jusqu'à la ligne 14 du programme C:
       int main() {
  10
            int x = 5;
  11
  12
            printf(" x = %d\n", 2);
  13
  14
  15
       }
  16
      \square le terminal affiche x = 5
      □ le terminal affiche "Faux"
      \square le terminal affiche 5
      \square le terminal affiche x = 2
```

```
19. Le code suivant :
     int age = 18;
     if (age < 18)
          printf("Mineur\n");
     else
     {
          printf("Majeur\n");
     }
    affichera:
      \square rien
      ☐ Mineur
        Majeur
      □ Majeur
      \square Mineur
20. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
     \Box int pgcd(int x, y);
      \square void pgcd(int x, int y);
      \square int pgcd(int y, int x);
```

 $\square$  int pgcd(int x, int x);

т.	-1
Licence	

### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?

```
☐ A && B
☐ (A == TRUE) && (B == TRUE)
☐ !(!A || B) == (A && !B)
```

2. Pour l'extrait de programme suivant :

 $\square$  (!A || B)

```
int i = 0;
int j = 0;
for (i = 0; i < 2; i = i + 1)
{
    for (j = 0; j < 3; j = j + 1)
    {
        printf("%d ", j);
    }
}</pre>
```

qu'est ce qui sera affiché?

```
□ 0 0 1 1 2 2 3
```

 $\square$  0 1 2 3 0 1 2

 $\Box$  0 1 2 0 1 2 3

□ 0 1 2 0 1 2

3. Au début de la fonction main() on place le code :

```
char b = 'A';
b = b + 2;
printf("%c\n", b);
```

Alors l'affichage sera :

С

□ A

	b

□В

```
4. Vous utilisez une boucle while quand :
    ☐ l'incrément de la variable de boucle n'est pas 1
    □ vous n'avez pas déclaré de fonction
    \Box vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
    □ vous avez déjà fait un for dans le même pro-
       gramme principal
5. Si le code :
  struct toto s
     int n;
     double x;
  }:
  précède la fonction main(), alors on peut écrire en
  début de main() :
    \square int toto.n = 3;
    \square toto_s struct z = {3, 0.5};
    \square int struct toto_s = {3, -1e10};
    □ struct toto_s toto;
    \square toto_s n, x;
6. Le code suivant :
   int age = 20;
   if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Mineur
    □ rien
    □ Majeur
    ☐ Mineur
       Majeur
7. Avant de faire appel à une fonction il est nécessaire
  de:
```

 $\Box$  avoir défini une constante symbolique de la taille

□ l'avoir définie

□ l'avoir déclarée

de cette fonction

□ l'avoir déclarée et définie

```
8. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      \square une fonction appelée avant sa déclaration
      □ une fonction déclarée mais non définie
      \square un désaccord entre la déclaration et la définition
         d'une fonction
      ☐ une directive préprocesseur #include manquante
 9. L'ordonnancement par tourniquet permet :
      □ d'afficher des ronds colorés à l'écran
     \square de ne pas perdre de temps avec la commutation
         de contexte
      □ d'entretenir l'illusion que les processus tournent
         en parallèle
      □ de doubler la mémoire disponible
10. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
    une variable entière définie et initialisée, il est correct
    d'écrire :
     \square int n = carre();
     \square n = carre(int n):
     \square int carre(2);
      \square n = carre(n);
11. Le code suivant :
     int age = 20;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
    affichera:
      □ Majeur
      ☐ Mineur
         Majeur
      □ rien
```

□ Mineur

```
12. Une variable booléenne est un variable :
     ☐ réelle positive
     □ jamais nulle
     \square qui est vraie ou fausse
     □ NaN (not a number, qui n'est pas un nombre)
     □ à laquelle une valeur vient d'être affectée
13. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
   {
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
14. Le code suivant :
    int i;
    for (i = 4; i > 0; i = i - 1)
```

```
printf("%d ", i);
     }
     printf("\n");
    affichera:
      \square 0 1 2 3
      \Box 01234
      \square 4 3 2 1
      \Box 4 3 2 1 0
15. La virtualisation de la mémoire permet notamment de
    stocker des portions inactives de la mémoire de travail
    sur le disque dur. Mais on perd :
      □ certaines données de la mémoire de travail
      □ les fichiers du disque
      \square des processus
      \square en temps d'accès
16. Le bus système sert à :
      ☐ Écrire des données sur le dique dur
      □ transporter les processus du tourniquet au pro-
         cesseur
      ☐ Arriver à l'heure en cours
      ☐ Transférer des données et intructions entre pro-
         cesseur et mémoire
17. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
      \square une accolade manquante
      □ un point-virgule en trop
      ☐ un point-virgule manquant
      \square une accolade en trop
```

```
18. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
      \square x - 1 = racine(x);
     \Box x = racine(racine(x)*racine(x));
      \square x = racine(2/3);
      \square x = racine(x * x) - racine(x);
19. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
         return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 0
      \Box 5
      \square 7
20. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse syntaxique
```

□ analyse sémantique

 $\square$  analyse harmonique

 $\square$  analyse lexicale

### Éléments d'informatique – contrôle continue

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N}^{\circ} \text{ INE}: \end{array}$ 

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :
```

```
□ int factorielle(int 2);
□ n = factorielle();
□ n = factorielle(p, q);
□ printf("%d", factorielle(n));
```

2. Soit la fonction g définie par :

```
int g(int a)
{
  printf("a = \n", %d);
  if (1 > 0)
  {
    return 5;
  }
  return 7;
```

Alors l'expression g(0) prendra la valeur :

```
\begin{array}{c}
\square \ 7 \\
\square \ 5 \\
\square \ 0
\end{array}
```

3. Soit la fonction f définie par :

```
int f(int a)
{
  printf("a = \n", %d);
  if (a > 0)
  {
    return f(a - 1) + 1;
  }
  return 4;
}
```

Alors l'expression f(1) prendra la valeur :

 $\square$  1

```
4.\  \, {\rm Soit} un programme contenant les lignes suivantes :
```

```
int i = 0;
int j = 0;
for (i = 0; i < 3; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);</pre>
```

qu'est ce qui sera affiché par ce printf?

```
\Box j = 4
\Box j = %d
\Box j = 5
```

□ j = 0

5. Si cette erreur apparaît à la compilation :

error: expected ';' before '}' token que doiton chercher dans le programme?

```
□ un point-virgule en trop
```

```
\square une accolade manquante
```

```
\Box une accolade en trop
```

 $\Box$  un point-virgule man quant

6. Après exécution jusqu'à la ligne 15 du programme C:

```
10  int main() {
11    int x = 5;
12    int y = 3;
13
14    x = y;
15
16    ...
17  }
□ la variable x vaut 5 et la variable y vaut 3
```

- $\Box\,$ la variable x vaut 3
- $\square$  la variable y vaut 5
- □ le programme affiche "Faux"

7. Quel est le problème d'un programme comportant les lignes suivantes?

```
while (1)
{
  printf("coucou\n");
}
```

- $\square$  il ne compile pas
- $\square$  il risque d'afficher bonjour à la place de coucou
- $\square$  il n'affiche rien
- $\Box\,$ il comporte une boucle infinie

8. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?

```
□ A && B
□ (!A || B)
```

9. Le code suivant :

```
int age = 15;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");
}</pre>
```

affichera:

rien

Maj	eur

Mineu
Maion

```
10. Le code suivant :
                                                              15. Le code suivant :
                                                                                                                                qu'est ce qui sera affiché?
     int i;
                                                                   int somme = 0:
                                                                                                                                  \Box i = 5
     for (i = 4; i > 0; i = i - 1)
                                                                   int i:
                                                                                                                                  \Box j = 0
                                                                   for (i = 1; i < 4; i = i + 1)
                                                                                                                                  \Box j = 4
         printf("%d ", i);
                                                                      somme = somme + i;
                                                                                                                                  \Box i = %d
     printf("\n");
                                                                                                                            18. Si n est une variable entière pour demander sa valeur
                                                                   printf("%d", somme);
    affichera:
                                                                                                                                à l'utilisateur, on utilise plutôt :
      \square 4 3 2 1
                                                                                                                                  □ un débogueur
                                                                  affichera:
      \Box 4 3 2 1 0
                                                                    \square 42
                                                                                                                                  ☐ printf("Valeur de n ? %d\n", n);
      \Box \ 0\ 1\ 2\ 3\ 4
                                                                    \Box 0
                                                                                                                                  □ printf("Valeur de n ? %g\n", n);
      \square 0 1 2 3
                                                                    \Box 6
                                                                                                                                   □ scanf("%d", &n);
11. Laquelle des analyses suivantes ne fait pas partie des
                                                                    \square 1
    étapes de la compilation :
                                                                                                                            19. Le code suivant :
                                                              16. Les lignes
      \square analyse lexicale
                                                                                                                                  int i;
      \square analyse harmonique
                                                                  int i;
                                                                                                                                  for (i = 0; i < 7; i = i + 2)
                                                                  int x=0;
      □ analyse sémantique
                                                                  for(i=0,i<5,i=i+1)
      \square analyse syntaxique
                                                                                                                                      printf("%d ", i);
12. L'écriture 101 en binaire correspond au nombre natu-
                                                                    x=x+1;
                                                                                                                                  printf("\n");
    rel:
      \Box 5
                                                                                                                                affichera:
                                                                    □ comportent une erreur qui sera détectée au cours
      \Box 4
                                                                       de l'édition de lien
                                                                                                                                  \Box 02468
      \square 3
                                                                    □ comportent une erreur qui ne sera pas détectée
                                                                                                                                  \Box 0 1 2 3 4 5 6 7
      \square 101
                                                                    □ comportent une erreur qui sera détectée au cours
                                                                                                                                  \Box 0 1 2 3 4 5 6
13. Vous avez déclaré préalablement un ensemble de fonc-
                                                                       de l'analyse syntaxique
                                                                                                                                  \square 0 2 4 6
    tions utilisées par votre programme principal. L'ordre
                                                                    \square ne comportent aucune erreur
    dans lequel vous devez maintenant définir ces fonctions
                                                                                                                            20. Au début de la fonction main() on place le code :
                                                              17. Soit un programme contenant les lignes suivantes :
    est l'ordre :
                                                                   int i = 0;
                                                                                                                                  char i;
      □ un ordre quelconque
                                                                                                                                  for (i = 'A'; i \le 'F'; i = i + 1)
                                                                   int j = 0;
      \square dans lequel ces fonctions sont appelées dans le
                                                                   for (i = 0; i < 0; i = i + 1)
         main
                                                                                                                                    printf("%c", i);
      \square dans lequel vous avez déclaré ces fonction
                                                                        for (j = 0; j < 5; j = j + 1)
                                                                                                                                 printf("\n");
      □ alphabétique
                                                                        {
14. Un enregistrement permet de grouper plusieurs valeurs
                                                                                                                                Alors l'affichage sera :
                                                                        }
    dans:
                                                                                                                                  \square A
      \square ses blocs
                                                                   printf("j = %d\n", j);
                                                                                                                                  □ cccccc
      \square ses chants
                                                                                                                                  ☐ ABCDEF
      \square ses champs
                                                                   }
                                                                                                                                   \square i
      □ ses cases
```

т		-1
	acence	

### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Un fichier source est:
  - ☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
  - □ un document de référence du système
  - □ un document qui doit être protégé
  - $\square$  un document illisible pour les humains
  - $\Box$  un fichier texte qui sera traduit en instructions processeur
- 2. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction
  - $\square$  int loop n;
  - $\square$  int %d;
  - $\square$  int k;
  - □ loop i;
- 3. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
  - $\square$  en temps d'accès
  - $\square$  des processus
  - $\square$  les fichiers du disque
  - □ certaines données de la mémoire de travail
- 4. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);
...
}</pre>
```

qu'est ce qui sera affiché?

```
\Box j = 4
    \Box i = %d
    \Box j = 5
    \Box j = 0
5. L'ordonnancement par tourniquet permet :
    ☐ d'entretenir l'illusion que les processus tournent
       en parallèle
    \square de ne pas perdre de temps avec la commutation
       de contexte
    ☐ de doubler la mémoire disponible
    ☐ d'afficher des ronds colorés à l'écran
6. Soit le programme principal suivant :
   int main()
   {
   int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(3.5)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
```

7. L'écriture 111 en binaire correspond au nombre natu-

 $\Box$  7

rel:

- $\square$  3
- $\square$  8
- □ 111

```
8. Dans la commande gcc, l'option -Wall signifie :

qu'il faut indenter le fichier source
que l'on veut voir tous les avertissements
qu'on veut changer alétoirement de fond d'écran
qu'il faut lancer un déboggueur

9. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les ex-
```

```
pressions booléennes suivantes, laquelle a pour valeur TRUE?

□ (A == TRUE) && (B == TRUE)

□ (!A || B)

□ A && B
```

10. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :

- □ n = carre(int n);
  □ int carre(2);
  □ n = carre(n);
  □ int n = carre();
- 11. Pour l'extrait de programme suivant :

 $\square$  !(!A || B) == (A && !B)

```
int produit = 1;
int serie[4] = {2, 2, 2, 2};
for (i = 0; i < 4; i = i + 1)
{
   produit = produit * serie[i];
}
printf("produit = %d", produit);</pre>
```

La valeur affichée est :

- □ 4 □ 16
- 12. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
□ printf("%d", factorielle(n));
□ n = factorielle();
□ n = factorielle(p, q);
□ int factorielle(int 2):
```

```
13. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
      \Box 0
     \square 3
14. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%d et y=%d\n,x,y");
     \Box printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n",x y);
15. Soit la fonction f définie par :
   int f(int a)
   {
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
```

```
return 4;
   Alors l'expression f(1) prendra la valeur :
     \Box 4
     \Box 5
     \Box 0
     \Box 1
16. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
    du struct, on écrit :
     □ void afficher_date(date_s d);
     ☐ struct date_s afficher_date(struct date_s |d);
     □ void afficher_date(struct date_s d);
     ☐ int afficher_date(date_s d);
17. Un bit est:
     ☐ l'instruction qui met fin à un programme
     \square un battement d'horloge processeur
     □ la longueur d'un mot mémoire
     \square un chiffre binaire (0 ou 1)
18. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
```

```
printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box i = 0
     \Box j = 5
     \Box j = 4
     \Box j = %d
19. Au début de la fonction main() on place le code :
    for (i = 'A'; i <= 'F'; i = i + 1)
       printf("%c", i);
     printf("\n");
   Alors l'affichage sera:
     ☐ ABCDEF
     \Box i
     □ ccccc
      □ A
20. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \square int pgcd(int y, int x);
     \Box int pgcd(int x, int x);
     \square void pgcd(int x, int y);
```

 $\Box$  int pgcd(int x, y);

### Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit la fonction g définie par :
   int g(int a)
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 0
    \Box 5
    \Box 7
2. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
    else
        printf("Majeur\n");
   affichera:
    □ rien
    □ Majeur
     ☐ Mineur
     ☐ Mineur
       Majeur
3. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
```

```
{
        }
    }
   printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
    \Box j = 5
    \Box j = 0
    \Box j = 4
    \Box j = %d
4. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
  pressions booléennes suivantes, laquelle a pour valeur
  TRUE?
    \square (!A | | B)
    □ A && B
    \square !(!A || B) == (A && !B)
    \square (A == TRUE) && (B == TRUE)
5. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
    \square 3
    \Box 4
     \Box 0
6. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
    ☐ #define taille = N
    □ #define N 3
    \square #define N = 3
     ☐ #define taille = 3
```

```
7. Le code suivant :
     int somme = 0;
     int i;
     for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
     printf("%d", somme);
   affichera:
     \Box 6
     \Box 1
      \Box 0
     \square 42
 8. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
     \square void exposant(double x^n);
     \square double exposant(double x, int n);
      \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
 9. Un registre du processeur est :
      □ une unité de calcul spécialisée de l'ordinateur
      □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
      □ une gamme de fréquence de fonctionnement du
         processeur
      \square un composant qui contient la liste des fichiers du
        système
10. Une variable booléenne est un variable :
      □ à laquelle une valeur vient d'être affectée
      ☐ jamais nulle
      ☐ réelle positive
      □ NaN (not a number, qui n'est pas un nombre)
      □ qui est vraie ou fausse
```

```
11. Le code suivant :
     int i;
     for (i = 4; i \ge 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 4 3 2 1
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 1 2 3 4
      \Box 0 1 2 3 4
12. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit:
      □ void afficher_date(struct date_s d);
     □ void afficher_date(date_s d);
     ☐ int afficher_date(date_s d);
     □ struct date_s afficher_date(struct date_s d);
13. Le type des réels en C est :
     \square real
      □ char
      ☐ double
      \square int
14. Vous utilisez une boucle while quand :
      □ vous avez déjà fait un for dans le même pro-
        gramme principal
      ☐ l'incrément de la variable de boucle n'est pas 1
      \square vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
      □ vous n'avez pas déclaré de fonction
```

```
15. Sous unix (ou linux), la commande cd permet de :
     □ ouvir un bureau partagé (common desktop)
     \square détruire un fichier
     □ jouer de la musique
     ☐ récupérer un programme arrêté avec la commande
      □ changer de répertoire courant
16. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ créer un répertoire
     □ ouvrir un fichier texte
     □ créer un fichier texte
17. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
     ☐ int factorielle(int 2);
     \square n = factorielle():
     ☐ printf("%d", factorielle(n));
      \square n = factorielle(p, q);
18. Le bus système sert à :
     ☐ Arriver à l'heure en cours
     □ transporter les processus du tourniquet au pro-
        cesseur
      ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     ☐ Écrire des données sur le dique dur
19. Soit un programme contenant les lignes suivantes :
     int i = 0:
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
     {
         for (j = 0; j < 5; j = j + 1)
```

```
{
         }
    }
    printf("j = %d\n", j);
    }
   qu'est ce qui sera affiché?
     \Box i = 0
     \Box j = 5
     \Box j = 4
     \Box j = %d
20. Soit le programme principal suivant :
   int main()
   ₹
    int a = 3;
     int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
   {
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
```

 $\Box$  f(a,b)=13, a=8, b=5

 $\Box$  f(3,5)=8, a=3, b=5

### Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
1. Après exécution jusqu'à la ligne 15 du programme C :
```

```
10 ...
11 int main() {
12 int x = 5;
13
14 x = 3 * x + 1;
15
16 ...
17 }
```

- $\Box$  la variable x vaut  $-\frac{1}{2}$
- □ la variable x vaut 16
- $\Box$ le programme affiche \*\*\*\*
- $\Box$ le programme affiche x
- 2. Soit le programme principal suivant :

```
int main()
{
  int a = 3;
  int b = 5;
  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
  return EXIT_SUCCESS;
}
appelant la fonction f ainsi définie:
  int f(int a, int b)
{
    a = a + b;
    return a;
}
```

L'affichage dans le main est le suivant :

```
☐ f(a,b)=13, a=8, b=5
☐ f(a,b)=8, a=8, b=5
☐ f(a,b)=8, a=3, b=5
☐ f(3,5)=8, a=3, b=5
```

- 3. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :
  - $\square$  void saisie\_utilisateur(int n);
  - ☐ saisie\_utilisateur(scanf(%d));
  - ☐ int saisie\_utilisateur();
  - □ void saisie\_utilisateur(char c);
- 4. Un programme en langage C doit comporter une et une seule définition de la fonction :
  - $\square$  init
  - □ begin
  - $\square$  include
  - $\square$  main
- 5. Si a et b sont deux variables de type :

```
struct toto_s
{
  int n;
  double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition :

- $\square$  a{n, x} == b{n, x}
- $\square (a.n == b.n) \&\& (a.x == b.x)$
- $\square$  a == b
- $\Box$  a = b
- 6. Le code suivant :

```
int age = 18;
if (age < 18)
{
    printf("Mineur\n");
}
else
{
    printf("Majeur\n");
}</pre>
```

affichera:

 $\square$  Mineur

- $\square$  rien
- ☐ Mineur Majeur
- □ Majeur
- 7. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :
  - $\Box$ mettre les blocs en séquence les uns à la suite des autres
  - $\Box\,$ répéter un bloc tant qu'une condition est vérifée
  - $\hfill \square$  sélectionner entre deux blocs à l'aide d'une condition
  - $\square$  retourner un bloc
- 8. Le code suivant :

```
int somme = 0;
int i;
for (i = 1; i < 4; i = i + 1)
{
   somme = somme + i;
}
printf("%d", somme);</pre>
```

affichera:

- $\Box$  0
- $\square$  42
- $\Box$  1
- $\Box$  6
- 9. Vous utilisez une boucle while quand :
  - $\Box$  l'incrément de la variable de boucle n'est pas 1
  - $\square$  vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
  - □ vous avez déjà fait un for dans le même programme principal
  - □ vous n'avez pas déclaré de fonction

```
10. Le code suivant :
                                                            14. Au début de la fonction main() on place le code :
                                                                                                                            Alors l'expression f(0) prendra la valeur :
     int i;
                                                                 char i;
                                                                                                                              \square 3
                                                                 for (i = 'A'; i \le 'F'; i = i + 1)
     for (i = 4; i \ge 0; i = i - 1)
                                                                                                                              \Box 4
                                                                                                                              \Box 0
         printf("%d ", i);
                                                                   printf("%c", i);
                                                                                                                        17. Pour déclarer une fonction factorielle qui prend en
    printf("\n");
                                                                 printf("\n");
                                                                                                                            argument un entier et renvoie sa factorielle on écrit :
   affichera:
                                                                Alors l'affichage sera :
                                                                                                                              ☐ int factorielle(int x);
                                                                  ☐ ABCDEF
     \square 1 2 3 4
                                                                  \Box i
                                                                                                                              ☐ struct int factorielle(int n);
     \Box 01234
                                                                  □ cccccc
     \Box 43210
                                                                                                                              ☐ int factorielle();
                                                                  □ A
     \square 4 3 2 1
                                                                                                                              ☐ int factorielle(double n);
                                                            15. Soit un programme contenant les lignes suivantes :
11. Dans la commande gcc, l'option -Wall signifie :
                                                                                                                        18. Après la déclaration : int mccarthy(int n);, il est
      ☐ que l'on veut voir tous les avertissements
                                                                 int i = 0;
                                                                                                                            correct d'écrire :
                                                                 int i = 0;
      □ qu'il faut indenter le fichier source
                                                                 for (i = 0; i < 3; i = i + 1)
                                                                                                                              \square n = mccarthy();
     □ qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut lancer un déboggueur
                                                                                                                             \square x = mccarthy(n);
                                                                     for (j = 0; j < 5; j = j + 1)
12. Avant de faire appel à une fonction il est nécessaire
                                                                                                                              \Box int mccarthy(int 2);
   de:
                                                                                                                              \square n = mccarthy(p, q);
      □ l'avoir définie
                                                                     }
                                                                                                                        19. On considère deux variables booléennes A et B initia-
     □ l'avoir déclarée
                                                                 printf("j = %d\n", j);
                                                                                                                            lisées à TRUE et FALSE respectivement. Parmi les ex-
     □ avoir défini une constante symbolique de la taille
                                                                                                                            pressions booléennes suivantes, laquelle a pour valeur
        de cette fonction
                                                                                                                            TRUE?
                                                                qu'est ce qui sera affiché par ce printf?
      □ l'avoir déclarée et définie
                                                                 \Box j = %d
                                                                                                                              □ A && B
13. Si le code :
                                                                 \Box j = 5
                                                                                                                              \square !(!A || B) == (A && !B)
   struct toto s
                                                                  \Box i = 0
                                                                                                                              \square (A == TRUE) && (B == TRUE)
                                                                  \Box j = 4
      int n;
                                                                                                                              \square (!A | | B)
                                                            16. Soit la fonction f définie par :
      double x;
                                                                int f(int a)
                                                                                                                        20. Le bus système sert à :
   précède la fonction main(), alors on peut écrire en
                                                                                                                              ☐ Écrire des données sur le dique dur
   début de main() :
                                                                  printf("a = \n", %d);
                                                                                                                              ☐ Arriver à l'heure en cours
                                                                  if (a > 0)
     ☐ struct toto_s toto;
                                                                  {
     \square toto_s n, x;
                                                                                                                              ☐ Transférer des données et intructions entre pro-
                                                                    return 3:
     \Box int struct toto_s = {3, -1e10};
                                                                                                                                 cesseur et mémoire
     \Box toto_s struct z = {3, 0.5};
                                                                                                                              □ transporter les processus du tourniquet au pro-
                                                                  return 4;
     \Box int toto.n = 3;
                                                                                                                                 cesseur
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme? □ un point-virgule en trop □ un point-virgule manquant  $\square$  une accolade en trop  $\square$  une accolade manquante 2. Quel est l'opérateur de différence en C : □ !=  $\sqcap \iff$  $\Box$  !  $\Box \neq$ 3. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : □ analyse sémantique  $\square$  analyse harmonique  $\square$  analyse lexicale  $\square$  analyse syntaxique 4. Le code suivant : int i: for (i = 1; i < 5; i = i + 1)printf("%d ", i); } printf("\n"); affichera: $\Box 01234$  $\Box$  4 3 2 1 0  $\square$  4 3 2 1  $\Box 1234$ 5. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  int carre(2);  $\square$  n = carre(int n):  $\square$  int n = carre();

 $\square$  n = carre(n):

```
6. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
    \square n = factorielle();
    \square n = factorielle(p, q);
    ☐ printf("%d", factorielle(n));
    ☐ int factorielle(int 2);
7. Après exécution jusqu'à la ligne 15 du programme C:
 10
 11
      int main() {
 12
           int x = 5;
13
 14
           x = 3 * x + 1;
 15
 16
     }
 17
    \square le programme affiche x
    \square la variable x vaut 16
    ☐ le programme affiche ****
    \square la variable x vaut -\frac{1}{2}
8. Une segmentation fault est une erreur qui survient
  lorsque:
     □ le programme tente d'afficher des caractères sur
        une ligne qui va au delà de la largeur de la fenêtre
        du terminal
    □ la division du programme en zones homogènes
        échoue
     □ le programme source a été enregistré sur le disque
        dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
    □ le programme tente d'accèder à une partie de la
        mémoire qui ne lui est pas réservée
9. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il comporte une boucle infinie
```

```
□ il n'affiche rien
      \square il ne compile pas
      \square il risque d'afficher bonjour à la place de coucou
10. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
      □ un ordre quelconque
      \square dans lequel ces fonctions sont appelées dans le
         main
      □ alphabétique
      □ dans lequel vous avez déclaré ces fonction
11. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
         return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \square 7
      \Box 5
      \Box 0
12. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
          for (j = 0; j < 5; j = j + 1)
     printf("j = %d\n", j);
     }
```

```
qu'est ce qui sera affiché?
      \Box j = 5
     \Box j = 0
     \Box j = %d
     \Box j = 4
13. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \Box 0
      \square 3
      \Box 4
14. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      ☐ l'avoir définie
      ☐ l'avoir déclarée
```

```
15. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
     ☐ (A == TRUE) && (B == TRUE)
     \square !(!A \mid | B) == (A \&\& !B)
     □ A && B
     \square (!A || B)
16. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \Box int pgcd(int x, int x);
     \Box int pgcd(int x, y);
     \Box int pgcd(int y, int x);
     \square void pgcd(int x, int y);
17. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     ☐ printf("Valeur de n ? %d\n", n);
     ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
     □ un débogueur
18. Le code suivant :
     int i;
    for (i = 4; i \ge 0; i = i - 1)
```

```
printf("%d ", i);
     printf("\n");
   affichera:
     \Box \ 4\ 3\ 2\ 1\ 0
     \Box 01234
     \square 4 3 2 1
     \square 1 2 3 4
19. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt :
     \square for(i=0;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
      \square for(i=0;i<5;i=i+1)
20. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle();
     ☐ int factorielle(double n);
      □ struct int factorielle(int n);
      ☐ int factorielle(int x);
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	
1, 11,12,	

Bar pai

réme : 1 points par réponse juste (unique); $-0,5$ points réponse fausse. Durée : 20 minutes.			
1.	Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :		
	$\Box$ dans lequel vous avez déclaré ces fonction		
	$\square$ un ordre quel conque		
	$\square$ alphabétique		
	$\Box$ dans lequel ces fonctions sont appelées dans le main		
2.	Le code suivant :		
	<pre>int i; for (i = 4; i &gt;= 0; i = i - 1) {     printf("%d ", i); }</pre>		
	<pre>printf("\n");</pre>		
	affichera :		
	$\Box \ 0\ 1\ 2\ 3\ 4$		
	$\square \ 4\ 3\ 2\ 1$		
	$\square \ 1 \ 2 \ 3 \ 4$		
	$\Box \ 4\ 3\ 2\ 1\ 0$		
3.	L'écriture $\underline{101}$ en binaire correspond au nombre naturel :		
	$\square$ 3		
	$\Box$ 5		
	□ 101		
	$\Box$ 4		
4.	Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :		
	□ mkdir TP4		
	$\square$ kwrite TP4		
	□ neu TP4		

☐ yppasswd

```
5. Une segmentation fault est une erreur qui survient
   lorsque:
     \square le programme source a été enregistré sur le disque
        dur en plusieurs morceaux et l'un d'entre eux ne
        peut pas être chargé par le compilateur
     □ le programme tente d'afficher des caractères sur
        une ligne qui va au delà de la largeur de la fenêtre
        du terminal
     □ la division du programme en zones homogènes
        échoue
     □ le programme tente d'accèder à une partie de la
        mémoire qui ne lui est pas réservée
6. Un registre du processeur est :
     □ une gamme de fréquence de fonctionnement du
        processeur
     \square une unité de calcul spécialisée de l'ordinateur
     □ un composant qui contient la liste des fichiers du
        système
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
7. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
  une variable réelle définie et initialisée, il est incorrect
  d'écrire :
    \square x - 1 = racine(x);
    \Box x = racine(x * x) - racine(x);
    \square x = racine(2/3);
     \square x = racine(racine(x)*racine(x));
8. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses chants
     \square ses blocs
     \square ses cases
```

```
9. Le code suivant :
     int i;
     for (i = 4; i > 0; i = i - 1)
          printf("%d ", i);
     printf("\n");
    affichera:
     \square 4 3 2 1
      \Box \ 4\ 3\ 2\ 1\ 0
      \square 0 1 2 3
      \Box 01234
10. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 7
      \square 0
      \Box 5
11. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
    renvoie cet entier on écrit:
      □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
      □ void saisie_utilisateur(char c);
      ☐ int saisie_utilisateur();
12. Après exécution jusqu'à la ligne 15 du programme C :
  10
       int main() {
  11
  12
            int x = 5:
  13
```

```
x = 3 * x + 1:
 14
 15
 16
       }
 17
      □ la variable x vaut 16
      \Box la variable x vaut -\frac{1}{2}
      \square le programme affiche x
      \square le programme affiche ****
13. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
         somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
    La valeur de somme affichée est :
      \square 10
      \Box 6
      \Box 0
      \square 15
14. On souhaite faire une boucle de contrôle de saisie : tant
    que l'entier n n'appartient pas à l'intervalle [a..b], on
    recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
```

int n;

```
scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
    Quelle est la condition cond :
     \Box (a<n) || (n>b)
     \square (n<=a) && (n<=b)
     □ (a<=n) && (n<=b)
     □ a<=n<=b
15. Vous utilisez une boucle while quand:
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     ☐ l'incrément de la variable de boucle n'est pas 1
16. Le code suivant :
     int i:
    for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
     }
    printf("\n");
   affichera:
     \Box 02468
     \Box 8 6 4 2 0
     \square 8 2
      \square 8 6 4 2
```

```
17. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      ☐ int factorielle(double n);
     ☐ int factorielle();
      \square int factorielle(int x);
      □ struct int factorielle(int n);
18. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square x = mccarthy(n);
      \square int mccarthy(int 2);
     \square n = mccarthy();
     \square n = mccarthy(p, q);
19. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      \square init
      □ main
      □ begin
      □ include
20. Dans la commande gcc, l'option -Wall signifie :
      □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
     \square que l'on veut voir tous les avertissements
```

□ qu'on veut changer alétoirement de fond d'écran

### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

	eponee masser Baros v 20 mmares.
1.	Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
	☐ int pgcd(2);
	☐ int n = pgcd();
	$\square \ n = pgcd(n, 3);$
	$\square = pgcd(int p, int q);$
2	
۷.	Au début de la fonction main() on place le code :
	char i;
	for (i = 'A'; i <= 'F'; i = i + 1) {
	printf("%c", i);
	}
	<pre>printf("\n");</pre>
	Alors l'affichage sera :
	□ ABCDEF
	Πi
	 □ A
3.	La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :
	$\square$ en temps d'accès
	☐ les fichiers du disque
	☐ certaines données de la mémoire de travail
	☐ des processus
4	
4.	Quel est l'opérateur de différence en C :
	<del>- ·</del>
	□ != -
5.	Un enregistrement permet de grouper plusieurs valeurs
	dans:
	□ ses chants
	□ ses cases
	□ ses blocs
	$\square$ ses champs

```
6. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(racine(x)*racine(x));
    \Box x = racine(x * x) - racine(x);
    \square x = racine(2/3);
     \square x - 1 = racine(x);
7. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 0
     \Box 4
8. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <studlib.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
9. Le code suivant :
    int i;
    for (i = 8; i > 0; i = i - 2)
    {
        printf("%d ", i);
    printf("\n");
   affichera:
    \Box 8 6 4 2 0
    \Box 02468
     \square 8 2
     \Box 8 6 4 2
```

```
10. Soit la fonction f définie par :
   int f(int a)
   {
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \square 5
      \Box 0
      \Box 4
      \Box 1
11. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      □ 111
      \square 8
      \square 3
      \square 7
12. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
      □ dans lequel ces fonctions sont appelées dans le
         main
      □ un ordre quelconque
      \square dans lequel vous avez déclaré ces fonction
      □ alphabétique
13. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
    avec la commande :
     \square printf("x=%d et y=%d\n",x,y);
     \Box printf("x=%d et y=%d\n,x,y");
     \Box printf("x=%d et y=%d\n",x y);
      \square printf("x=%x et y=%y\n");
```

```
14. Si le code :
    struct toto_s
    {
      int n;
      double x;
    };
    précède la fonction main(), alors on peut écrire en
    début de main() :
     \Box int struct toto_s = {3, -1e10};
     \square toto_s n, x;
     \Box toto_s struct z = {3, 0.5};
     \square int toto.n = 3;
     □ struct toto_s toto;
15. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ int char;
     ☐ char "c";
     □ char 'c';
     □ char c;
16. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
    on chercher dans le programme?
     □ une directive préprocesseur #include manquante
     ☐ une fonction appelée avant sa déclaration
```

```
\Box un désaccord entre la déclaration et la définition
         d'une fonction
      \square une fonction déclarée mais non définie
17. Le code suivant :
     int somme = 0;
     int i;
     for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
     printf("%d", somme);
    affichera:
      \Box 1
      \Box 0
      \Box 6
      \square 42
18. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
```

<pre>printf("Majeur\n");</pre>
}
affichera:
□ rien
☐ Mineur
☐ Majeur
□ Mineur Majeur
19. Sous unix (ou linux), la commande 1s permet de :
$\square$ voir des clips musicaux
□ afficher la liste de fichiers contenus dans un répertoire
$\hfill\Box$ afficher le contenu d'un fichier texte
$\square$ compiler un programme
20. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
$\Box$ un point-virgule man quant
$\Box$ un point-virgule en trop
$\Box$ une accolade en trop
□ une accolade manquante

т.	-1
Licence	

# ${\rm \acute{E}l\acute{e}ments}\ d'in \underline{formatique} - \underline{contr\^{o}le}\ continue$

Prénom:	Nom ·
	TOIL .
N° INE ·	
11 11111 .	

Barème : 1 points par réponse juste (unique) ; $-0,5$ points par réponse fausse. Durée : 20 minutes.	6. Sous unix (ou linux), la commande cd permet de :  □ ouvir un bureau partagé (common desktop)	11. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs
1. Le type des réels en C est :	☐ récupérer un programme arrêté avec la commande	passés en arguments on écrit :
☐ double	ab □ détruire un fichier	☐ int pgcd(int x, y); ☐ int pgcd(int x, int x);
□ char	☐ jouer de la musique	□ void pgcd(int x, int y);
□ real	☐ changer de répertoire courant	☐ int pgcd(int y, int x);
□ int	7. Si cette erreur apparaît à la compilation :	12. Sous unix (ou linux), la commande 1s permet de :
2. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :	Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit-on chercher dans le programme?  ☐ une directive préprocesseur #include manquante ☐ une variable non déclarée	<ul> <li>□ afficher le contenu d'un fichier texte</li> <li>□ afficher la liste de fichiers contenus dans un répertoire</li> <li>□ voir des clips musicaux</li> </ul>
$\hfill\Box$ dans lequel vous avez déclaré ces fonction	□ un caractère interdit en C	$\square$ compiler un programme 13. Au début de la fonction main() on place le code :
$\Box$ alphabétique	$\Box$ une faute de frappe dans un appel de fonction	char i;
$\Box$ dans lequel ces fonctions sont appelées dans le main	8. Si x est une variable réelle (de type double) alors $x=3/2$ lui affecte la valeur :	for (i = 'A'; i <= 'F'; i = i + 1) {
$\square$ un ordre quelconque		<pre>printf("%c", i);</pre>
3. Un enregistrement permet de grouper plusieurs valeurs dans :		<pre>} printf("\n");</pre>
□ ses champs		Alors l'affichage sera : $\Box$ i
□ ses blocs	9. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :	☐ ABCDEF
ses cases	□ analyse lexicale	□ A
	$\square$ analyse harmonique	
□ ses chants	$\Box$ analyse sémantique	14. Soient deux variables entières x et y initialisées à 4 et
4. L'écriture <u>101</u> en binaire correspond au nombre naturel :	☐ analyse syntaxique  10. Après exécution jusqu'à la ligne 15 du programme C :	5 respectivement. L'affichage $x=4$ et $y=5$ est obtenu avec la commande :
$\Box$ 3	10 int main() {	$\Box$ printf("x=%d et y=%d\n",x y);
□ 101	11 int $x = 5$ ;	$\Box$ printf("x=%d et y=%d\n,x,y");
$\Box$ 4	12 int y = 3; 13	$\Box \text{ printf("x=\%d et y=\%d\n",x,y);}$
□ 5	14 x = y;	☐ printf("x=%x et y=%y\n");
5. Un programme en langage C doit comporter une et une seule définition de la fonction :	15 16 17 }	15. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?
$\Box$ include	☐ la variable y vaut 5	une directive préprocesseur #include manquante
□ main	☐ la variable x vaut 5 et la variable y vaut 3	une fonction appelée avant sa déclaration
□ begin	$\Box$ la variable x vaut 3	<ul> <li>□ une fonction déclarée mais non définie</li> <li>□ un désaccord entre la déclaration et la définition</li> </ul>
$\Box$ init	$\Box$ le programme affiche "Faux"	d'une fonction

```
16. Après exécution jusqu'à la ligne 14 du programme C :
                                                                  \Box la longueur d'un mot mémoire
                                                                  \square un chiffre binaire (0 ou 1)
       int main() {
  10
                                                            18. Pour l'extrait de programme suivant :
  11
            int x = 5;
  12
                                                                  int i = 0;
            printf(" x = %d\n", 2);
  13
                                                                  int j = 0;
  14
                                                                  for (i = 0; i < 2; i = i + 1)
  15
      }
  16
                                                                      for (j = 0; j < 3; j = j + 1)
     \square le terminal affiche x = 5
                                                                           printf("%d ", j);
     \square le terminal affiche x = 2
                                                                      }
      □ le terminal affiche "Faux"
                                                                qu'est ce qui sera affiché?
      \square le terminal affiche 5
                                                                  \Box 0 1 2 0 1 2
17. Un bit est:
                                                                   \Box 0 1 2 3 0 1 2
     \square un battement d'horloge processeur
                                                                  \Box 0 0 1 1 2 2 3
     ☐ l'instruction qui met fin à un programme
                                                                   \Box 0 1 2 0 1 2 3
```

### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     \square il comporte une boucle infinie
     □ il n'affiche rien
     \square il ne compile pas
     ☐ il risque d'afficher bonjour à la place de coucou
2. Le langage C est un langage
     □ interprété
     □ compilé
     □ lu, écrit, parlé
     \square composé
3. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
     ☐ int factorielle(int 2):
     □ printf("%d", factorielle(n));
     □ n = factorielle();
     \square n = factorielle(p, q);
4. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0: i < 2: i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
    }
   qu'est ce qui sera affiché?
     \Box 0 1 2 3 0 1 2
     \Box 0 1 2 0 1 2 3
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 0 1 2
```

```
5. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square n = mccarthy();
     \Box int mccarthy(int 2);
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses cases
     □ ses chants
     \square ses champs
     \square ses blocs
7. Si carre est une fonction prenant en entrée un en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \square int carre(2);
     \square n = carre(int n);
     \square int n = carre():
     \square n = carre(n);
8. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     ☐ kwrite TP4
     □ new TP4
     ☐ yppasswd
     ☐ mkdir TP4
9. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     \square main
     \square include
     \square init
     □ begin
```

```
10. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre:
      \square dans lequel vous avez déclaré ces fonction
      □ un ordre quelconque
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
11. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 4
      \square 8
      \Box 0
      \square 16
12. Si racine est une fonction prenant en entrée un réel
    et renvoyant la racine carrée de cet réel, et que x est
    une variable réelle définie et initialisée, il est incorrect
    d'écrire :
     \square x = racine(2/3);
     \Box x = racine(x * x) - racine(x);
     \square x = racine(racine(x)*racine(x));
     \square x - 1 = racine(x);
13. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
```

return 7;

Alors l'expression g(0) prendra la valeur :	$\square$ 4	19. L'ordonnancement par tourniquet permet :
□ 7	$\square$ 3	
$\Box$ 0	□ 101	☐ de ne pas perdre de temps avec la commutation
$\square$ 5	16. Pour compiler un programme prog.c, on utilise la	de contexte
14. Le code suivant :	ligne de commande :	$\square$ de doubler la mémoire disponible
int age = 15;	☐ gcc prog.exe -Wall -o prog.c	$\hfill \Box$ d'afficher des ronds colorés à l'écran
if (age < 18)	☐ gcc -Wall prog.exe -o prog.c	☐ d'entretenir l'illusion que les processus tournent
{	☐ gcc prog.c -o -Wall prog.exe	en parallèle
<pre>printf("Mineur\n"); }</pre>	☐ gcc -Wall prog.c -o prog.exe	
else	17. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu	20. Le code suivant :
{	avec la commande :	int i;
<pre>printf("Majeur\n");</pre>	<pre>□ printf("x=%d et y=%d\n,x,y");</pre>	for $(i = 0; i < 5; i = i + 1)$
}	$\Box$ printf("x=%d et y=%d\n",x y);	{
off shape	$\square$ printf("x=%x et y=%y\n");	<pre>printf("%d ", i);</pre>
affichera:	$\Box$ printf("x=%d et y=%d\n",x,y);	<pre>printf("\n");</pre>
□ Majeur □ rien	18. Si cet avertissement apparaît à la compilation :	<del>-</del>
□ Mineur	warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	' affichera:
□ mineur Majeur	, que don-on chercher dans le programme :  ☐ une directive préprocesseur #include manquante	$\Box \ 0\ 1\ 2\ 3$
□ Mineur	un désaccord entre la déclaration et la définition	$\Box \ 4\ 3\ 2\ 1\ 0$
15. L'écriture 101 en binaire correspond au nombre natu-	d'une fonction	
rel:	$\hfill\Box$ une fonction déclarée mais non définie	$\Box \ 0\ 1\ 2\ 3\ 4$
$\Box$ 5	$\hfill\Box$ une fonction appelée avant sa déclaration	$\square \ 4\ 3\ 2\ 1$

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

1. Un registre du processeur est : □ une gamme de fréquence de fonctionnement du processeur □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs □ un composant qui contient la liste des fichiers du système  $\square$  une unité de calcul spécialisée de l'ordinateur 2. Soit un programme contenant les lignes suivantes : int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1)for (j = 0; j < 5; j = j + 1)}  $printf("j = %d\n", j);$ qu'est ce qui sera affiché par ce printf?  $\Box$  j = 5  $\Box$  j = %d  $\Box$  j = 4  $\Box$  j = 0 3. Les lignes int i; int x=0; for(i=0,i<5,i=i+1) { x=x+1; □ comportent une erreur qui sera détectée au cours de l'édition de lien □ comportent une erreur qui sera détectée au cours de l'analyse syntaxique  $\square$  ne comportent aucune erreur

□ comportent une erreur qui ne sera pas détectée

```
4. Vous avez déclaré préalablement un ensemble de fonc-
   tions utilisées par votre programme principal. L'ordre
   dans lequel vous devez maintenant définir ces fonctions
   est l'ordre:
     □ dans lequel vous avez déclaré ces fonction
    □ alphabétique
     □ un ordre quelconque
     \square dans lequel ces fonctions sont appelées dans le
        main
5. Si factorielle est une fonction prenant en entrée un
   entier et renvoyant un entier, il est correct d'écrire :
    ☐ printf("%d", factorielle(n));
    \square n = factorielle(p, q);
     ☐ int factorielle(int 2);
     \square n = factorielle();
6. Soit la fonction g définie par :
   int g(int a)
   {
     printf("a = \n", %d);
     if (1 > 0)
       return 5;
     return 7;
   Alors l'expression g(0) prendra la valeur :
     \Box 5
     \Box 0
     \square 7
7. Le code suivant :
    int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
```

_	
	affichera:
	☐ Mineur
	□ rien
	□ Majeur
	☐ Mineur
	Majeur
	Un fichier source est :
	$\Box$ un document illisible pour les humains
	$\square$ un document de référence du système
	$\hfill \square$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\Box$ un document qui doit être protégé
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	$\Box$ analyse harmonique
	$\square$ analyse syntaxique
	$\Box$ analyse sémantique
	$\Box$ analyse lexicale
	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	$\Box$ une accolade en trop
	$\Box$ un point-virgule en trop
	$\Box$ un point-virgule man quant
	$\Box$ une accolade man quante
	On souhaite faire une boucle de contrôle de saisie : tant que l'entier ${\tt n}$ n'appartient pas à l'intervalle $[ab]$ , on recommence la saisie de ${\tt n}$ . Soit le programme suivant :
	<pre>int a = 0; int b = 20; int n; scanf("%d", &amp;n); while(cond) {     scanf("%d", &amp;n);</pre>

9

10

11

}

Quelle est la condition cond :	□ A	18. Si <i>n</i> est une variable entière pour demander sa valeur
$\square (a < n) \mid \mid (n > b)$		à l'utilisateur, on utilise plutôt :
□ a<=n<=b	☐ ABCDEF	$\square$ scanf("%d", &n);
□ (n<=a) && (n<=b)	□i	$\square$ un débogueur
☐ (a<=n) && (n<=b)	15. Un programme en langage C doit comporter une et une	☐ printf("Valeur de n ? %d\n", n);
12. Pour déclarer une fonction saisie_utilisateur qui	seule définition de la fonction :	$\square$ printf("Valeur de n ? %g\n", n);
demande à l'utilisateur d'entrer un entier au clavier et	$\square$ main	19. Si cet avertissement apparaît à la compilation :
renvoie cet entier on écrit :	$\square$ init	warning: implicit declaration of function 'max'
☐ void saisie_utilisateur(int n);	$\square$ include	, que doit-on chercher dans le programme?
☐ int saisie_utilisateur();	$\square$ begin	$\Box$ une fonction appelée avant sa déclaration
☐ void saisie_utilisateur(char c);	16. Au début de la fonction main() on place le code :	$\hfill \square$ une directive préprocesseur $\# include$ manquante
☐ saisie_utilisateur(scanf(%d));	char b = 'A';	un désaccord entre la déclaration et la définition
13. L'écriture 111 en binaire correspond au nombre natu-	b = b + 2;	d'une fonction
rel:	<pre>printf("%c\n", b);</pre>	$\square$ une fonction déclarée mais non définie
□ 7	Alors l'affichage sera :	20. Le code suivant :
$\Box$ 3	□ С	int i;
□ 111	□В	for $(i = 4; i > 0; i = i - 1)$
□ 8	□ъ	{
	□ A	<pre>printf("%d ", i);</pre>
14. Au début de la fonction main() on place le code :	17. Vous utilisez une boucle while quand:	<pre>} printf("\n");</pre>
char i; for (i = 'A'; i <= 'F'; i = i + 1)	$\hfill\Box$ l'incrément de la variable de boucle n'est pas 1	affichera:
101 (1 - 'A'; 1 <- 'F'; 1 - 1 + 1)	$\square$ vous n'avez pas déclaré de fonction	
<pre>printf("%c", i);</pre>	$\square$ vous ne connaissez pas le nombre d'itérations de	$\Box \ 4\ 3\ 2\ 1\ 0$
}	la boucle à l'avance	$\square \ 4\ 3\ 2\ 1$
<pre>printf("\n");</pre>	□ vous avez déjà fait un for dans le même pro-	$\Box \ 0\ 1\ 2\ 3$
Alors l'affichage sera :	gramme principal	$\Box \ 0\ 1\ 2\ 3\ 4$

### Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un registre du processeur est : □ une gamme de fréquence de fonctionnement du processeur □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs  $\square$  un composant qui contient la liste des fichiers du système  $\Box$  une unité de calcul spécialisée de l'ordinateur 2. Vous utilisez une boucle while quand: □ vous n'avez pas déclaré de fonction □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance ☐ l'incrément de la variable de boucle n'est pas 1 □ vous avez déjà fait un for dans le même programme principal 3. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n"); ☐ il risque d'afficher bonjour à la place de coucou  $\square$  il ne compile pas  $\square$  il n'affiche rien  $\square$  il comporte une boucle infinie 4. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :  $\square$  n = factorielle(p, q); ☐ int factorielle(int 2); □ printf("%d", factorielle(n));  $\square$  n = factorielle(); 5. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\square$  1.5  $\square$  0.5  $\square$  0  $\Box$  1

```
6. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
    \square analyse syntaxique
    □ analyse sémantique
    \square analyse lexicale
    \square analyse harmonique
7. Les lignes
  int i;
  int x=0:
  for(i=0,i<5,i=i+1)
  {
     x=x+1;
     □ ne comportent aucune erreur
    □ comportent une erreur qui ne sera pas détectée
    □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
     □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
8. Pour l'extrait de programme suivant :
     int produit = 0;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
  La valeur affichée est :
    \square 16
    \Box 0
    \square 8
    \Box 4
9. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un fichier texte
    □ ouvrir un fichier texte
    □ créer un répertoire
     □ changer de répertoire courant
```

```
10. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 8
      \Box 4
      \Box 16
      \Box 0
11. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
 11
            int x = 5;
 12
            int y;
 13
 14
            y = x;
 15
 16
 17
      }
      \square la variable x vaut 0
      ☐ le programme affiche "Faux"
      \square la variable y vaut 5
      \square la variable x vaut 5 et la variable y vaut 0
12. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      }
      return 4;
   Alors l'expression f(0) prendra la valeur :
      \Box 4
      \square 3
      \square 0
```

```
13. Un enregistrement permet de grouper plusieurs valeurs
                                                                  Alors pour tester l'égalité de a et de b on utilise la
    dans:
                                                                  condition:
      \square ses chants
                                                                    \square (a.n == b.n) && (a.x == b.x)
      \square ses cases
                                                                    \square a\{n, x\} == b\{n, x\}
      \square ses blocs
                                                                    □ a == b
      \square ses champs
                                                                     \Box a = b
                                                              16. Le code suivant :
14. Le code suivant :
                                                                   int age = 20;
     int i;
     for (i = 0; i < 5; i = i + 1)
                                                                   if (age < 18)
                                                                   {
     {
                                                                        printf("Mineur\n");
          printf("%d ", i);
     printf("\n");
                                                                   printf("Majeur\n");
    affichera:
                                                                  affichera:
                                                                    \square Mineur
      \square 0 1 2 3
                                                                       Majeur
      \square 4 3 2 1
                                                                    □ Majeur
      \Box \ 4\ 3\ 2\ 1\ 0
                                                                    \square rien
      \Box 0 1 2 3 4
                                                                     \square Mineur
15. Si a et b sont deux variables de type:
                                                              17. Soient deux variables entières x et y initialisées à 4 et
    struct toto_s
                                                                  5 respectivement. L'affichage x=4 et y=5 est obtenu
                                                                  avec la commande :
      int n;
                                                                                                                                   \square 8
                                                                    \Box printf("x=%d et y=%d\n",x,y);
      double x;
                                                                                                                                   □ 111
                                                                    \Box printf("x=%d et y=%d\n",x y);
    };
```

$\Box$ printf("x=%d et y=%d\n,x,y");
$\Box$ printf("x=%x et y=%y\n");
18. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :
$\Box$ int pgcd(int x, int x);
$\square$ void pgcd(int x, int y);
$\square$ int pgcd(int x, y);
$\square$ int pgcd(int y, int x);
19. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :
$\square$ n = mccarthy(p, q);
☐ int mccarthy(int 2);
$\square$ n = mccarthy();
$\square$ x = mccarthy(n);
20. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
□ 7
$\square$ 3

т		-1
	100000	- 1
	acence	

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. On souhaite faire une boucle de contrôle de saisie : tant que l'entier n n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
    int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
      scanf("%d", &n);
  Quelle est la condition cond :
     \square (a<=n) && (n<=b)
     □ a<=n<=b
     \square (a<n) || (n>b)
     \square (n<=a) && (n<=b)
2. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
     \square printf("Valeur de n ? %g\n", n);
     □ un débogueur
     □ scanf("%d", &n);
```

3. Le type des réels en C est :

real
int
double
char

4. L'écriture 101 en binaire correspond au nombre naturel:

□ printf("Valeur de n ? %d\n", n);

```
\Box 4
\square 3
\Box 5
\square 101
```

```
5. L'ordonnancement par tourniquet permet :
     ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
     □ d'afficher des ronds colorés à l'écran
     ☐ de doubler la mémoire disponible
     ☐ de ne pas perdre de temps avec la commutation
        de contexte
6. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \Box 7
     \square 3
     \square 8
     \square 111
7. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
     {
        return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \Box 0
     \square 3
8. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #appart <stdlib.h>
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
     ☐ #include <studlib.h>
9. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
     \square ses chants
```

 $\square$  ses champs

□ ses cases

```
10. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
    avec la commande :
     \Box printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n,x,y");
     \square printf("x=%d et y=%d\n",x y);
     \square printf("x=%d et y=%d\n",x,y);
11. Pour l'extrait de programme suivant :
      int produit = 1;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \square 8
      \square 16
      \Box 0
      \Box 4
12. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square void exposant(double x^n);
     \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
      \square double exposant(double x, int n);
13. Pour l'extrait de programme suivant :
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
      for (i = 0; i < 4; i = i + 1)
        somme = somme + serie[i];
      printf("somme = %d",somme);
   La valeur de somme affichée est :
      \square 3
      \square 20
      \Box 16
```

 $\Box$  6

```
14. Au début de la fonction main() on place le code :
                                                                 \Box 1 2 1 2 3
     char i;
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                                 \Box 0 1 0 1 0 1 0 1
                                                                 \Box 0 1 2 0 1 2
       printf("%c", i);
                                                                 \Box 0 0 0 1 1 1
    printf("\n");
   Alors l'affichage sera:
     \square A
                                                                 ☐ char "c";
     ☐ ABCDEF
                                                                 □ char 'c';
     \square i
                                                                 □ char c;
     □ cccccc
                                                                 ☐ int char;
15. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     \square ouvrir un fichier texte
                                                               est l'ordre :
     \square créer un fichier texte
                                                                 \square un ordre quelconque
     □ changer de répertoire courant
16. Soit un programme contenant les lignes suivantes :
                                                                 □ alphabétique
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
                                                                int i = 0;
              printf("%d ", i);
                                                                int j = 0;
    }
```

```
qu'est ce qui sera affiché?
                                                                       {
                                                                       }
                                                                  }
                                                                  printf("j = %d\n", j);
                                                                  }
17. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
                                                                 qu'est ce qui sera affiché?
                                                                   \Box j = %d
                                                                   \Box j = 0
                                                                   \Box j = 5
                                                                   \Box j = 4
18. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
                                                             20. Pour l'extrait de programme suivant :
    dans lequel vous devez maintenant définir ces fonctions
                                                                   int produit = 0;
                                                                   int serie[4] = \{2, 2, 2, 2\};
                                                                   for (i = 0; i < 4; i = i + 1)
      \square dans lequel vous avez déclaré ces fonction
                                                                     produit = produit * serie[i];
      \square dans lequel ces fonctions sont appelées dans le
                                                                   printf("produit = %d", produit);
                                                                 La valeur affichée est :
19. Soit un programme contenant les lignes suivantes :
                                                                   \square 16
                                                                   \square 8
     for (i = 0; i < 0; i = i + 1)
                                                                   \Box 0
                                                                   \Box 4
          for (j = 0; j < 5; j = j + 1)
```

### Éléments d'informatique – contrôle continue

 $\square$  1

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour l'extrait de programme suivant :
    int i;
    int j;
    for(i=4;i>0;i=i-1)
      for(j=i;j<6;j=j+1)
        printf("*");
      printf(" ");
  qu'est ce qui sera affiché?
      **** **** ****
2. Après exécution jusqu'à la ligne 15 du programme C:
10
      int main() {
11
12
           int x = 5;
13
 14
          x = 3 * x + 1;
15
 16
17
     ☐ le programme affiche ****
     \square la variable x vaut 16
    \square la variable x vaut -\frac{1}{2}
     \square le programme affiche x
3. Le code suivant :
    int age = 18;
    if (age < 18)
        printf("Mineur\n");
    }
    else
```

```
{
         printf("Majeur\n");
    }
   affichera:
     □ Majeur
     \square rien
     □ Mineur
     □ Mineur
        Majeur
4. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
     □ chacun son tour, après que le processus précédent
        a terminé
     \square tous ensemble
     □ en parallèle, chacun dans un registre
     □ tour à tour, un petit peu à chaque fois
5. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
    \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
     \square x = racine(x * x) - racine(x);
     \square x = racine(2/3);
6. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise :
     ☐ int afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu(int char);
     ☐ double afficher_menu();
     □ void afficher_menu();
7. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \square 0.5
     \Box 0
```

```
8. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
    appelant la fonction f ainsi définie :
    int f(int a, int b)
      a = a + b:
      return a;
   L'affichage dans le main est le suivant :
      \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
      \Box f(a,b)=8, a=8, b=5
 9. Soit la fonction g définie par :
    int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      ₹
        return 5;
      return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 5
      \Box 7
      \square 0
10. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
    passés en arguments on écrit :
     \Box int pgcd(int x, int x);
      \square int pgcd(int y, int x);
     \Box int pgcd(int x, y);
```

 $\square$  void pgcd(int x, int y);

```
11. Soit un programme contenant les lignes suivantes :
                                                               affichera:
                                                                                                                      17. Si cette erreur apparaît à la compilation :
                                                                                                                          error: expected ';' before '}' token que doit-
                                                                 \Box 4 3 2 1 0
     int i = 0;
                                                                                                                          on chercher dans le programme?
                                                                 \square 1 2 3 4
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
                                                                 \square 4 3 2 1
                                                                                                                            \square une accolade en trop
                                                                 \Box 01234
                                                                                                                            \square une accolade manquante
         for (j = 0; j < 5; j = j + 1)
                                                           14. Un enregistrement permet de grouper plusieurs valeurs
                                                                                                                            □ un point-virgule en trop
                                                               dans:
                                                                                                                             ☐ un point-virgule manquant
                                                                 \square ses blocs
                                                                 \square ses cases
                                                                                                                      18. Pour déclarer une procédure afficher_date qui prend
                                                                 \square ses champs
                                                                                                                           en argument un struct date_s et affiche le contenu
    printf("j = %d\n", j);
                                                                                                                          du struct, on écrit :
                                                                 \square ses chants
     }
                                                           15. Pour l'extrait de programme suivant :
                                                                                                                            □ void afficher_date(date_s d);
                                                                 int produit = 0;
                                                                                                                            ☐ struct date_s afficher_date(struct date_s d);
                                                                 int serie[4] = \{2, 2, 2, 2\};
   qu'est ce qui sera affiché?
                                                                                                                            ☐ int afficher_date(date_s d);
                                                                 for (i = 0; i < 4; i = i + 1)
     \Box i = 5
                                                                                                                            □ void afficher_date(struct date_s d);
     \Box j = %d
                                                                   produit = produit * serie[i];
                                                                                                                      19. Pour compiler un programme prog.c, on utilise la
     \Box i = 0
                                                                                                                           ligne de commande :
                                                                 printf("produit = %d", produit);
     \Box i = 4
                                                                                                                            ☐ gcc prog.c -o -Wall prog.exe
                                                               La valeur affichée est :
12. Dans la commande gcc, l'option -Wall signifie :
                                                                 \Box 0
                                                                                                                            ☐ gcc prog.exe -Wall -o prog.c
     \square qu'on veut changer alétoirement de fond d'écran
                                                                 \square 16
                                                                                                                            ☐ gcc -Wall prog.exe -o prog.c
     □ qu'il faut indenter le fichier source
                                                                 \square 8
                                                                                                                            ☐ gcc -Wall prog.c -o prog.exe
     \square que l'on veut voir tous les avertissements
                                                                 \Box 4
                                                                                                                      20. Pour déclarer une fonction saisie_utilisateur qui
     □ qu'il faut lancer un déboggueur
                                                           16. On considère deux variables booléennes A et B initia-
                                                                                                                          demande à l'utilisateur d'entrer un entier au clavier et
                                                               lisées à TRUE et FALSE respectivement. Parmi les ex-
13. Le code suivant :
                                                                                                                          renvoie cet entier on écrit :
                                                               pressions booléennes suivantes, laquelle a pour valeur
     int i:
                                                               TRUE?
                                                                                                                            □ void saisie_utilisateur(char c);
    for (i = 1; i < 5; i = i + 1)
                                                                 \square !(!A || B) == (A && !B)
                                                                                                                            ☐ int saisie_utilisateur();
                                                                 ☐ (A == TRUE) && (B == TRUE)
         printf("%d ", i);
                                                                                                                            □ void saisie_utilisateur(int n);
                                                                 □ A && B
     }
                                                                                                                            □ saisie_utilisateur(scanf(%d));
                                                                 \square (!A || B)
     printf("\n");
```

### Éléments d'informatique – contrôle continue

 $\Box$  4

Prénom:	Nom:
N° INE ·	
11 11111 .	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Avant de faire appel à une fonction il est nécessaire de: □ l'avoir déclarée  $\square$  avoir défini une constante symbolique de la taille de cette fonction ☐ l'avoir définie □ l'avoir déclarée et définie 2. Le langage C est un langage  $\square$  interprété □ compilé □ lu, écrit, parlé  $\square$  composé 3. Le code suivant : int i; for (i = 4; i > 0; i = i - 1)printf("%d ", i); printf("\n"); affichera:  $\Box 01234$  $\square$  4 3 2 1  $\Box \ 4\ 3\ 2\ 1\ 0$  $\square$  0 1 2 3 4. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :

```
\square x = racine(x * x) - racine(x);
\square x = racine(2/3);
\square x - 1 = racine(x);
\square x = racine(racine(x)*racine(x));
```

5. Vous utilisez une boucle while quand:	
$\Box$ l'incrément de la variable de boucle n'est pas $1$	
□ vous avez déjà fait un <b>for</b> dans le même programme principal	
$\hfill \square$ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	
$\Box$ vous n'avez pas déclaré de fonction	
6. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :	
☐ int n = pgcd();	
☐ int pgcd(2);	
$\Box$ n = pgcd(int p, int q);	
$\square$ n = pgcd(n, 3);	
7. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :	
$\square$ void afficher_date(struct date_s d);	
$\square$ struct date_s afficher_date(struct date_s	d)
☐ int afficher_date(date_s d);	
☐ void afficher_date(date_s d);	
8. Soit la fonction f définie par :	
int f(int a)	
<pre>{   printf("a = \n", %d);   if (a &gt; 0)   {     return f(a - 1) + 1;   } }</pre>	
return 4; }	
Alors l'expression f(1) prendra la valeur :	
□ 5 □ 0	
$\Box$ 1	

enom o INE	: Nom :
9.	Un programme en langage C doit comporter une et une seule définition de la fonction :
	□ main
	□ include
	□ init
	□ begin
10.	Soit la fonction g définie par :
	<pre>int g(int a) {</pre>
	<pre>printf("a = \n", %d); if (1 &gt; 0) {    return 5; }</pre>
	return 7;
	}
	Alors l'expression $g(0)$ prendra la valeur :
	$\Box$ 5
i);	$\Box$ 7
	$\Box$ 0
11.	Si x est une variable réelle (de type double) alors $x = 3/2$ lui affecte la valeur :
	$\Box$ 0
	$\Box$ 1
	□ 1.5
	$\square$ 0.5
12.	Un fichier source est :
	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\square$ un document de référence du système
	$\square$ un document qui doit être protégé
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\square$ un document illisible pour les humains

```
13. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%x et y=%y\n");
     \square printf("x=%d et y=%d\n",x y);
     \square printf("x=%d et y=%d\n,x,y");
14. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
     printf("produit = %d", produit);
   La valeur affichée est :
     \Box 0
     \square 8
     \Box 16
     \Box 4
15. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
16. Le code suivant :
    int age = 18;
    if (age < 18)
    {
         printf("Mineur\n");
    }
```

```
else
     {
          printf("Majeur\n");
     }
    affichera:
      \square rien
      ☐ Mineur
      ☐ Mineur
         Majeur
      □ Majeur
17. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
    La valeur de somme affichée est :
      \square 15
      \Box 0
      \Box 10
      \Box 6
18. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
    Alors l'affichage sera :
      □ b
      \square B
      \Box C
      \square A
```

```
19. Soit le programme principal suivant :
   int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
     a = a + b;
     return a;
   L'affichage dans le main est le suivant :
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
20. Le code suivant :
    int i:
    for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 0 1 2 3 4 5 6 7
     \Box 02468
     \Box 0123456
```

 $\square$  0 2 4 6

## ${\bf\acute{E}} l\acute{e}ments\ d'in \underline{formatique-contr\^{o}le\ continue}$

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; $-0.5$ points par réponse fausse. Durée : $20$ minutes.	5. Un enregistrement permet de grouper plusieurs valeurs dans :	9. Vous utilisez une boucle whil ☐ l'incrément de la variable
<ol> <li>Dans la commande gcc, l'option -Wall signifie :         <ul> <li>qu'on veut changer alétoirement de fond d'écran</li> <li>qu'il faut lancer un déboggueur</li> <li>que l'on veut voir tous les avertissements</li> <li>qu'il faut indenter le fichier source</li> </ul> </li> <li>Après exécution jusqu'à la ligne 15 du programme C :     <ul> <li>int main() {</li> <li>int x = 5;</li> </ul> </li> </ol>	dans:  □ ses blocs □ ses chants □ ses champs □ ses cases  6. Quel est le problème d'un programme comportant les lignes suivantes?  while (1) {    printf("coucou\n"); } □ il n'affiche rien	□ l'incrément de la variable □ vous ne connaissez pas le la boucle à l'avance □ vous avez déjà fait un gramme principal □ vous n'avez pas déclaré e  10. Si racine est une fonction p et renvoyant la racine carrée une variable réelle définie et i d'écrire : □ x = racine(2/3); □ x - 1 = racine(x); □ x = racine(racine(x))
14	<ul> <li>□ il comporte une boucle infinie</li> <li>□ il ne compile pas</li> <li>□ il risque d'afficher bonjour à la place de coucou</li> </ul>	□ x = racine(x * x) - 1  11. Au début de la fonction main  char i;  for (i = 'A'; i <= 'F';
<ul> <li>□ la variable x vaut 16</li> <li>□ le programme affiche x</li> <li>□ le programme affiche ****</li> <li>□ la variable x vaut -½</li> <li>3. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu</li> </ul>	<ul> <li>7. Sur unix (ou linux), la commande mkdir permet de :  ouvrir un fichier texte  changer de répertoire courant  créer un fichier texte  créer un répertoire</li> <li>8. Pour l'extrait de programme suivant :  int i;</li> </ul>	{     printf("%c", i); } printf("\n"); Alors l'affichage sera: □ ABCDEF □ i
<pre>du struct, on écrit :</pre>	<pre>int j; for(i=4;i&gt;0;i=i-1) {     for(j=i;j&lt;6;j=j+1) }</pre>	□ cccccc □ A  12. Quel est l'opérateur de différe □ !=
□ void afficher_date(date_s d); □ struct date_s afficher_date(struct date_s  4. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la com-	<pre>printf("*"); d);    }     printf(" "); }</pre>	☐ <> ☐ ≠ ☐ !  13. Si cet avertissement apparaît warning: implicit declara
mande: □ mkdir TP4 □ kwrite TP4 □ new TP4	qu'est ce qui sera affiché?	, que doit-on chercher dans le  ☐ un désaccord entre la dé  d'une fonction  ☐ une fonction déclarée ma
□ yppasswd	_ **** *** *** *** _ ** ** ** ** **	☐ une fonction appelée ava ☐ une directive préprocesse

9. Vous utilisez une boucle while quand:
$\Box$ l'incrément de la variable de boucle n'est pas $1$
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
□ vous avez déjà fait un for dans le même programme principal
$\Box$ vous n'avez pas déclaré de fonction
10. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :
$\square$ x = racine(2/3);
$\square x - 1 = racine(x);$
$\square x = racine(racine(x) * racine(x));$
$\square x = racine(x * x) - racine(x);$
11. Au début de la fonction main() on place le code :
char i;
for (i = 'A'; i <= 'F'; i = i + 1)
{
<pre>printf("%c", i);</pre>
}
<pre>printf("\n");</pre>
Alors l'affichage sera :
□ ABCDEF □ i
□ A
$\hfill \square$ A 12. Quel est l'opérateur de différence en C :
□ A 12. Quel est l'opérateur de différence en C : □ !=
□ A   12. Quel est l'opérateur de différence en C :   □ !=   □ <>
□ A   12. Quel est l'opérateur de différence en C :   □ !=   □ <>   □ $\neq$
□ A  12. Quel est l'opérateur de différence en C : □ != □ <> □ $\neq$ □ !
□ A  12. Quel est l'opérateur de différence en C: □!= □ <> □ ≠ □!  13. Si cet avertissement apparaît à la compilation: warning: implicit declaration of function 'max', que doit-on chercher dans le programme? □ un désaccord entre la déclaration et la définition
□ A  12. Quel est l'opérateur de différence en C: □!= □ <> □ ≠ □!  13. Si cet avertissement apparaît à la compilation: warning: implicit declaration of function 'max', que doit-on chercher dans le programme? □ un désaccord entre la déclaration et la définition d'une fonction
□ A  12. Quel est l'opérateur de différence en C : □!= □ <> □ ≠ □!  13. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme? □ un désaccord entre la déclaration et la définition d'une fonction □ une fonction déclarée mais non définie
□ A  12. Quel est l'opérateur de différence en C: □!= □ <> □ ≠ □!  13. Si cet avertissement apparaît à la compilation: warning: implicit declaration of function 'max', que doit-on chercher dans le programme? □ un désaccord entre la déclaration et la définition d'une fonction

14. L'écriture $\underline{101}$ en binaire correspond au nombre natu-	□ j = 5	18. Une variable booléenne est un variable :
rel:	□ j = 0	$\square$ qui est vraie ou fausse
$\square$ 3		□ NaN (not a number, qui n'est pas un nombre)
$\Box$ 5		$\square$ à laquelle une valeur vient d'être affectée
$\Box$ 4	16. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et	☐ réelle positive
$\Box$ 101	renvoie cet entier on écrit :	$\Box$ jamais nulle
15. Soit un programme contenant les lignes suivantes :	□ void saisie_utilisateur(char c);	19. Pour déclarer un tableau d'entiers de taille 5, on peut
int i = 0;		utiliser l'instruction
int j = 0; for (i = 0; i < 0; i = i + 1)	□ void saisie_utilisateur(int n);	☐ int tab[] = 5;
{	$\square$ int saisie_utilisateur();	$\Box$ int toto[5];
for $(j = 0; j < 5; j = j + 1)$	$\square$ saisie_utilisateur(scanf(%d));	☐ int toto[taille=5];
{	17. Si cette erreur apparaît à la compilation :	☐ char tableau[5];
 }	erreur: conflicting types for 'max', que doit-	$\Box$ int[] new tableau(5);
}	on chercher dans le programme?	20. Laquelle des analyses suivantes ne fait pas partie des
$printf("j = %d\n", j);$	☐ une directive préprocesseur #include manquante	étapes de la compilation :
 }	☐ un désaccord entre la déclaration et la définition	$\square$ analyse sémantique
ı	d'une fonction	$\square$ analyse syntaxique
qu'est ce qui sera affiché?	$\square$ une fonction déclarée mais non définie	$\square$ analyse lexicale
☐ j = %d	$\square$ une fonction appelée avant sa déclaration	$\square$ analyse harmonique
_ 3	'	'

### Éléments d'informatique – contrôle continue

4. Le code suivant :

printf("\n");

for (i = 0; i < 5; i = i + 1)

printf("%d ", i);

int i;

{

Prénom:	Nom:
N° INE ·	
IN IINE.	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max' , que doit-on chercher dans le programme? ☐ une directive préprocesseur #include manquante □ un désaccord entre la déclaration et la définition d'une fonction □ une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration 2. Soit le programme principal suivant : int main() int a = 3; int b = 5; printf("f(a,b)=%d, a=%d,  $b=%d\n",f(a,b),a,b$ ); return EXIT\_SUCCESS; appelant la fonction f ainsi définie : int f(int a, int b) a = a + b; return a; L'affichage dans le main est le suivant :  $\Box$  f(a,b)=8, a=3, b=5  $\Box$  f(a,b)=13, a=8, b=5  $\Box$  f(a,b)=8, a=8, b=5  $\Box$  f(3,5)=8, a=3, b=5 3. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :  $\square$  n = factorielle(p, q);

□ printf("%d", factorielle(n));

 $\square$  n = factorielle();

☐ int factorielle(int 2);

affichera:  $\Box$  4 3 2 1 0  $\Box 01234$  $\square$  0 1 2 3  $\square$  4 3 2 1 5. Quels calculs peut-on programmer en programmation structurée? ☐ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine □ il y a des calculs programmables en langage machine et qui ne sont pas programmables en programmation structurée □ certains programmes sont de vrais plats de spaghetti □ en programmation structurée on peut programmer tous les calculs programmables en langage machine 6. Si le code : struct toto\_s int n; double x; }; précède la fonction main(), alors on peut écrire en début de main() :  $\square$  int struct toto\_s = {3, -1e10};  $\square$  toto\_s n, x;  $\square$  int toto.n = 3;  $\square$  toto\_s struct z = {3, 0.5}; □ struct toto s toto:

```
7. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses blocs
      \square ses champs
      □ ses cases
      \square ses chants
 8. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
      \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
 9. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
      ☐ #define taille = 3
      \square #define taille = N
      □ #define N 3
      \square #define N = 3
10. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
      ☐ une fonction appelée avant sa déclaration
      □ une directive préprocesseur #include manquante
      □ une fonction déclarée mais non définie
      □ un désaccord entre la déclaration et la définition
         d'une fonction
11. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      \square int[] new tableau(5);
      ☐ char tableau[5]:
      \square int toto[5]:
      \square int tab[] = 5;
      ☐ int toto[taille=5];
```

12. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est	15. Sur unix (ou linux), la commande mkdir permet de :	18. Vous avez déclaré préalablemen tions utilisées par votre prograr
une variable réelle définie et initialisée, il est incorrect	$\square$ ouvrir un fichier texte	dans lequel vous devez maintena
d'écrire :	$\Box$ créer un fichier texte	est l'ordre :
$\square$ x = racine(x * x) - racine(x);	$\Box$ créer un répertoire	$\Box$ dans lequel ces fonctions
$\square$ x - 1 = racine(x);	□ changer de répertoire courant	main
$\square$ x = racine(2/3);	•	$\square$ un ordre quelconque
<pre>     x = racine(racine(x)*racine(x)); </pre>	16. Pour déclarer une fonction saisie_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et	☐ dans lequel vous avez décla
13. Un fichier source est :	renvoie cet entier on écrit :	$\square$ alphabétique
$\square$ un fichier que l'ont doit citer dans les documents	☐ int saisie_utilisateur();	19. Le bus système sert à :
produits sur l'ordinateur		☐ Transférer des données et
$\Box$ un document illisible pour les humains	□ void saisie_utilisateur(int n);	cesseur et mémoire
$\square$ un document de référence du système	☐ void saisie_utilisateur(char c);	☐ Écrire des données sur le d
☐ un fichier texte qui sera traduit en instructions processeur	☐ saisie_utilisateur(scanf(%d));	□ transporter les processus of cesseur
□ un document qui doit être protégé	17. Soit la fonction g définie par :	☐ Arriver à l'heure en cours
14. Après exécution jusqu'à la ligne 15 du programme C :	int g(int a)	20. Après exécution jusqu'à la ligne
10 int main() {	{ printf("a = \n", %d);	10
11 int x = 5;	if  (1 > 0)	11 int main() {
12 int y = 3;	{	$12 \qquad \text{int } x = 5;$
13	return 5;	13
14 x = y;	}	14 x = 3 * x + 1; 15
15	return 7;	1.0
16 17 }	}	17 }
	Alors l'expression g(0) prendra la valeur :	
$\square$ la variable x vaut 3		□ la variable x vaut 16
$\square$ la variable x vaut 5 et la variable y vaut 3	$\Box$ 5	$\Box$ la variable x vaut $-\frac{1}{2}$
$\square$ la variable y vaut 5	$\Box$ 0	☐ le programme affiche ****
$\Box$ le programme affiche "Faux"	□ 7	$\Box$ le programme affiche x

18. Vous avez déclaré préalablement un ensemble de fonc-
tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre:
$\Box$ dans lequel ces fonctions sont appelées dans le main
$\hfill\Box$ un ordre quel conque
$\Box$ dans lequel vous avez déclaré ces fonction
□ alphabétique
19. Le bus système sert à :
☐ Transférer des données et intructions entre processeur et mémoire
$\Box$ Écrire des données sur le dique dur
$\Box$ transporter les processus du tourniquet au processeur
☐ Arriver à l'heure en cours
20. Après exécution jusqu'à la ligne 15 du programme C :
10
<pre>11 int main() {</pre>
12   int x = 5;
13
14
16
17 }
□ la variable x vaut 16
$\Box$ la variable x vaut $-\frac{1}{2}$

т		-1
	100000	- 1
	acence	

### Éléments d'informatique – contrôle continue

Prénom:	Nom:
NIO INII	1.0111
N° INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 2; i = i + 1)
   for (j = 0; j < 3; j = j + 1)
        printf("%d ", i);
}
```

qu'est ce qui sera affiché?

```
\Box 0 0 0 1 1 1
```

- $\Box$  0 1 0 1 0 1 0 1
- $\Box$  0 1 2 0 1 2
- $\Box$  1 2 1 2 3
- 2. Si a et b sont deux variables de type :

```
struct toto_s
{
  int n;
  double x;
};
```

Alors pour tester l'égalité de a et de b on utilise la condition:

```
\square a == b
```

- $\square$  (a.n == b.n) && (a.x == b.x)
- $\Box$  a = b
- $\square$  a{n, x} == b{n, x}
- 3. Une variable booléenne est un variable :
  - $\square$  jamais nulle
  - ☐ réelle positive
  - $\square$  qui est vraie ou fausse
  - □ NaN (not a number, qui n'est pas un nombre)
  - $\Box$  à la quelle une valeur vient d'être affectée

```
4. L'écriture 101 en binaire correspond au nombre natu-
  rel:
     \square 3
```

- $\Box$  5
- $\Box$  4  $\square$  101
- 5. Soit le programme principal suivant :

```
int main()
int a = 3:
int b = 5:
printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
return EXIT_SUCCESS;
```

appelant la fonction f ainsi définie :

```
int f(int a, int b)
  a = a + b;
  return a;
```

L'affichage dans le main est le suivant :

- $\Box$  f(a,b)=8, a=3, b=5
- $\Box$  f(3,5)=8, a=3, b=5
- $\Box$  f(a,b)=8, a=8, b=5
- $\Box$  f(a,b)=13, a=8, b=5
- 6. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
  - □ analyse sémantique
  - $\square$  analyse harmonique
  - $\square$  analyse syntaxique
  - $\square$  analyse lexicale
- 7. Les lignes

```
int i;
int x=0;
for(i=0,i<5,i=i+1)
  x=x+1;
```

- □ comportent une erreur qui ne sera pas détectée
- □ comportent une erreur qui sera détectée au cours de l'édition de lien
- $\square$  ne comportent aucune erreur
- □ comportent une erreur qui sera détectée au cours de l'analyse syntaxique
- 8. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
\square n = factorielle():
```

- ☐ printf("%d", factorielle(n));
- ☐ int factorielle(int 2):
- $\square$  n = factorielle(p, q);
- 9. Le bus système sert à :
  - ☐ Écrire des données sur le dique dur
  - ☐ Arriver à l'heure en cours
  - □ transporter les processus du tourniquet au processeur
  - ☐ Transférer des données et intructions entre processeur et mémoire
- 10. Soit la fonction f définie par :

```
int f(int a)
  printf("a = \n", %d);
  if (a > 0)
  {
    return f(a - 1) + 1;
  return 4;
```

Alors l'expression f(1) prendra la valeur :

- $\Box$  4
- $\Box$  0
- $\Box$  1
- $\Box$  5

```
11. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 0
      \Box 4
12. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      \Box init
      □ begin
      \square include
      □ main
13. Le langage C est un langage
      □ interprété
      □ lu, écrit, parlé
      □ compilé
      □ composé
14. Le type des réels en C est :
      \square real
      \square int
      ☐ double
      □ char
```

```
15. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      ☐ int factorielle();
      ☐ int factorielle(double n);
      □ struct int factorielle(int n);
      \square int factorielle(int x);
16. Une de ces manière de composer les blocs de pro-
    grammes ne fait pas partie des opérations de la pro-
    grammation structurée :
     \Boxmettre les blocs en séquence les uns à la suite des
        autres
      □ répéter un bloc tant qu'une condition est vérifée
      □ sélectionner entre deux blocs à l'aide d'une condi-
        tion
      \square retourner un bloc
17. Lorsqu'un programme utilise printf ou scanf il faut
    qu'il contienne l'instruction préprocesseur :
      ☐ #include <studlib.h>
      ☐ #include <stdio.h>
      ☐ #include <studio.h>
      ☐ #appart <stdlib.h>
18. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
```

```
printf("j = %d\n", j);
   qu'est ce qui sera affiché par ce printf?
     \Box i = 0
     \Box j = 5
      \Box j = %d
     \Box j = 4
19. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 4
      \square 16
      \Box 0
      \square 8
20. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
      \square n = carre(n);
```

 $\square$  n = carre(int n);

 $\square$  int n = carre();

 $\square$  int carre(2);

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le langage C est un langage
     □ composé
     □ lu, écrit, parlé
     □ interprété
     □ compilé
2. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
     ☐ #include <studlib.h>
     ☐ #include <studio.h>
    ☐ #appart <stdlib.h>
     ☐ #include <stdio.h>
3. Soit la fonction f définie par :
  int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
  Alors l'expression f(1) prendra la valeur :
     \Box 5
     \Box 4
    \Box 1
     \Box 0
4. Soit le programme principal suivant :
  int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
```

```
int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=13, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
5. Pour l'extrait de programme suivant :
    int i;
    int j;
    for(i=4;i>0;i=i-1)
      for(j=i;j<6;j=j+1)
        printf("*");
      printf(" ");
   qu'est ce qui sera affiché?
         ** *** **** *****
         ***** *** ***
6. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
    ☐ int factorielle(double n);
    □ struct int factorielle(int n);
    ☐ int factorielle();
     \square int factorielle(int x);
7. Sous unix (ou linux), la commande 1s permet de :
     \Box afficher le contenu d'un fichier texte
    □ voir des clips musicaux
    □ afficher la liste de fichiers contenus dans un
       répertoire
     \square compiler un programme
```

```
8. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
     ☐ char 'c';
     ☐ int char:
     ☐ char "c";
     ☐ char c:
9. Le code suivant :
     int somme = 0;
     int i;
    for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
    printf("%d", somme);
    affichera:
     \Box 1
     \Box 6
     \Box 42
     \Box 0
10. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit :
     ☐ struct date_s afficher_date(struct date_s d);
     □ void afficher_date(date_s d);
     □ void afficher date(struct date s d):
     ☐ int afficher_date(date_s d);
11. L'ordonnancement par tourniquet permet :
     ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
     \square de doubler la mémoire disponible
     □ d'afficher des ronds colorés à l'écran
     \square de ne pas perdre de temps avec la commutation
        de contexte
```

12. Vous utilisez une boucle while quand :	$\Box$ toto_s struct z = {3, 0.5};	□ 1 2 1 2 3
$\Box$ vous avez déjà fait un for dans le même pro-	□ toto_s n, x;	$\Box$ 0 0 0 1 1 1
gramme principal	☐ int struct toto_s = {3, -1e10};	$\Box$ 0 1 0 1 0 1 0 1
$\Box$ vous n'avez pas déclaré de fonction	15. Un enregistrement permet de grouper plusieurs valeurs	□ 0 1 2 0 1 2
$\hfill \square$ l'incrément de la variable de boucle n'est pas $1$	dans:	18. Si cette erreur apparaît à la compilation :
$\Box$ vous ne connaissez pas le nombre d'itérations de	$\square$ ses chants	Undefined symbols :"_prinft" ou
la boucle à l'avance	☐ ses cases	référence indéfinie vers « prinft » que doit
13. Quel est le problème d'un programme comportant les	□ ses blocs	on chercher dans le programme?
lignes suivantes?	$\square$ ses champs	□ une variable non déclarée
while (1)	16. Sur un ordinateur avec un seul processeur, habituelle-	☐ une faute de frappe dans un appel de fonction
<pre>{     printf("coucou\n");</pre>	ment les processus sont exécutés :	$\Box$ un caractère interdit en C
}	$\square$ tour à tour, un petit peu à chaque fois	$\square$ une directive préprocesseur <b>#include</b> manquant
$\Box$ il comporte une boucle infinie	☐ chacun son tour, après que le processus précédent a terminé	19. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la pro
$\square$ il n'affiche rien	$\Box$ tous ensemble	grammation structurée :
$\Box$ il ne compile pas	□ en parallèle, chacun dans un registre	$\Box$ retourner un bloc
$\Box$ il risque d'afficher bonjour à la place de coucou	17. Soit un programme contenant les lignes suivantes :	$\square$ sélectionner entre deux blocs à l'aide d'une condition
14. Si le code :	int i = 0;	$\square$ mettre les blocs en séquence les uns à la suite de
struct toto_s	int j = 0; for (i = 0; i < 2; i = i + 1)	autres
{    int n;	{	$\hfill\Box$ répéter un bloc tant qu'une condition est vérifée
double x;	for (j = 0; j < 3; j = j + 1)	20. L'écriture <u>101</u> en binaire correspond au nombre natu
};	{	rel:
précède la fonction main(), alors on peut écrire en	printf("%d ", i);	$\square$ 101
début de main() :	}	$\square$ 3
$\Box$ int toto.n = 3;		$\Box$ 4
☐ struct toto_s toto;	qu'est ce qui sera affiché?	$\Box$ 5

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un fichier source est: un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un fichier texte qui sera traduit en instructions □ un document illisible pour les humains □ un document qui doit être protégé  $\square$  un document de référence du système 2. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?  $\square$  une accolade manquante  $\square$  une accolade en trop ☐ un point-virgule manquant □ un point-virgule en trop 3. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect d'écrire :  $\square$  x = racine(x \* x) - racine(x);  $\square$  x = racine(racine(x)\*racine(x));  $\square$  x = racine(2/3);  $\square$  x - 1 = racine(x); 4. Un enregistrement permet de grouper plusieurs valeurs dans:  $\square$  ses chants □ ses cases  $\square$  ses champs  $\square$  ses blocs 5. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :  $\square$  printf("x=%d et y=%d\n",x,y);  $\square$  printf("x=%x et y=%y\n");

 $\square$  printf("x=%d et y=%d\n,x,y");

 $\square$  printf("x=%d et y=%d\n",x y);

```
6. Pour déclarer une fonction pgcd qui calcule et renvoie
    le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
     \Box int pgcd(int x, y);
     \square void pgcd(int x, int y);
     \Box int pgcd(int x, int x);
     \square int pgcd(int y, int x);
 7. Si factorielle est une fonction prenant en entrée un
    entier et renvoyant un entier, il est correct d'écrire :
     ☐ int factorielle(int 2):
     ☐ printf("%d", factorielle(n));
     \square n = factorielle();
     \square n = factorielle(p, q);
 8. Le type des réels en C est :
     □ double
     \square real
      □ int.
     □ char
 9. Pour déclarer une fonction saisie_utilisateur qui
    demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit:
     ☐ int saisie utilisateur():
     □ void saisie_utilisateur(char c);
     □ saisie_utilisateur(scanf(%d));
      □ void saisie_utilisateur(int n);
10. La virtualisation de la mémoire permet notamment de
   stocker des portions inactives de la mémoire de travail
   sur le disque dur. Mais on perd :
     □ les fichiers du disque
     □ certaines données de la mémoire de travail
     □ en temps d'accès
      \square des processus
11. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu();
     □ double afficher_menu();
     □ void afficher_menu();
      ☐ int afficher_menu(int char);
```

```
12. Si n est une variable entière pour demander sa valeur
    à l'utilisateur, on utilise plutôt :
      □ printf("Valeur de n ? %d\n", n);
      □ scanf("%d", &n);
     ☐ printf("Valeur de n ? %g\n", n);
      □ un débogueur
13. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
      \Box 4
      \square 8
      \Box 0
      \Box 16
14. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \square 1.5
      \Box 0
      \Box 1
      \square 0.5
15. Soit la fonction f définie par :
    int f(int a)
    {
      printf("a = \n", %d);
      if (a > 0)
      {
        return 3;
      return 4:
   }
    Alors l'expression f(0) prendra la valeur :
      \Box 0
      \square 3
```

 $\Box$  4

16.	Le bus système sert à :
	$\Box$ Écrire des données sur le dique dur
	$\Box$ Arriver à l'heure en cours
	$\Box$ transporter les processus du tourniquet au processeur
	$\Box$ Transférer des données et intructions entre processeur et mémoire
17.	Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
	$\square$ n = pgcd(int p, int q);
	$\square$ n = pgcd(n, 3);
	☐ int n = pgcd();
	☐ int pgcd(2);

lisées à TRUE et FALSE respectivement. Parmi les ex-

```
pressions booléennes suivantes, laquelle a pour valeur TRUE?

☐ (!A || B)
☐ A && B
☐ !(!A || B) == (A && !B)
☐ (A == TRUE) && (B == TRUE)

20. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci :
Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »
☐ l'analyse sémantique
☐ l'analyse harmonique
☐ l'analyse des entrées clavier
```

□ l'édition de liens

# Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Soit la fonction g définie par :

```
int g(int a)
  {
     printf("a = \n", %d);
     if (1 > 0)
       return 5:
     return 7;
  Alors l'expression g(0) prendra la valeur :
    \Box 7
    \Box 5
    \Box 0
2. Pour l'extrait de programme suivant :
     int produit = 1;
     int serie[4] = \{2, 2, 2, 2\};
     for (i = 0; i < 4; i = i + 1)
       produit = produit * serie[i];
     printf("produit = %d", produit);
  La valeur affichée est :
    \Box 16
    \Box 0
    \Box 4
```

3. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

```
\Box char c;
□ char 'c';
☐ int char:
☐ char "c":
```

 $\square$  8

4. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt : ☐ printf("Valeur de n ? %g\n", n); □ printf("Valeur de n ? %d\n", n); □ un débogueur □ scanf("%d", &n); 5. Après exécution jusqu'à la ligne 15 du programme C: int main() { 10 11 int x = 5; 12 int y = 3; 13 14 x = y;15 16 17 } ☐ le programme affiche "Faux"  $\square$  la variable x vaut 5 et la variable y vaut 3 □ la variable y vaut 5  $\square$  la variable x vaut 3 6. Un registre du processeur est : □ une unité de calcul spécialisée de l'ordinateur □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs  $\square$  une gamme de fréquence de fonctionnement du processeur □ un composant qui contient la liste des fichiers du système 7. Soit la fonction f définie par : int f(int a) {  $printf("a = \n", %d);$ if (a > 0)return 3; return 4;

Alors l'expression f(0) prendra la valeur :

 $\square$  3

 $\square$  0

 $\Box$  4

```
8. Si cette erreur apparaît à la compilation :
    erreur: conflicting types for 'max', que doit-
   on chercher dans le programme?
     □ une fonction déclarée mais non définie
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une directive préprocesseur #include manquante
     \square une fonction appelée avant sa déclaration
9. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit:
     □ void saisie_utilisateur(int n);
     ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
     ☐ int saisie_utilisateur();
10. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
     □ qu'on veut changer alétoirement de fond d'écran
     ☐ que l'on veut voir tous les avertissements
11. Le code suivant :
     int somme = 0;
     int i;
    for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
    printf("%d", somme);
   affichera:
     \Box 0
     \square 42
     \Box 1
     \Box 6
```

12. Au début de la fonction main() on place le code :	$\square$ l'analyse des entrées clavier	18. Un bit est:
char i;	☐ l'édition de liens	$\hfill\Box$ la longueur d'un mot mémoire
for (i = 'A'; i <= 'F'; i = i + 1)	$\Box$ l'analyse sémantique	$\Box$ l'instruction qui met fin à un programme
{	15. Sur un ordinateur avec un seul processeur, habituelle-	$\square$ un chiffre binaire (0 ou 1)
<pre>printf("%c", i); }</pre>	ment les processus sont exécutés :	□ un battement d'horloge processeur
<pre>printf("\n");</pre>	$\hfill \square$ chacun son tour, après que le processus précédent	19. Quel est le problème d'un programme comportant les
Alors l'affichage sera :	a terminé	lignes suivantes?
☐ ABCDEF	$\Box$ en parallèle, chacun dans un registre	while (1)
□ A	$\Box$ tour à tour, un petit peu à chaque fois	{
□i	$\Box$ tous ensemble	<pre>printf("coucou\n");</pre>
□ сссссс	16. Sur unix (ou linux), la commande mkdir permet de :	}
13. Pour déclarer une fonction pgcd qui calcule et renvoie	□ créer un répertoire	$\square$ il comporte une boucle infinie
le plus grand diviseur commun de deux entiers positifs	□ créer un fichier texte	$\Box$ il risque d'afficher bonjour à la place de coucou
passés en arguments on écrit :	□ ouvrir un fichier texte	$\square$ il n'affiche rien
$\square$ int pgcd(int y, int x);	□ changer de répertoire courant	$\square$ il ne compile pas
$\Box$ int pgcd(int x, int x);	17. Pour déclarer une fonction exposant qui prend en ar-	20. Pour afficher à l'aide de printf("%d\n",tab[i]);
$\Box$ int pgcd(int x, y);	gument un réel $x$ et un entier positif $n$ et renvoie la	le contenu d'un tableau de 5 entiers initialisé au
$\square$ void pgcd(int x, int y);	valeur de $x^n$ on écrit :	préalable, on utilise plutôt :
14. Quelle étape de la compilation vient d'échouer lors-	$\square$ exposant(double x, int n, int r);	☐ for(i=0;i<=5;i=i+1)
qu'on a un message comme celui-ci :	$\Box$ double exposant(double x, int n);	☐ for(i=1;i<=5;i=i+1)
Undefined symbols :"_prinft" ou référence indéfinie vers « prinft »	$\Box$ int exposant(double n, int x);	☐ for(i=0;i<5;i=i+1)
☐ l'analyse harmonique	$\square$ void exposant(double x^n);	☐ for(i=1;i<5;i=i+1)

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour déclarer une procédure afficher_date qui prend
  en argument un struct date_s et affiche le contenu
  du struct, on écrit:
    □ struct date_s afficher_date(struct date_s d);
    ☐ int afficher_date(date_s d);
    □ void afficher_date(date_s d);
    □ void afficher_date(struct date_s d);
2. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Mineur
    □ rien
    □ Majeur
    ☐ Mineur
       Majeur
3. Soit le programme principal suivant :
  int main()
   int a = 3;
    int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
    a = a + b;
    return a:
  L'affichage dans le main est le suivant :
    \Box f(3,5)=8, a=3, b=5
```

```
\Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=13, a=8, b=5
4. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
    \square un désaccord entre la déclaration et la définition
        d'une fonction
    ☐ une fonction appelée avant sa déclaration
    \square une fonction déclarée mais non définie
    □ une directive préprocesseur #include manquante
5. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
  pressions booléennes suivantes, laquelle a pour valeur
  TRUE?
    \square (!A | | B)
    \square !(!A \mid | B) == (A \&\& !B)
    □ A && B
    ☐ (A == TRUE) && (B == TRUE)
6. Si a et b sont deux variables de type:
   struct toto_s
     int n;
     double x;
  };
  Alors pour tester l'égalité de a et de b on utilise la
  condition:
    \Box a = b
    \square a == b
    \Box (a.n == b.n) && (a.x == b.x)
    \square a{n, x} == b{n, x}
7. Soient deux variables entières x et y initialisées à 4 et
  5 respectivement. L'affichage x=4 et y=5 est obtenu
  avec la commande :
    \square printf("x=%d et y=%d\n",x,y);
    \square printf("x=%d et y=%d\n",x y);
    \Box printf("x=%x et y=%y\n");
```

 $\square$  printf("x=%d et y=%d\n,x,y");

```
8. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
    on chercher dans le programme?
      \square une accolade en trop
      \square une accolade manquante
      □ un point-virgule en trop
      ☐ un point-virgule manquant
 9. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square int exposant(double n, int x);
      \square exposant(double x, int n, int r);
      \square double exposant(double x, int n);
      \square void exposant(double x^n);
10. Si carre est une fonction prenant en entrée un en-
    tier et renvoyant le carré de cet entier, et que n est
    une variable entière définie et initialisée, il est correct
    d'écrire :
      \square n = carre(n):
      \square int n = carre():
      \square int carre(2);
      \square n = carre(int n);
11. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 5
      \square 101
      \Box 4
      \square 3
12. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      \Box int tab[] = 5;
      \square int[] new tableau(5);
      \square int toto[5]:
      ☐ int toto[taille=5];
```

□ char tableau[5];

3. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?		{     for (j = 0; j < 5; j = j + 1)     f
□ une directive préprocesseur #include manquante	□ certains programmes sont de vrais plats de spa- ghetti	
☐ une fonction déclarée mais non définie	$\square$ il y a des calculs programmables en programma-	} }
$\Box$ une fonction appelée avant sa déclaration	tion structurée qui ne sont pas programmables en langage machine	<pre>printf("j = %d\n", j);</pre>
$\hfill\Box$ un désaccord entre la déclaration et la définition d'une fonction	☐ il y a des calculs programmables en langage ma- chine et qui ne sont pas programmables en pro-	qu'est ce qui sera affiché par ce printf?
4. Le code suivant :	grammation structurée	□ j = 4 -
int i;	□ en programmation structurée on peut program-	□ j = 5
for (i = 0; i < 7; i = i + 2)	mer tous les calculs programmables en langage machine	□ j = %d
{		□ j = 0
printf("%d ", i);	17. Soit la fonction f définie par :	19. Un fichier source est:
<pre>printf("\n");</pre>	int f(int a)	$\Box$ un document de référence du système
affichera:	<pre>printf("a = \n", %d); if (a &gt; 0)</pre>	☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
$\square \ 0\ 1\ 2\ 3\ 4\ 5\ 6$	11 (a > 0) {	$\square$ un document illisible pour les humains
$\square \ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7$	return 3;	$\square$ un fichier texte qui sera traduit en instructions
$\Box \ 0\ 2\ 4\ 6\ 8$	}	processeur
$\Box \ 0 \ 2 \ 4 \ 6$	return 4;	$\Box$ un document qui doit être protégé
5. Un registre du processeur est :	Alors l'expression f(0) prendra la valeur :	20. Vous avez déclaré préalablement un ensemble de fonc- tions utilisées par votre programme principal. L'ordre
☐ un composant qui contient la liste des fichiers du système	□ 3 □ 4	dans lequel vous devez maintenant définir ces fonctions est l'ordre :
□ une unité de calcul spécialisée de l'ordinateur		$\square$ alphabétique
☐ une gamme de fréquence de fonctionnement du processeur	18. Soit un programme contenant les lignes suivantes :	$\Box$ dans le quel ces fonctions sont appelées dans le main
□ une case mémoire interne au processeur qui sera	int i = 0;	☐ dans lequel vous avez déclaré ces fonction
manipulée directement lors des calculs	int j = 0; for (i = 0; i < 3; i = i + 1)	$\Box$ un ordre quelconque

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit le programme principal suivant :
  int main()
    int a = 3;
    int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
2. Pour déclarer une fonction exposant qui prend en ar-
  gument un réel x et un entier positif n et renvoie la
  valeur de x^n on écrit :
    \square double exposant(double x, int n);
    \square void exposant(double x^n);
    \square int exposant(double n, int x);
    \square exposant(double x, int n, int r);
3. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
   for (i = 0; i < 3; i = i + 1)
        for (j = 0; j < 5; j = j + 1)
        }
   printf("j = %d\n", j);
```

```
qu'est ce qui sera affiché par ce printf?
    \Box j = %d
    \Box j = 5
    \Box i = 0
    \Box j = 4
4. Pour déclarer une fonction saisie_utilisateur qui
  demande à l'utilisateur d'entrer un entier au clavier et
  renvoie cet entier on écrit:
    □ void saisie_utilisateur(char c);
    ☐ int saisie_utilisateur();
    ☐ saisie_utilisateur(scanf(%d));
    □ void saisie_utilisateur(int n);
5. Si carre est une fonction prenant en entrée un en-
  tier et renvoyant le carré de cet entier, et que n est
  une variable entière définie et initialisée, il est correct
  d'écrire :
    \square n = carre(n);
    \square n = carre(int n);
    \square int n = carre():
    \square int carre(2);
6. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
    □ void afficher_menu();
    ☐ int afficher_menu();
    ☐ int afficher_menu(int char);
    □ double afficher_menu();
    ☐ char afficher_menu(printf("menu"));
7. Pour déclarer une procédure afficher_date qui prend
  en argument un struct date_s et affiche le contenu
  du struct, on écrit:
    □ void afficher_date(date_s d);
    ☐ int afficher_date(date_s d);
    □ struct date_s afficher_date(struct date_s d);
    □ void afficher_date(struct date_s d);
```

```
8. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \Box 5
      \Box 1
      \square 0
      \Box 4
 9. Un registre du processeur est :
     □ une unité de calcul spécialisée de l'ordinateur
     □ une gamme de fréquence de fonctionnement du
         processeur
      □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
      \square un composant qui contient la liste des fichiers du
         système
10. Sous unix (ou linux), pour créer un répertoire TP4
    dans le répertoire courant on peut utiliser la com-
   mande:
     □ mkdir TP4
     □ new TP4
     ☐ kwrite TP4
     □ yppasswd
11. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse harmonique
      \square analyse lexicale
      □ analyse sémantique
      \square analyse syntaxique
```

12.	Un bit est:
	$\Box$ l'instruction qui met fin à un programme
	$\Box$ la longueur d'un mot mémoire
	$\Box$ un battement d'horloge processeur
	$\square$ un chiffre binaire (0 ou 1)
13.	Le code suivant :
	<pre>int somme = 0; int i; for (i = 1; i &lt; 4; i = i + 1) {</pre>
	<pre>somme = somme + i; }</pre>
	<pre>printf("%d", somme);</pre>
	affichera :
	$\Box$ 42
	□ 1
	$\Box$ 0
	$\Box$ 6
14.	Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
	☐ for(i=1;i<=5;i=i+1)
	☐ for(i=0;i<=5;i=i+1)
	☐ for(i=0;i<5;i=i+1)
	☐ for(i=1;i<5;i=i+1)

```
15. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \Box 0
     \square 3
     \Box 4
16. On considère deux variables booléennes A et B initia-
    lisées à TRUE et FALSE respectivement. Parmi les ex-
    pressions booléennes suivantes, laquelle a pour valeur
    TRUE?
     ☐ (A == TRUE) && (B == TRUE)
     \square !(!A || B) == (A && !B)
     □ A && B
     \square (!A || B)
17. Le bus système sert à :
     □ transporter les processus du tourniquet au pro-
         cesseur
      ☐ Arriver à l'heure en cours
     □ Transférer des données et intructions entre pro-
         cesseur et mémoire
     ☐ Écrire des données sur le dique dur
```

18.	Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit- on chercher dans le programme?  □ une variable non déclarée
	$\Box$ une faute de frappe dans un appel de fonction
	$\Box$ un caractère interdit en C
	$\Box$ une directive préprocesseur $\verb"#include"$ manquante
19.	Si a et b sont deux variables de type :
	<pre>struct toto_s {   int n;</pre>
	double x;
	<b>}</b> ;
	Alors pour tester l'égalité de a et de b on utilise la condition :
	$\square (a.n == b.n) & (a.x == b.x)$
	□ a == b
	□ a = b
	$\square a\{n, x\} == b\{n, x\}$
20.	Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :
	$\Box$ tous ensemble
	□ chacun son tour, après que le processus précédent a terminé
	$\square$ tour à tour, un petit peu à chaque fois

□ en parallèle, chacun dans un registre

т		-1
	100000	- 1
	acence	

## Éléments d'informatique – contrôle continue

 $\square$  0

Prénom:	Nom ·
I ICHOIII .	Nom.
N° INE ·	
IN IINE .	

par réponse fausse. Durée : 20 minutes.

- Barème: 1 points par réponse juste (unique); -0.5 points 1. Un registre du processeur est : □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs □ une unité de calcul spécialisée de l'ordinateur  $\square$  un composant qui contient la liste des fichiers du □ une gamme de fréquence de fonctionnement du processeur 2. Le langage C est un langage □ lu, écrit, parlé □ composé □ interprété □ compilé 3. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit : ☐ int saisie\_utilisateur(); □ void saisie\_utilisateur(char c); □ void saisie utilisateur(int n): □ saisie\_utilisateur(scanf(%d)); 4. Pour l'extrait de programme suivant : int i = 0;
  - int j = 0; for (i = 0; i < 3; i = i + 1)for (j = 0; j < 2; j = j + 1)printf("%d ", i); printf("\n"); qu'est ce qui sera affiché?  $\Box$  0 0 1 1 2 2  $\Box$  0 1 0 1 0 1  $\Box$  1 2 3 1 2  $\Box$  0 1 2 0 1 2

```
5. Si x est une variable réelle (de type double) alors
  x = 3/2 lui affecte la valeur :
     \Box 1
     \square 1.5
     \square 0.5
     \Box 0
6. La virtualisation de la mémoire permet notamment de
  stocker des portions inactives de la mémoire de travail
  sur le disque dur. Mais on perd :
     ☐ des processus
     □ certaines données de la mémoire de travail
     □ les fichiers du disque
     \square en temps d'accès
7. Pour déclarer une fonction pgcd qui calcule et renvoie
   le plus grand diviseur commun de deux entiers positifs
  passés en arguments on écrit :
    \Box int pgcd(int x, int x);
     \square int pgcd(int y, int x);
    \square void pgcd(int x, int y);
    \Box int pgcd(int x, y);
8. Soit la fonction f définie par :
  int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \Box 4
     \square 3
```

```
9. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ char afficher_menu(printf("menu"));
     ☐ double afficher_menu();
     ☐ int afficher_menu(int char);
     ☐ int afficher_menu();
     □ void afficher_menu();
10. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 5
     \square 3
     \square 101
     \Box 4
11. Pour l'extrait de programme suivant :
      int somme = 0;
      for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
        i = i + 1; /* attention ! */
      printf("somme = %d",somme);
   La valeur de somme affichée est :
     \square 15
     \Box 6
     \Box 10
     \Box 0
12. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1:
      return 4;
   Alors l'expression f(1) prendra la valeur :
```

 $\Box$  5

```
\Box 1
     \Box 0
     \Box 4
13. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
    printf("\n");
    affichera:
     \square 8 2
     \Box 8 6 4 2 0
     \Box 02468
     \square 8 6 4 2
14. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", i);
         }
     }
```

qu'est ce qui sera affiché?	18. Si cette erreur apparaît à la compilation :  Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit- on chercher dans le programme ?  ☐ un caractère interdit en C ☐ une variable non déclarée ☐ une faute de frappe dans un appel de fonction ☐ une directive préprocesseur #include manquante  19. Vous avez déclaré préalablement un ensemble de fonc- tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions
<ul> <li>□ for(i=1;i&lt;5;i=i+1)</li> <li>16. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?</li> <li>□ char 'c';</li> <li>□ int char;</li> <li>□ char "c";</li> </ul>	est l'ordre :  dans lequel vous avez déclaré ces fonction alphabétique un ordre quelconque dans lequel ces fonctions sont appelées dans le main
□ char c;  17. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit- on chercher dans le programme? □ une accolade manquante □ un point-virgule en trop □ une accolade en trop	20. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :  ☐ int afficher_date(date_s d); ☐ void afficher_date(date_s d); ☐ void afficher_date(struct date_s d); ☐ void afficher_date(struct date_s d); ☐ character date = afficher_date(struct date_s d);
☐ un point-virgule manquant	<pre>    struct date_s afficher_date(struct date_s d);</pre>

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Dans la commande gcc, l'option -Wall signifie : □ qu'on veut changer alétoirement de fond d'écran □ qu'il faut indenter le fichier source ☐ que l'on veut voir tous les avertissements □ qu'il faut lancer un déboggueur 2. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme? ☐ un point-virgule manquant  $\square$  une accolade en trop  $\square$  une accolade manquante □ un point-virgule en trop 3. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\Box$  0  $\square$  0.5  $\square$  1.5  $\Box$  1 4. Quel est le problème d'un programme comportant les lignes suivantes? while (1) printf("coucou\n"); □ il comporte une boucle infinie  $\square$  il ne compile pas  $\square$  il risque d'afficher bonjour à la place de coucou □ il n'affiche rien 5. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :  $\square$  n = carre(int n);  $\square$  int carre(2);  $\square$  int n = carre();  $\square$  n = carre(n);

```
6. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
    ☐ #appart <stdlib.h>
    ☐ #include <stdio.h>
    ☐ #include <studio.h>
    ☐ #include <studlib.h>
7. Soit le programme principal suivant :
  int main()
   int a = 3;
   int b = 5;
   printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
   return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a:
  L'affichage dans le main est le suivant :
    \Box f(a,b)=13, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
8. Si factorielle est une fonction prenant en entrée un
  entier et renvoyant un entier, il est correct d'écrire :
    □ printf("%d", factorielle(n));
    \square n = factorielle();
    \square n = factorielle(p, q);
    ☐ int factorielle(int 2);
9. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
    ☐ #include <studlib.h>
    ☐ #include <stdio.h>
    ☐ #include <studio.h>
```

☐ #appart <stdlib.h>

```
10. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
   renvoie cet entier on écrit :
     ☐ int saisie_utilisateur();
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(int n);
     □ void saisie_utilisateur(char c);
11. Les lignes
   int i;
   int x=0;
   for(i=0,i<5,i=i+1)
      x=x+1;
     □ comportent une erreur qui ne sera pas détectée
     \square ne comportent aucune erreur
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
     □ comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
12. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
     int a = 0;
     int b = 20;
     int n;
     scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
   Quelle est la condition cond :
     □ (a<n) || (n>b)
     □ (a<=n) && (n<=b)
     \square (n<=a) && (n<=b)
```

□ a<=n<=b

13. Un bit est:	16. Pour l'extrait de programme suivant :	☐ Mineur
$\Box$ un battement d'horloge processeur	int i = 0;	□ Majeur
□ la longueur d'un mot mémoire	int j = 0;	☐ Mineur
☐ l'instruction qui met fin à un programme	for (i = 0; i < 2; i = i + 1)	Majeur
☐ un chiffre binaire (0 ou 1)  14. Soit un programme contenant les lignes suivantes :	for (j = 0; j < 3; j = j + 1) {	18. Pour déclarer une procée gument et qui ne renvoie
<pre>int i = 0; int j = 0;</pre>	<pre>printf("%d ", j); }</pre>	☐ void afficher_men
for $(i = 0; i < 2; i = i + 1)$	}	☐ int afficher_menu
{     for (j = 0; j < 3; j = j + 1)	qu'est ce qui sera affiché?	☐ int afficher_menu
{	□ 0 1 2 0 1 2 3	☐ double afficher_r
<pre>printf("%d ", i);</pre>	□ 0 1 2 3 0 1 2	☐ char afficher_men
}	□ 0 1 2 0 1 2	19. Pour déclarer une procéd
	□ 0 0 1 1 2 2 3	en argument un struct
qu'est ce qui sera affiché?	17. Le code suivant :	du struct, on écrit :
$\Box$ 1 2 1 2 3	int age = 15;	☐ int afficher_date
$\Box$ 0 1 2 0 1 2	if (age < 18)	□ void afficher_dat
$\Box$ 0 0 0 1 1 1	printf("Mineur\n");	☐ void afficher_dat
	}	□ struct date_s afi
15. Le bus système sert à :  ☐ Écrire des données sur le dique dur	else {	20. Le type des réels en C es
☐ transporter les processus du tourniquet au pro-	<pre>printf("Majeur\n");</pre>	□ real
cesseur	}	□ char
☐ Transférer des données et intructions entre pro-	<i>m</i> ,	□ int
cesseur et mémoire	affichera:	□ double
☐ Arriver à l'heure en cours	□ rien	

т		-1
	100000	- 1
	acence	

## Éléments d'informatique – contrôle continue

Prénom: Nom:  $N^{\circ}$  INE:

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :

```
\square int pgcd(int x, int x);
\square int pgcd(int x, y);
\Box int pgcd(int y, int x);
\square void pgcd(int x, int y);
```

2. Pour déclarer une fonction saisie\_utilisateur qui demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :

```
□ void saisie_utilisateur(int n);
□ void saisie_utilisateur(char c);
☐ int saisie_utilisateur();
□ saisie_utilisateur(scanf(%d));
```

3. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :

```
\square n = factorielle(p, q);
☐ int factorielle(int 2);
☐ printf("%d", factorielle(n));
\square n = factorielle();
```

4. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  $\mathbf{n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de n. Soit le programme suivant :

```
int a = 0;
 int b = 20;
 int n:
 scanf("%d", &n);
 while(cond)
   scanf("%d", &n);
Quelle est la condition cond :
  \square (a<n) || (n>b)
  □ a<=n<=b
  \square (n<=a) && (n<=b)
```

 $\square$  (a<=n) && (n<=b)

```
5. Le code suivant :
   int i:
   for (i = 4; i \ge 0; i = i - 1)
        printf("%d ", i);
   printf("\n");
  affichera:
    \square 4 3 2 1
    \Box 4 3 2 1 0
    \Box 01234
```

6. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :

```
\square en temps d'accès
\square des processus
□ les fichiers du disque
```

 $\square$  1 2 3 4

□ certaines données de la mémoire de travail

7. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?

```
\square (A == TRUE) && (B == TRUE)
     \square (!A || B)
     □ A && B
     \square !(!A | | B) == (A && !B)
8. Soit la fonction g définie par :
   int g(int a)
```

```
printf("a = \n", %d);
if (1 > 0)
  return 5;
return 7;
```

Alors l'expression g(0) prendra la valeur :

```
\Box 7
     \Box 5
     \square 0
9. Le code suivant :
    int i;
    for (i = 4; i > 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 01234
     \square 4 3 2 1
     \square 0 1 2 3
     \Box \ 4\ 3\ 2\ 1\ 0
```

10. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?

```
□ un point-virgule en trop
\square une accolade manquante
□ un point-virgule manquant
```

 $\square$  une accolade en trop

11. Après exécution jusqu'à la ligne 15 du programme C:

```
10
     int main() {
11
12
          int x = 5;
13
14
          x = 3 * x + 1:
15
16
17
    }
    \square le programme affiche x
```

 $\square$  la variable x vaut  $-\frac{1}{2}$ 

□ la variable x vaut 16

□ le programme affiche \*\*\*\*

12. Afin de représenter la taille d'un tableau, définir une	15. Si le code :	18. Pour l'extrait d
constante symbolique N valant 3.	struct toto_s	int produit
$\square$ #define N = 3	{	int serie[4
☐ #define N 3	<pre>int n; double x;</pre>	for (i = 0; {
$\square$ #define taille = 3	};	produit =
$\square$ #define taille = N	précède la fonction main(), alors on peut écrire en début de main() :	} printf("pro
13. Quel est le problème d'un programme comportant les lignes suivantes?	□ toto_s n, x;	La valeur affich
· ·	☐ int toto.n = 3;	$\Box$ 4
while (1) {	☐ struct toto_s toto;	□ 8
ι printf("coucou\n");	$\Box$ toto_s struct z = {3, 0.5};	$\Box$ 16
}	$\square$ int struct toto_s = {3, -1e10};	$\Box$ 0
$\hfill\Box$ il risque d'afficher bonjour à la place de coucou	16. Avant de faire appel à une fonction il est nécessaire de :	19. Le bus système ☐ Transférer
$\square$ il ne compile pas	□ l'avoir déclarée	cesseur et
□ il n'affiche rien	☐ l'avoir déclarée et définie	□ Écrire des
☐ il comporte une boucle infinie	□ l'avoir définie	☐ Arriver à l
□ if comporte time boucle immine	$\hfill\Box$ avoir défini une constante symbolique de la taille	□ transporte
14. Un enregistrement permet de grouper plusieurs valeurs	de cette fonction	cesseur
dans:	17. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	20. Le langage C es
$\square$ ses champs	☐ int char;	□ lu, écrit, p
$\square$ ses blocs	☐ char 'c';	□ compilé
□ ses cases	□ char c;	□ interprété
$\square$ ses chants	□ char "c";	□ composé
·		

```
de programme suivant :
 = 0;
4] = \{2, 2, 2, 2\};
 i < 4; i = i + 1)
= produit * serie[i];
oduit = %d", produit);
hée est :
sert à :
des données et intructions entre pro-
mémoire
données sur le dique dur
l'heure en cours
er les processus du tourniquet au pro-
est un langage
parlé
```

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square int mccarthy(int 2);
     \square x = mccarthy(n);
     \square n = mccarthy(p, q);
     \square n = mccarthy();
2. Un registre du processeur est :
     □ une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     □ une unité de calcul spécialisée de l'ordinateur
     □ une gamme de fréquence de fonctionnement du
        processeur
     \square un composant qui contient la liste des fichiers du
        système
3. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
   recommence la saisie de n. Soit le programme suivant :
    int a = 0;
    int b = 20;
    int n;
    scanf("%d", &n);
    while(cond)
       scanf("%d", &n);
   Quelle est la condition cond :
     \square (a<n) || (n>b)
     \square (n<=a) && (n<=b)
     □ a<=n<=b
     \square (a<=n) && (n<=b)
4. Soit la fonction f définie par :
```

int f(int a)

if (a > 0)

 $printf("a = \n", %d);$ 

```
return 3:
     return 4:
   Alors l'expression f(0) prendra la valeur :
     \Box 0
     \square 3
5. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
     \square int k;
     \square int %d;
     \square int loop n;
     □ loop i;
6. Le langage C est un langage
     □ composé
     □ lu, écrit, parlé
     □ compilé
     □ interprété
7. L'ordonnancement par tourniquet permet :
     □ d'entretenir l'illusion que les processus tournent
        en parallèle
     ☐ de ne pas perdre de temps avec la commutation
        de contexte
     □ d'afficher des ronds colorés à l'écran
     ☐ de doubler la mémoire disponible
8. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 4
     \square 3
     \square 101
     \Box 5
9. Le type des réels en C est :
     \square int
     □ real
     □ double
```

□ char

```
10. Après exécution du programme :
       lecture 8 r0
        valeur 3 r1
        mult r1 r0
        valeur 1 r2
        add r2 r0
        ecriture r0 8
        stop
      □ la case mémoire 8 contiendra 16
      \square le bus explose
      □ la case mémoire 8 contiendra 0
      □ le terminal affiche 8
11. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      □ main
      \square include
      \square init
      □ begin
12. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il ne compile pas
      □ il n'affiche rien
      ☐ il risque d'afficher bonjour à la place de coucou
      □ il comporte une boucle infinie
13. Pour déclarer une fonction exposant qui prend en ar-
    gument un réel x et un entier positif n et renvoie la
    valeur de x^n on écrit :
      \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
      \square double exposant(double x, int n);
      \square void exposant(double x^n);
```

14. Sous unix (ou linux), pour créer un répertoire TP4	17. Si a et b sont deux variables de type:	ret
dans le répertoire courant on peut utiliser la com-	struct toto_s	}
mande:	{	retur
$\square$ mkdir TP4	int n;	}
$\square$ kwrite TP4	<pre>double x;</pre>	Alors l'e
□ new TP4	};	
$\square$ yppasswd	Alors pour tester l'égalité de a et de b on utilise la	$\square$ 4
15. Laquelle des analyses suivantes ne fait pas partie des		
étapes de la compilation :	$\square (a.n == b.n) \&\& (a.x == b.x)$	
□ analyse sémantique	□ a = b	$\Box$ 5
$\square$ analyse harmonique	□ a == b	20. Le code
□ analyse syntaxique	$\Box a\{n, x\} == b\{n, x\}$	int so
□ analyse lexicale	18. Un enregistrement permet de grouper plusieurs valeurs	int i;
16. Le code suivant :	dans:	for (i
int i;	□ ses blocs	{
for (i = 0; i < 5; i = i + 1) {	$\square$ ses chants	somm }
<pre>printf("%d ", i);</pre>	$\square$ ses champs	printi
}	□ ses cases	
<pre>printf("\n");</pre>	19. Soit la fonction f définie par :	affichera
affichera:	int f(int a)	
$\square \ 4\ 3\ 2\ 1$	f (Int a)	
$\Box \ 4\ 3\ 2\ 1\ 0$	<pre>printf("a = \n", %d);</pre>	$\Box$ 42
$\square$ 0 1 2 3	if (a > 0)	□ 6
$\Box \ 0\ 1\ 2\ 3\ 4$	{	□ 1

```
eturn f(a - 1) + 1;
ırn 4;
'expression f(1) prendra la valeur :
suivant:
somme = 0;
(i = 1; i < 4; i = i + 1)
mme = somme + i;
f("%d", somme);
ra:
```

т		-1
	acence	

 $\Box$  16

# Éléments d'informatique – contrôle continue

Prénom :	Nom:	
$N^{\circ}$ INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Quel est le problème d'un programme comportant les lignes suivantes? while (1) printf("coucou\n");  $\square$  il ne compile pas  $\Box$  il risque d'afficher bonjour à la place de coucou □ il n'affiche rien  $\square$  il comporte une boucle infinie 2. Pour l'extrait de programme suivant : int somme = 0: for (i = 0; i < 5; i = i + 1)somme = somme + i;i = i + 1; /\* attention ! \*/ printf("somme = %d",somme); La valeur de somme affichée est :  $\Box$  10  $\Box$  0  $\Box$  6  $\square$  15 3. Pour l'extrait de programme suivant : int produit = 1; int  $serie[4] = \{2, 2, 2, 2\};$ for (i = 0; i < 4; i = i + 1)produit = produit \* serie[i]; printf("produit = %d", produit); La valeur affichée est :  $\Box$  4  $\Box$  0  $\square$  8

```
4. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
     \square #define taille = 3
     ☐ #define taille = N
     \square #define N = 3
     \square #define N 3
5. Pour l'extrait de programme suivant :
    int i = 0;
    int j = 0;
    for (i = 0; i < 2; i = i + 1)
        for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
    }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 0 1 2
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 3 0 1 2
6. Pour déclarer une fonction exposant qui prend en ar-
   gument un réel x et un entier positif n et renvoie la
   valeur de x^n on écrit :
     \square int exposant(double n, int x);
     \square exposant(double x, int n, int r);
    \square double exposant(double x, int n);
     \square void exposant(double x^n);
7. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 8
     □ 111
     \square 7
     \square 3
```

```
8. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      □ l'avoir définie
      □ l'avoir déclarée
 9. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square int n = pgcd();
     \square n = pgcd(int p, int q);
      \square n = pgcd(n, 3);
      \square int pgcd(2);
10. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt :
      \square for(i=0;i<=5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
      \square for(i=1;i<5;i=i+1)
      \square for(i=0;i<5;i=i+1)
11. Une variable booléenne est un variable :
      ☐ jamais nulle
      □ NaN (not a number, qui n'est pas un nombre)
      \square qui est vraie ou fausse
      □ à laquelle une valeur vient d'être affectée
      ☐ réelle positive
12. Pour déclarer une variable qui sera utilisée comme va-
    riable de boucle on peut utiliser l'instruction
      □ loop i;
      \square int k;
      \square int loop n;
      \square int %d;
13. Si n est une variable entière pour demander sa valeur
    à l'utilisateur, on utilise plutôt :
      □ scanf("%d", &n);
      □ un débogueur
      □ printf("Valeur de n ? %d\n", n);
      ☐ printf("Valeur de n ? %g\n", n);
```

```
14. Après exécution jusqu'à la ligne 15 du programme C:
 10
       int main() {
 11
 12
           int x = 5;
 13
 14
           x = 3 * x + 1:
 15
 16
 17
      }
      \Box le programme affiche ****
     \square le programme affiche x
     \square la variable x vaut -\frac{1}{2}
      □ la variable x vaut 16
15. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc prog.c -o -Wall prog.exe
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.exe -o prog.c
16. Soit la fonction f définie par :
   int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1:
      return 4;
   Alors l'expression f(1) prendra la valeur :
```

```
\Box 0
      \Box 4
      \Box 1
      \Box 5
17. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
      \square 3
      \Box 4
      \Box 0
18. Dans la commande gcc, l'option -Wall signifie :
     □ qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
     ☐ que l'on veut voir tous les avertissements
19. Pour l'extrait de programme suivant :
     int i = 0;
     int i = 0;
     for (i = 0; i < 3; i = i + 1)
         for (j = 0; j < 2; j = j + 1)
```

```
{
              printf("%d ", i);
         }
     }
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 0 1 1 2 2
     \Box 0 1 2 0 1 2
     \Box 1 2 3 1 2
     \Box 0 1 0 1 0 1
20. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
```

 $\Box$  f(a,b)=8, a=8, b=5

 $\Box$  0

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
$N^{\circ}$ INE :	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes. 1. Après exécution du programme : lecture 8 r0 valeur 3 r1 mult r1 r0 valeur 1 r2 add r2 r0 ecriture r0 8 stop 5 □ la case mémoire 8 contiendra 0  $\square$  le bus explose  $\Box$  le terminal affiche 8 □ la case mémoire 8 contiendra 16 2. Sur unix (ou linux), la commande mkdir permet de : □ créer un répertoire  $\square$  créer un fichier texte □ changer de répertoire courant  $\Box$  ouvrir un fichier texte 3. Soit la fonction f définie par : int f(int a) printf("a =  $\n$ ", %d); if (a > 0){ return 3; } return 4; Alors l'expression f(0) prendra la valeur :  $\Box$  4  $\Box$  0  $\square$  3 4. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\square$  1.5  $\Box$  1  $\square$  0.5

F T1/ 1: 444 11 1 1 1
5. L'écriture <u>111</u> en binaire correspond au nombre natu-
rel:
$\Box$ 7
$\square$ 3
□ 8
□ 111
6. Si <i>n</i> est une variable entière pour demander sa valeur
à l'utilisateur, on utilise plutôt :
$\square$ un débogueur
□ scanf("%d", &n);
☐ printf("Valeur de n ? %g\n", n);
☐ printf("Valeur de n ? %d\n", n);
7. Laquelle de ces écritures correspond à la déclaration
d'une variable de type caractère en langage C?
☐ char "c";
□ int char:
,
□ char 'c';
□ char c;
8. Au début de la fonction main() on place le code :
char b = 'A';
b = b + 2;
<pre>printf("%c\n", b);</pre>
Alors l'affichage sera :
□ b
□ A
<del></del>
□ C
$\square$ B
9. Après exécution jusqu'à la ligne 15 du programme C :
10 int main() {
11 int $x = 5$ ;
12 int y = 3;
13
14   x = y;
15
16
17 }
□ le programme affiche "Faux"
$\Box$ la variable x vaut 3
□ la variable y vaut 5
*
$\square$ la variable x vaut 5 et la variable y vaut 3

10.	Laquelle des analyses suivantes ne fait pas partie de étapes de la compilation :
	$\Box$ analyse lexicale
	$\Box$ analyse sémantique
	$\Box$ analyse harmonique
	$\Box$ analyse syntaxique
11.	L'ordonnancement par tourniquet permet :
	$\Box$ de doubler la mémoire disponible
	$\Box$ d'afficher des ronds colorés à l'écran
	$\Box$ d'entretenir l'illusion que les processus tournen en parallèle
	$\Box$ de ne pas perdre de temps avec la commutation de contexte
12.	L'écriture $\underline{101}$ en binaire correspond au nombre naturel :
	$\Box$ 4
	□ 101
	□ 5
	$\square$ 3
13.	Au début de la fonction main() on place le code :
	<pre>char i; for (i = 'A'; i &lt;= 'F'; i = i + 1) {    printf("%c", i); }</pre>
	<pre>printf("\n");</pre>
	Alors l'affichage sera :
	□ A
	□ ABCDEF
	□ сссссс
	П;

14. Un registre du processeur est :  ☐ un composant qui contient la liste des fichiers du	un désaccord entre la déclaration et la définition d'une fonction	19. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :
système  une gamme de fréquence de fonctionnement du processeur	☐ une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration ☐ une directive préprocesseur #include manquante	$\Box$ printf("x=%d et y=%d\n",x y);
processeur  une unité de calcul spécialisée de l'ordinateur  une case mémoire interne au processeur qui sera manipulée directement lors des calculs  15. Soit la fonction g définie par :  int g(int a) {     printf("a = \n", %d);     if (1 > 0)     {         return 5;     }     return 7; }  Alors l'expression g(0) prendra la valeur :	□ une directive préprocesseur #include manquante  17. Pour l'extrait de programme suivant :  int produit = 0;  int serie[4] = {2, 2, 2, 2};  for (i = 0; i < 4; i = i + 1)  {     produit = produit * serie[i];    }     printf("produit = %d", produit);  La valeur affichée est :  □ 0  □ 16 □ 8 □ 4  18. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.  □ #define taille = 3 □ #define N 3 □ #define taille = N □ #define N = 3	<pre>□ printf("x=%d et y=%d\n,x,y"); □ printf("x=%x et y=%y\n"); □ printf("x=%d et y=%d\n",x,y);  20. Soit un programme contenant les lignes suivantes:     int i = 0;     int j = 0;     for (i = 0; i &lt; 3; i = i + 1)     {         for (j = 0; j &lt; 5; j = j + 1)         {          }     }     printf("j = %d\n", j);  qu'est ce qui sera affiché par ce printf? □ j = 5 □ j = %d □ j = 4 □ j = 0</pre>

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes.

1. Un registre du processeur est : □ une unité de calcul spécialisée de l'ordinateur  $\square$  un composant qui contient la liste des fichiers du système □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs □ une gamme de fréquence de fonctionnement du processeur 2. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :  $\square$  0.5  $\square$  1.5  $\Box$  0  $\Box$  1 3. Si a et b sont deux variables de type : struct toto\_s int n; double x; }; Alors pour tester l'égalité de a et de b on utilise la condition:  $\square$  a{n, x} == b{n, x}  $\Box$  a = b □ a == b  $\square (a.n == b.n) & (a.x == b.x)$ 4. Le bus système sert à : □ Transférer des données et intructions entre processeur et mémoire □ transporter les processus du tourniquet au processeur ☐ Arriver à l'heure en cours

☐ Écrire des données sur le dique dur

```
5. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
  Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
    □ l'analyse des entrées clavier
    ☐ l'analyse sémantique
    ☐ l'analyse harmonique
    □ l'édition de liens
6. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3:
     return 4;
   Alors l'expression f(0) prendra la valeur :
    \square 0
    \Box 4
    \square 3
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
    □ char 'c';
    □ char c;
    ☐ int char;
    ☐ char "c";
8. Un bit est:
    □ un battement d'horloge processeur
    ☐ l'instruction qui met fin à un programme
    \square un chiffre binaire (0 ou 1)
```

 $\Box$  la longueur d'un mot mémoire

```
9. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      □ il risque d'afficher bonjour à la place de coucou
      \square il n'affiche rien
      \square il comporte une boucle infinie
      \square il ne compile pas
10. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
          printf("%d ", i);
     printf("\n");
    affichera:
      \Box \ 4\ 3\ 2\ 1\ 0
      \Box 01234
      \square 4 3 2 1
      \Box 0123
11. Si pgcd est une fonction prenant en entrée deux entiers
    et renvoyant un entier, il est correct d'écrire :
      \square n = pgcd(n, 3);
      \square int n = pgcd();
      \square int pgcd(2);
      \square n = pgcd(int p, int q);
12. Après exécution jusqu'à la ligne 15 du programme C :
  10
       int main() {
 11
            int x = 5;
 12
            int y;
  13
  14
            y = x;
  15
  16
```

17

}

```
\square la variable x vaut 5 et la variable y vaut 0
                                                           15. Si factorielle est une fonction prenant en entrée un
                                                                                                                           qu'est ce qui sera affiché?
                                                               entier et renvoyant un entier, il est correct d'écrire :
      \square la variable x vaut 0
                                                                                                                             \Box 0 1 2 0 1 2
                                                                 □ printf("%d", factorielle(n));
     \square le programme affiche "Faux"
                                                                                                                             \Box 0 0 1 1 2 2
                                                                 \square n = factorielle(p, q);
     \square la variable v vaut 5
                                                                                                                             □ 1 2 3 1 2
                                                                 \square n = factorielle():
13. Soit le programme principal suivant :
                                                                                                                             \Box 0 1 0 1 0 1
                                                                 ☐ int factorielle(int 2);
   int main()
                                                           16. Pour déclarer une fonction saisie_utilisateur qui
                                                                                                                       19. La virtualisation de la mémoire permet notamment de
                                                                demande à l'utilisateur d'entrer un entier au clavier et
                                                                                                                           stocker des portions inactives de la mémoire de travail
     int a = 3;
                                                               renvoie cet entier on écrit :
                                                                                                                           sur le disque dur. Mais on perd :
     int b = 5:
                                                                 □ void saisie_utilisateur(char c);
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                             □ les fichiers du disque
                                                                 ☐ int saisie_utilisateur();
     return EXIT_SUCCESS;
                                                                 □ void saisie_utilisateur(int n);
                                                                                                                             \square des processus
                                                                 ☐ saisie_utilisateur(scanf(%d));
                                                                                                                             □ en temps d'accès
   appelant la fonction f ainsi définie :
                                                           17. Pour afficher à l'aide de printf("%d\n",tab[i]);
                                                                                                                             □ certaines données de la mémoire de travail
   int f(int a, int b)
                                                               le contenu d'un tableau de 5 entiers initialisé au
                                                               préalable, on utilise plutôt :
                                                                                                                       20. Soit la fonction g définie par :
      a = a + b;
                                                                 \square for(i=0;i<=5;i=i+1)
      return a;
                                                                                                                           int g(int a)
                                                                 ☐ for(i=1;i<5;i=i+1)
                                                                 \square for(i=0:i<5:i=i+1)
   L'affichage dans le main est le suivant :
                                                                                                                             printf("a = \n", %d);
                                                                 \square for(i=1;i<=5;i=i+1)
     \Box f(a,b)=8, a=3, b=5
                                                                                                                             if (1 > 0)
                                                           18. Pour l'extrait de programme suivant :
                                                                                                                             {
     \Box f(a,b)=8, a=8, b=5
                                                                                                                                return 5;
                                                                int i = 0:
     \Box f(3,5)=8, a=3, b=5
                                                                int j = 0;
     \Box f(a,b)=13, a=8, b=5
                                                                for (i = 0; i < 3; i = i + 1)
                                                                                                                             return 7;
14. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
                                                                     for (j = 0; j < 2; j = j + 1)
                                                                                                                           Alors l'expression g(0) prendra la valeur :
      ☐ #include <studio.h>
                                                                                                                             \Box 5
                                                                          printf("%d ", i);
      ☐ #appart <stdlib.h>
                                                                     }
                                                                                                                             \square 7
      ☐ #include <studlib.h>
                                                                                                                             \Box 0
     ☐ #include <stdio.h>
                                                                printf("\n");
```

# Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Soient deux variables entières x et y initialisées à 4 et
   5 respectivement. L'affichage x=4 et y=5 est obtenu
   avec la commande :
     \square printf("x=%d et y=%d\n,x,y");
    \square printf("x=%d et y=%d\n",x y);
    \square printf("x=%d et y=%d\n",x,y);
     \square printf("x=%x et y=%y\n");
2. Le code suivant :
    int somme = 0;
    int i;
    for (i = 1; i < 4; i = i + 1)
      somme = somme + i;
    printf("%d", somme);
   affichera:
     \Box 6
     \Box 1
     \Box 0
     \square 42
3. Une variable booléenne est un variable :
     □ qui est vraie ou fausse
     □ à laquelle une valeur vient d'être affectée
     ☐ réelle positive
     □ NaN (not a number, qui n'est pas un nombre)
     □ jamais nulle
4. Si cette erreur apparaît à la compilation :
   error: expected ';' before '}' token que doit-
   on chercher dans le programme?
     ☐ un point-virgule manquant
     \square une accolade en trop
     ☐ un point-virgule en trop
     \square une accolade manquante
```

```
5. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il comporte une boucle infinie
     □ il n'affiche rien
     \square il ne compile pas
6. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
   , que doit-on chercher dans le programme?
     \Box une directive préprocesseur #include manquante
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     \square une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
7. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return 3;
     return 4;
   Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 4
     \Box 0
8. Au début de la fonction main() on place le code :
    char i;
    for (i = 'A'; i \le 'F'; i = i + 1)
      printf("%c", i);
```

printf("\n");

```
Alors l'affichage sera:
      \square A
      □ ccccc
      ☐ ABCDEF
      \Box i
 9. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
        return f(a - 1) + 1;
      return 4;
    Alors l'expression f(1) prendra la valeur :
      \square 5
      \Box 4
      \Box 1
      \Box 0
10. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      □ ses chants
      \square ses blocs
      \square ses champs
      \square ses cases
11. Pour déclarer une procédure afficher_date qui prend
    en argument un struct date_s et affiche le contenu
    du struct, on écrit :
      □ void afficher_date(date_s d);
      □ void afficher_date(struct date_s d);
     ☐ struct date_s afficher_date(struct date_s d);
      ☐ int afficher_date(date_s d);
12. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ l'avoir définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      □ l'avoir déclarée
```

13. Sous unix (ou linux), pour créer un répertoire TP4
dans le répertoire courant on peut utiliser la com-
mande:
□ mkdir TP4
☐ new TP4
☐ kwrite TP4
$\square$ yppasswd
14. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les expressions booléennes suivantes, laquelle a pour valeur TRUE?
□ !(!A    B) == (A && !B)
□ : (:A     B) (A && :B) □ A && B
☐ (!A    B)
$\Box (A == TRUE) \&\& (B == TRUE)$
15. Après exécution jusqu'à la ligne 14 du programme C :
10 int main() {
11 int x = 5;
12 13 printf(" x = %d\n", 2);
13 printf(" x = %d\n", 2); 14
15
16 }
$\Box$ le terminal affiche x = 2
$\Box$ le terminal affiche x = 5
□ le terminal affiche 5
□ le terminal affiche "Faux"

```
16. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
    printf("Majeur\n");
    affichera:
     ☐ Mineur
     \square rien
     □ Majeur
     \square Mineur
        Majeur
17. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
     \square que l'on veut voir tous les avertissements
     □ qu'on veut changer alétoirement de fond d'écran
18. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise :
     □ void afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu();
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu(int char);
```

```
19. Le code suivant :
     int i;
     for (i = 1; i < 5; i = i + 1)
          printf("%d ", i);
     printf("\n");
    affichera:
      \Box 1 2 3 4
      \square 4 3 2 1
      \Box 01234
      \Box \ 4\ 3\ 2\ 1\ 0
20. Si a et b sont deux variables de type:
    struct toto_s
      int n;
      double x;
    };
    Alors pour tester l'égalité de a et de b on utilise la
    condition:
      \square a\{n, x\} == b\{n, x\}
      \Box a = b
      □ a == b
```

 $\square (a.n == b.n) && (a.x == b.x)$ 

## Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° INE ·	
IN IIND.	

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

- 1. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au préalable, on utilise plutôt :
  - ☐ for(i=1;i<=5;i=i+1) ☐ for(i=1;i<5;i=i+1) ☐ for(i=0;i<5;i=i+1)

 $\square$  for(i=0;i<=5;i=i+1)

- 2. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :
  - □ printf("x=%d et y=%d\n",x,y);
    □ printf("x=%d et y=%d\n",x y);
    □ printf("x=%d et y=%d\n,x,y");
    □ printf("x=%x et y=%y\n");
- 3. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 0; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
    }
}
printf("j = %d\n", j);
...
}</pre>
```

qu'est ce qui sera affiché?

```
\Box j = \%d

\Box j = 0

\Box j = 5

\Box j = 4
```

4. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :

```
□ n = pgcd(n, 3);
□ n = pgcd(int p, int q);
□ int pgcd(2);
□ int n = pgcd();
```

5. Soit un programme contenant les lignes suivantes :

```
int i = 0;
int j = 0;
for (i = 0; i < 3; i = i + 1)
{
    for (j = 0; j < 5; j = j + 1)
    {
        ...
}</pre>
```

qu'est ce qui sera affiché par ce printf?

 $printf("j = %d\n", j);$ 

- $\Box j = 4$   $\Box j = %d$   $\Box j = 0$
- ☐ j = 5
  6. Pour déclarer une procédure afficher\_menu sans ar-

gument et qui ne renvoie rien on utilise :

```
□ void afficher_menu();
□ int afficher_menu();
□ int afficher_menu(int char);
□ char afficher_menu(printf("menu"));
□ double afficher_menu();
```

7. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :

```
□ alphabétique
```

- $\Box$  un ordre quel conque
- $\Box$  dans lequel ces fonctions sont appelées dans le main
- $\Box$  dans lequel vous avez déclaré ces fonction
- 8. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

□ char 'c';
$\square$ char c;
$\square$ int char;
Char "c".

- 9. Sous unix (ou linux), la commande 1s permet de :
  - $\square$  compiler un programme
  - $\square$  afficher le contenu d'un fichier texte
  - □ voir des clips musicaux
  - $\Box$  afficher la liste de fichiers contenus dans un répertoire
- 10. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct d'écrire :

```
□ n = carre(n);
□ int n = carre();
□ n = carre(int n);
□ int carre(2);
```

11. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction

```
□ loop i;
□ int %d;
□ int loop n;
□ int k;
```

12. On souhaite faire une boucle de contrôle de saisie : tant que l'entier  ${\tt n}$  n'appartient pas à l'intervalle [a..b], on recommence la saisie de  ${\tt n}$ . Soit le programme suivant :

```
int a = 0;
int b = 20;
int n;
scanf("%d", &n);
while(cond)
{
   scanf("%d", &n);
}
```

Quelle est la condition cond :

$(a \le n)$	&&	(n<=b)
(n<=a)	&&	(n<=b)
(a <n)< td=""><td>11 (</td><td>(n&gt;b)</td></n)<>	11 (	(n>b)

□ a<=n<=b

```
13. Soit la fonction f définie par :
                                                              16. Soit la fonction g définie par :
                                                                                                                                  \square la variable x vaut 0
                                                                                                                                  \square la variable y vaut 5
    int f(int a)
                                                                  int g(int a)
    {
                                                                                                                                  \square la variable x vaut 5 et la variable y vaut 0
      printf("a = \n", %d);
                                                                    printf("a = \n", %d);
      if (a > 0)
                                                                    if (1 > 0)
                                                                                                                            19. Le code suivant :
                                                                    {
                                                                                                                                 int i:
        return f(a - 1) + 1;
                                                                       return 5;
                                                                                                                                 for (i = 4; i \ge 0; i = i - 1)
      return 4;
                                                                    return 7;
                                                                                                                                      printf("%d ", i);
    Alors l'expression f(1) prendra la valeur :
                                                                  Alors l'expression g(0) prendra la valeur :
                                                                                                                                 printf("\n");
      \Box 5
                                                                    \Box 5
                                                                                                                                affichera:
      \Box 1
                                                                    \Box 7
                                                                                                                                  \square 4 3 2 1
      \Box 0
                                                                    \Box 0
                                                                                                                                  \Box \ 4\ 3\ 2\ 1\ 0
      \Box 4
                                                             17. Si n est une variable entière pour demander sa valeur
                                                                                                                                  \square 1 2 3 4
14. Laquelle des analyses suivantes ne fait pas partie des
                                                                  à l'utilisateur, on utilise plutôt :
                                                                                                                                  \Box 01234
    étapes de la compilation :
                                                                    □ un débogueur
      \square analyse sémantique
                                                                                                                            20. Pour l'extrait de programme suivant :
                                                                    ☐ printf("Valeur de n ? %d\n", n);
      \square analyse lexicale
                                                                                                                                  int somme = 0;
                                                                    ☐ printf("Valeur de n ? %g\n", n);
      \square analyse harmonique
                                                                                                                                  for (i = 0; i < 5; i = i + 1)
                                                                    □ scanf("%d", &n);
      \square analyse syntaxique
                                                              18. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                                     somme = somme + i;
15. Au début de la fonction main() on place le code :
                                                                                                                                    i = i + 1; /* attention ! */
                                                                     int main() {
                                                                10
     char b = 'A';
                                                                11
                                                                          int x = 5;
     b = b + 2;
                                                                                                                                  printf("somme = %d",somme);
                                                                12
     printf("%c\n", b);
                                                                          int y;
                                                                                                                                La valeur de somme affichée est :
                                                                13
    Alors l'affichage sera :
                                                                14
                                                                          y = x;
                                                                                                                                  \Box 0
      \square B
                                                                15
                                                                                                                                  \Box 10
      \Box C
                                                                16
                                                                17
                                                                     }
                                                                                                                                  \square 15
      \square A
                                                                                                                                  \Box 6
      □ b
                                                                    □ le programme affiche "Faux"
```

# Éléments d'informatique – contrôle continue

Prénom : Nom : N° INE :

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Pour l'extrait de programme suivant :
     int somme = 0;
     for (i = 0; i < 5; i = i + 1)
        somme = somme + i;
       i = i + 1; /* attention ! */
     printf("somme = %d",somme);
   La valeur de somme affichée est :
     \Box 6
     \square 10
     \Box 0
     \square 15
2. Le langage C est un langage
     □ compilé
     □ lu, écrit, parlé
     □ composé
     □ interprété
3. Le code suivant :
    int i;
    for (i = 8; i > 0; i = i - 2)
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 8 6 4 2
     \Box 02468
     \square 8 2
     \Box 8 6 4 2 0
4. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square int mccarthy(int 2);
     \square x = mccarthy(n);
     \square n = mccarthy();
     \square n = mccarthy(p, q);
```

```
5. Soit un programme contenant les lignes suivantes :
    int i = 0;
    int j = 0;
    for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
         }
    printf("j = %d\n", j);
   qu'est ce qui sera affiché?
     \Box j = 0
     \Box j = 5
     \Box j = 4
     \Box j = %d
6. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
     \square n = pgcd(n, 3);
    \square n = pgcd(int p, int q);
     \square int pgcd(2);
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
     □ ses cases
     □ ses chants
     \square ses champs
8. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x = racine(2/3);
    \square x = racine(racine(x)*racine(x));
```

 $\square$  x = racine(x \* x) - racine(x);

 $\square$  x - 1 = racine(x);

```
9. On considère deux variables booléennes A et B initia-
   lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
   TRUE?
      □ A && B
      ☐ (A == TRUE) && (B == TRUE)
      \square (!A || B)
      \square !(!A \mid | B) == (A \&\& !B)
10. Avant de faire appel à une fonction il est nécessaire
   de:
      □ l'avoir définie
      □ l'avoir déclarée
      □ l'avoir déclarée et définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
11. L'ordonnancement par tourniquet permet :
      ☐ d'entretenir l'illusion que les processus tournent
         en parallèle
      □ d'afficher des ronds colorés à l'écran
      □ de doubler la mémoire disponible
      ☐ de ne pas perdre de temps avec la commutation
         de contexte
12. Après exécution jusqu'à la ligne 15 du programme C:
 10
       int main() {
 11
            int x = 5;
 12
            int y;
 13
 14
            y = x;
 15
 16
 17
       }
      \Box la variable x vaut 0
      □ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 0
      \square la variable v vaut 5
```

13.	Au début de la fonction main() on place le code :
	<pre>char b = 'A'; b = b + 2; printf("%c\n", b);</pre>
	Alors l'affichage sera :
	□ C
	□В
	□ b
	□ A
14.	Soit la fonction g définie par :
	<pre>int g(int a) {</pre>
	<pre>printf("a = \n", %d); if (1 &gt; 0) {</pre>
	return 5;
	<pre>} return 7;</pre>
	}
	Alors l'expression g(0) prendra la valeur :
	□ 5
	□ 7
	$\Box$ 0
15.	Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :
	☐ int pgcd(int x, y);

```
\square void pgcd(int x, int y);
     \Box int pgcd(int y, int x);
     \Box int pgcd(int x, int x);
16. Pour l'extrait de programme suivant :
      int produit = 0;
      int serie[4] = \{2, 2, 2, 2\};
      for (i = 0; i < 4; i = i + 1)
        produit = produit * serie[i];
      printf("produit = %d", produit);
   La valeur affichée est :
     \Box 16
     \square 8
     \Box 0
     \Box 4
17. Le code suivant :
     int i;
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
    printf("\n");
    affichera:
     \Box 4 3 2 1 0
     \Box \ 0\ 1\ 2\ 3\ 4
     \Box 1 2 3 4
     \square 4 3 2 1
```

18.	Quel est le problème d'un programme comportant les lignes suivantes?
	while (1) {
	<pre>printf("coucou\n"); }</pre>
	□ il ne compile pas
	<ul><li>□ il risque d'afficher bonjour à la place de coucou</li><li>□ il n'affiche rien</li></ul>
	$\Box$ il comporte une boucle infinie
19.	Un fichier source est :
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\square$ un document de référence du système
	$\Box$ un document illisible pour les humains
	$\Box$ un document qui doit être protégé
	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
20.	Vous utilisez une boucle while quand :
	$\square$ vous avez déjà fait un for dans le même programme principal
	$\hfill \square$ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
	$\square$ vous n'avez pas déclaré de fonction
	☐ l'incrément de la variable de houcle n'est pas 1

т	•	-1
- 1	acence	- 1

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Le code suivant :
    int age = 15;
    if (age < 18)
        printf("Mineur\n");
    else
        printf("Majeur\n");
    }
  affichera:
     □ Mineur
     □ Mineur
       Majeur
     \square rien
     □ Majeur
2. Un programme en langage C doit comporter une et une
  seule définition de la fonction :
     \square begin
    □ main
     \square include
     \square init
3. Pour compiler un programme prog.c, on utilise la
  ligne de commande :
     ☐ gcc -Wall prog.c -o prog.exe
    ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
4. Pour déclarer une fonction pgcd qui calcule et renvoie
  le plus grand diviseur commun de deux entiers positifs
  passés en arguments on écrit :
    \square int pgcd(int x, y);
    \square int pgcd(int x, int x);
     \square int pgcd(int y, int x);
    \square void pgcd(int x, int y);
```

```
5. L'écriture 111 en binaire correspond au nombre natu-
  rel:
    \square 3
    \square 8
    □ 111
    \square 7
6. Si cette erreur apparaît à la compilation :
   erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
    □ un désaccord entre la déclaration et la définition
       d'une fonction
    ☐ une fonction appelée avant sa déclaration
    □ une fonction déclarée mais non définie
    ☐ une directive préprocesseur #include manquante
7. Si le code :
  struct toto_s
     int n:
     double x;
  };
  précède la fonction main(), alors on peut écrire en
  début de main():
    \square int toto.n = 3;
    \square int struct toto_s = {3, -1e10};
    \square toto_s n, x;
    □ struct toto_s toto;
    \Box toto_s struct z = {3, 0.5};
8. Si factorielle est une fonction prenant en entrée un
  entier et renvoyant un entier, il est correct d'écrire :
    ☐ int factorielle(int 2);
    \square n = factorielle();
    ☐ printf("%d", factorielle(n));
    \square n = factorielle(p, q);
9. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
  renvoie cet entier on écrit :
    □ void saisie_utilisateur(int n);
    ☐ int saisie_utilisateur();
    ☐ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
```

```
10. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int i = 0;
     for (i = 0; i < 2; i = i + 1)
          for (j = 0; j < 3; j = j + 1)
              printf("%d ", i);
     }
    qu'est ce qui sera affiché?
      \Box 0 0 0 1 1 1
      \Box 0 1 0 1 0 1 0 1
      \Box 1 2 1 2 3
      \Box 0 1 2 0 1 2
11. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \Box 1
      \Box 0
      \square 0.5
      \square 1.5
12. Un bit est:
      ☐ l'instruction qui met fin à un programme
      □ la longueur d'un mot mémoire
      \square un chiffre binaire (0 ou 1)
      □ un battement d'horloge processeur
13. Soit la fonction f définie par :
    int f(int a)
      printf("a = \n", %d);
      if (a > 0)
      {
        return f(a - 1) + 1;
      return 4;
```

Alors l'expression f(1) prendra la valeur :

```
\Box 4
     \Box 0
     \Box 1
     \Box 5
14. Le langage C est un langage
     □ lu, écrit, parlé
     □ composé
     □ compilé
     □ interprété
15. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 2; i = i + 1)
         for (j = 0; j < 3; j = j + 1)
             printf("%d ", j);
     }
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 0 1 2 0 1 2 3
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 3 0 1 2
16. Soit un programme contenant les lignes suivantes :
     int i = 0;
     int j = 0;
     for (i = 0; i < 0; i = i + 1)
         for (j = 0; j < 5; j = j + 1)
```

```
{
         }
     printf("j = %d\n", j);
     }
    qu'est ce qui sera affiché?
     □ j = 5
     \Box j = %d
     \Box j = 4
      \Box j = 0
17. Au début de la fonction main() on place le code :
     char b = 'A';
     b = b + 2;
     printf("%c\n", b);
    Alors l'affichage sera :
      □b
      \square A
      \Box C
      \square B
18. Le code suivant :
     for (i = 4; i \ge 0; i = i - 1)
     {
         printf("%d ", i);
     printf("\n");
    affichera:
      \square 4 3 2 1
```

```
\Box 43210
     \Box 01234
     \square 1 2 3 4
19. Pour afficher à l'aide de printf("%d\n",tab[i]);
   le contenu d'un tableau de 5 entiers initialisé au
   préalable, on utilise plutôt :
     ☐ for(i=0;i<5;i=i+1)
     \square for(i=0;i<=5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
     \square for(i=1;i<5;i=i+1)
20. Soit le programme principal suivant :
    int main()
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
   appelant la fonction f ainsi définie :
   int f(int a, int b)
      a = a + b;
      return a;
   L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
```

## Éléments d'informatique - contrôle continue

Prénom:	Nom:
NIO INII	1.0111
N° INE :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes.

```
1. Soit le programme principal suivant :
  int main()
    int a = 3;
    int b = 5;
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
  appelant la fonction f ainsi définie :
  int f(int a, int b)
     a = a + b;
     return a;
  L'affichage dans le main est le suivant :
     \Box f(a,b)=13, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
2. Vous utilisez une boucle while quand:
     □ vous n'avez pas déclaré de fonction
     □ vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
     □ vous avez déjà fait un for dans le même pro-
       gramme principal
     ☐ l'incrément de la variable de boucle n'est pas 1
3. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
     ☐ Mineur
     □ Majeur
```

```
\square rien
    □ Mineur
       Majeur
4. Pour compiler un programme prog.c, on utilise la
  ligne de commande :
    ☐ gcc -Wall prog.c -o prog.exe
    ☐ gcc prog.exe -Wall -o prog.c
    ☐ gcc -Wall prog.exe -o prog.c
    ☐ gcc prog.c -o -Wall prog.exe
5. On souhaite faire une boucle de contrôle de saisie : tant
  que l'entier n n'appartient pas à l'intervalle [a..b], on
  recommence la saisie de n. Soit le programme suivant :
   int a = 0;
   int b = 20;
    int n;
   scanf("%d", &n);
   while(cond)
      scanf("%d", &n);
  Quelle est la condition cond:
    ☐ a<=n<=b
    □ (n<=a) && (n<=b)
    \square (a<=n) && (n<=b)
    □ (a<n) || (n>b)
6. Si cette erreur apparaît à la compilation :
  erreur: conflicting types for 'max', que doit-
  on chercher dans le programme?
    □ un désaccord entre la déclaration et la définition
       d'une fonction
    □ une fonction déclarée mais non définie
    ☐ une fonction appelée avant sa déclaration
    ☐ une directive préprocesseur #include manquante
7. Sur unix (ou linux), la commande mkdir permet de :
    \square créer un fichier texte
    □ changer de répertoire courant
    □ ouvrir un fichier texte
    □ créer un répertoire
```

```
8. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
      □ struct int factorielle(int n);
      ☐ int factorielle(int x);
      ☐ int factorielle();
      ☐ int factorielle(double n);
 9. Le type des réels en C est :
      \square real
      □ double
      \square int
      □ char
10. Après exécution jusqu'à la ligne 15 du programme C:
 10
 11
       int main() {
 12
            int x = 5;
 13
 14
            x = 3 * x + 1;
 15
 16
 17
      \square le programme affiche x
      ☐ le programme affiche ****
      \square la variable x vaut -\frac{1}{2}
      □ la variable x vaut 16
11. Pour afficher à l'aide de printf("%d\n",tab[i]);
    le contenu d'un tableau de 5 entiers initialisé au
    préalable, on utilise plutôt :
      \square for(i=0;i<5;i=i+1)
      \square for(i=1;i<5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
      \square for(i=0;i<=5;i=i+1)
12. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 101
      \Box 5
      \square 3
      \Box 4
```

13.	Un enregistrement permet de grouper plusieurs valeurs dans :
	$\square$ ses chants
	□ ses cases
	$\square$ ses champs
	$\square$ ses blocs
14.	Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre :
	$\square$ alphabétique
	$\Box$ un ordre quel conque
	$\Box$ dans lequel ces fonctions sont appelées dans le main
	$\Box$ dans lequel vous avez déclaré ces fonction
15.	Un fichier source est :
	$\hfill\Box$ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	$\Box$ un document qui doit être protégé
	$\Box$ un fichier texte qui sera traduit en instructions processeur
	$\Box$ un document de référence du système
	$\Box$ un document illisible pour les humains

```
16. Lorsqu'un programme utilise printf ou scanf il faut
    qu'il contienne l'instruction préprocesseur :
      ☐ #appart <stdlib.h>
      ☐ #include <studio.h>
      ☐ #include <stdio.h>
      ☐ #include <studlib.h>
17. Si cet avertissement apparaît à la compilation :
    warning: implicit declaration of function 'max'
    , que doit-on chercher dans le programme?
      \square un désaccord entre la déclaration et la définition
         d'une fonction
      \square une fonction appelée avant sa déclaration
      \square une directive préprocesseur #include manquante
      \square une fonction déclarée mais non définie
18. Quel est l'opérateur de différence en C :
      □ !
      □ <>
      □ !=
     \square \neq
19. Pour l'extrait de programme suivant :
     int i = 0;
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
```

```
for (j = 0; j < 2; j = j + 1)
              printf("%d ", i);
         }
    printf("\n");
   qu'est ce qui sera affiché?
     \Box 0 1 2 0 1 2
     \Box 1 2 3 1 2
     \Box 0 1 0 1 0 1
     \Box 0 0 1 1 2 2
20. Après exécution jusqu'à la ligne 15 du programme C:
       int main() {
 11
            int x = 5;
 12
            int y;
 13
 14
            y = x;
 15
 16
 17
      }
     \square la variable x vaut 5 et la variable y vaut 0
     \Boxla variable y vaut 5
```

 $\Box$  la variable x vaut 0

 $\square$  le programme affiche "Faux"

т		-4
	acence	\ I

## Éléments d'informatique – contrôle continue

Prénom:	Nom:	
$N^{\circ}$ INE :		

Barème : 1 points par réponse juste (unique) ; -0.5 points par réponse fausse. Durée : 20 minutes.

```
1. Si racine est une fonction prenant en entrée un réel
   et renvoyant la racine carrée de cet réel, et que x est
   une variable réelle définie et initialisée, il est incorrect
   d'écrire :
     \square x - 1 = racine(x);
     \square x = racine(2/3):
     \square x = racine(x * x) - racine(x);
     \square x = racine(racine(x)*racine(x)):
2. Soit la fonction f définie par :
   int f(int a)
     printf("a = \n", %d);
     if (a > 0)
       return f(a - 1) + 1;
     return 4;
   Alors l'expression f(1) prendra la valeur :
     \square 5
     \Box 0
     \Box 1
     \Box 4
3. Pour l'extrait de programme suivant :
    int i;
    int j;
    for(i=4;i>0;i=i-1)
      for(j=i;j<6;j=j+1)
         printf("*");
      printf(" ");
   qu'est ce qui sera affiché?
```

```
 ***** *** ***
4. Un fichier source est:
    ☐ un fichier que l'ont doit citer dans les documents
       produits sur l'ordinateur
    □ un document de référence du système
    \square un document illisible pour les humains
    □ un document qui doit être protégé
    □ un fichier texte qui sera traduit en instructions
       processeur
5. Pour déclarer une procédure afficher_date qui prend
   en argument un struct date_s et affiche le contenu
   du struct, on écrit :
    □ void afficher_date(struct date_s d);
    □ struct date_s afficher_date(struct date_s d);
    □ void afficher_date(date_s d);
    ☐ int afficher_date(date_s d);
6. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    printf("Majeur\n");
  affichera:
    □ Mineur
    ☐ Mineur
       Majeur
    □ Majeur
    \square rien
7. Sous unix (ou linux), la commande cd permet de :
    \Box\,récupérer un programme arrêté avec la commande
    □ ouvir un bureau partagé (common desktop)
    □ détruire un fichier
    □ jouer de la musique
    □ changer de répertoire courant
```

```
8. Soit la fonction g définie par :
   int g(int a)
      printf("a = \n", %d);
      if (1 > 0)
      {
        return 5;
      }
      return 7;
   Alors l'expression g(0) prendra la valeur :
      \Box 0
      \square 5
      \square 7
 9. Quel est le problème d'un programme comportant les
   lignes suivantes?
    while (1)
      printf("coucou\n");
      \square il comporte une boucle infinie
      □ il n'affiche rien
      \square il ne compile pas
      ☐ il risque d'afficher bonjour à la place de coucou
10. Vous utilisez une boucle while quand:
      □ vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      □ vous n'avez pas déclaré de fonction
      ☐ l'incrément de la variable de boucle n'est pas 1
      □ vous avez déjà fait un for dans le même pro-
         gramme principal
```

```
11. On souhaite faire une boucle de contrôle de saisie : tant
                                                                  \square 0
                                                                                                                              \Box 8 6 4 2
   que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                 \Box 4
                                                                                                                              \square 8 2
   recommence la saisie de n. Soit le programme suivant :
                                                                  \square 3
                                                                                                                              \Box 02468
     int a = 0;
                                                                                                                              \Box 8 6 4 2 0
                                                            14. Sous unix (ou linux), la commande 1s permet de :
     int b = 20;
                                                                  \square afficher le contenu d'un fichier texte
                                                                                                                        18. Soit le programme principal suivant :
     int n:
     scanf("%d", &n);
                                                                  □ afficher la liste de fichiers contenus dans un
                                                                                                                            int main()
     while(cond)
                                                                     répertoire
                                                                  □ compiler un programme
                                                                                                                             int a = 3;
       scanf("%d", &n);
                                                                                                                             int b = 5;
                                                                  □ voir des clips musicaux
                                                                                                                             printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                            15. Les lignes
                                                                                                                             return EXIT_SUCCESS;
   Quelle est la condition cond :
                                                                int i;
      □ a<=n<=b
                                                                int x=0;
                                                                                                                            appelant la fonction f ainsi définie :
     \square (a<=n) && (n<=b)
                                                                for(i=0,i<5,i=i+1)
     \square (a<n) || (n>b)
                                                                                                                            int f(int a, int b)
                                                                {
      \square (n<=a) && (n<=b)
                                                                  x=x+1;
                                                                                                                              a = a + b;
12. Le code suivant :
                                                                                                                              return a;
                                                                  □ comportent une erreur qui sera détectée au cours
     int i:
                                                                     de l'édition de lien
     for (i = 0; i < 7; i = i + 2)
                                                                                                                            L'affichage dans le main est le suivant :
                                                                  □ comportent une erreur qui sera détectée au cours
                                                                                                                              \Box f(a,b)=13, a=8, b=5
         printf("%d ", i);
                                                                     de l'analyse syntaxique
     }
                                                                                                                              \Box f(a,b)=8, a=3, b=5
                                                                  \square ne comportent aucune erreur
     printf("\n");
                                                                                                                              \Box f(3,5)=8, a=3, b=5
                                                                  □ comportent une erreur qui ne sera pas détectée
   affichera:
                                                                                                                              \Box f(a,b)=8, a=8, b=5
                                                            16. Sous unix (ou linux), pour créer un répertoire TP4
     \Box 01234567
                                                                                                                        19. Laquelle de ces écritures correspond à la déclaration
                                                                dans le répertoire courant on peut utiliser la com-
     \Box 0246
                                                                                                                            d'une variable de type caractère en langage C?
                                                                mande:
     \Box 02468
                                                                                                                              □ char c:
                                                                  □ mkdir TP4
     \Box 0 1 2 3 4 5 6
                                                                                                                              ☐ char "c";
                                                                  □ new TP4
13. Soit la fonction f définie par :
                                                                                                                              ☐ int char:
                                                                 ☐ kwrite TP4
   int f(int a)
                                                                                                                              □ char 'c';
                                                                  ☐ yppasswd
   ₹
                                                                                                                        20. Sur un ordinateur avec un seul processeur, habituelle-
                                                            17. Le code suivant :
      printf("a = \n", %d);
                                                                                                                            ment les processus sont exécutés :
      if (a > 0)
                                                                 int i:
                                                                 for (i = 8; i > 0; i = i - 2)
                                                                                                                              \square tour à tour, un petit peu à chaque fois
        return 3;
                                                                                                                              □ en parallèle, chacun dans un registre
                                                                     printf("%d ", i);
                                                                                                                              \square tous ensemble
      return 4;
                                                                 printf("\n");
                                                                                                                              □ chacun son tour, après que le processus précédent
                                                                                                                                 a terminé
                                                                affichera:
   Alors l'expression f(0) prendra la valeur :
```