Éléments d'informatique – contrôle continue

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

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. Le type des réels en C est :
     □ char
     □ real
     □ double
     \square int
3. Vous utilisez une boucle while quand:
    \Box 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
     □ vous avez déjà fait un for dans le même pro-
       gramme principal
4. 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(int n);
    □ void saisie_utilisateur(char c);
     □ saisie_utilisateur(scanf(%d));
```

```
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 :
    \Box 6
    \square 16
    \square 20
    \square 3
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
    \Box (a.n == b.n) && (a.x == b.x)
    \square a\{n, x\} == b\{n, x\}
    \square a == b
7. 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);
    ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
8. 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?
       ***** *** ***
        ** *** ****
          **** **** ****
 9. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 5
     \square 101
     \Box 4
     \square 3
10. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
    affichera:
     □ rien
     ☐ Mineur
      □ Majeur
     □ Mineur
        Majeur
11. 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
     \square tous ensemble
      □ chacun son tour, après que le processus précédent
        a terminé
```

12. Pour afficher à l'aide de printf("%d\n",tab[i]); le contenu d'un tableau de 5 entiers initialisé au	□ 1.5	☐ dans lequel ces fonctions sont appelées dans le main
préalable, on utilise plutôt :	□ 1 □ 0	un ordre quelconque
☐ for(i=0;i<=5;i=i+1)	16. Sur unix (ou linux), la commande mkdir permet de :	□ alphabétique
☐ for(i=1;i<=5;i=i+1)	□ créer un répertoire	19. Pour déclarer une variable qui sera utilisée comme va-
☐ for(i=0;i<5;i=i+1)	□ créer un fichier texte	riable de boucle on peut utiliser l'instruction
☐ for(i=1;i<5;i=i+1)	□ ouvrir un fichier texte	☐ loop i;
13. Sous unix (ou linux), pour créer un répertoire TP4	\Box changer de répertoire courant	☐ int %d;
dans le répertoire courant on peut utiliser la com- mande :	17. Quel est le problème d'un programme comportant les	☐ int k;
□ mkdir TP4	lignes suivantes?	\Box int loop n;
\square yppasswd	while (1)	20. Le code suivant :
□ new TP4	<pre>t printf("coucou\n");</pre>	int i;
\square kwrite TP4	}	for (i = 1; i < 5; i = i + 1)
14. Laquelle de ces écritures correspond à la déclaration	$\hfill\Box$ il risque d'afficher bonjour à la place de coucou	{ printf("%d ", i);
d'une variable de type caractère en langage C?	$\hfill\Box$ il comporte une boucle infinie	}
□ char c;	\Box il ne compile pas	<pre>printf("\n");</pre>
☐ char 'c';	\square il n'affiche rien	affichera:
☐ int char; ☐ char "c";	18. Vous avez déclaré préalablement un ensemble de fonc-	$\Box \ 0 \ 1 \ 2 \ 3 \ 4$
, and the second	tions utilisées par votre programme principal. L'ordre	$\square 4 3 2 1$
15. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :	dans lequel vous devez maintenant définir ces fonctions est l'ordre :	$\Box \ 4\ 3\ 2\ 1\ 0$
□ 0.5	☐ dans lequel vous avez déclaré ces fonction	$\square \ 1 \ 2 \ 3 \ 4$

Éléments d'informatique – contrôle continue

Prénom:	Vom:
N° etu :	

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 et définie □ l'avoir définie ☐ l'avoir déclarée □ avoir défini une constante symbolique de la taille de cette fonction 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 : \square int n = carre(); \square n = carre(int n); \square n = carre(n); \square int carre(2): 3. Le code suivant : int age = 18; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); } affichera: □ Majeur ☐ Mineur Majeur □ Mineur □ 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^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. 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
    \Box 0 1 2 0 1 2
6. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
    □ ouvrir un fichier texte
    □ créer un fichier texte
    □ créer un répertoire
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. 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
    \Box C
     □В
     \square A
```

```
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(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
10. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
11. Le langage C est un langage
     □ lu, écrit, parlé
     □ compilé
     □ interprété
     □ composé
12. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     \square analyse harmonique
     □ analyse sémantique
     \square analyse lexicale
     □ analyse syntaxique
```

13. Laquelle de ces écritures correspond à la déclaration	Quelle est la condition cond :	18. Un fichier source est:
d'une variable de type caractère en langage C?	□ (n<=a) && (n<=b)	$\hfill\Box$ un document illisible pour les humains
☐ char c;	☐ a<=n<=b	$\hfill\Box$ un document qui doit être protégé
☐ char "c";	□ (a<=n) && (n<=b)	$\hfill \square$ un document de référence du système
☐ int char; ☐ char 'c';	□ (a <n) (n="" ="">b)</n)>	☐ un fichier texte qui sera traduit en instructions processeur
14. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire : □ int n = pgcd();	16. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	
☐ int pgcd(2);	\Box une directive préprocesseur $\verb"#include"$ manquante	19. Pour déclarer une fonction factorielle qui prend en
\square n = pgcd(n, 3);	☐ une fonction appelée avant sa déclaration	argument un entier et renvoie sa factorielle on écrit :
\square n = pgcd(int p, int q);	☐ un désaccord entre la déclaration et la définition	\Box int factorielle(int x);
15. On souhaite faire une boucle de contrôle de saisie : tant	d'une fonction	☐ struct int factorielle(int n);
que l'entier \mathbf{n} n'appartient pas à l'intervalle $[ab]$, on	$\hfill\Box$ une fonction déclarée mais non définie	☐ int factorielle(double n);
recommence la saisie de n. Soit le programme suivant :	17. Pour déclarer une fonction saisie_utilisateur qui	☐ int factorielle();
<pre>int a = 0; int b = 20; int n;</pre>	demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :	20. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
scanf("%d", &n);	☐ int saisie_utilisateur();	☐ #include <studlib.h></studlib.h>
while(cond)	☐ void saisie_utilisateur(int n);	☐ #appart <stdlib.h></stdlib.h>
<pre>{ scanf("%d", &n);</pre>	☐ saisie_utilisateur(scanf(%d));	☐ #include <stdio.h></stdio.h>
}	☐ void saisie_utilisateur(char c);	☐ #include <studio.h></studio.h>

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. 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 \Box 4 2. Le bus système sert à : □ transporter les processus du tourniquet au processeur ☐ Arriver à l'heure en cours □ Transférer des données et intructions entre processeur et mémoire ☐ Écrire des données sur le dique dur 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){ } $printf("j = %d\n", j);$ } qu'est ce qui sera affiché? \Box j = %d \Box j = 0

 \Box i = 5

 \Box j = 4

```
4. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
     □ 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
     \square tous ensemble
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)=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
6. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
    ☐ char 'c';
    \Box char c;
    ☐ char "c";
    ☐ int char:
7. Une variable booléenne est un variable :
    □ NaN (not a number, qui n'est pas un nombre)
    □ à laquelle une valeur vient d'être affectée
    □ qui est vraie ou fausse
     ☐ réelle positive
     □ jamais nulle
```

```
8. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      □ analyse sémantique
      \square analyse harmonique
      \square analyse syntaxique
      \square analyse lexicale
 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 n = carre();
      \square int carre(2);
      \square n = carre(int n);
      \square n = carre(n);
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}
      ☐ le programme affiche ****
      \square le programme affiche x
      □ la variable x vaut 16
11. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
    on chercher dans le programme?
      \square un point-virgule en trop
      ☐ un point-virgule manquant
      \square une accolade en trop
      □ une accolade manquante
```

```
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 :
      \square A
      ☐ ABCDEF
     \Box i
13. 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
     \square il n'affiche rien
     \square il ne compile pas
14. 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
```

```
15. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \Box 7
      □ 111
      \square 8
      \square 3
16. 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.n == b.n) \&\& (a.x == b.x)
      \Box a = b
17. Un enregistrement permet de grouper plusieurs valeurs
      \square ses champs
      \square ses cases
      \square ses chants
      \square ses blocs
18. Vous utilisez une boucle while quand:
      \Box 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
      \square vous avez déjà fait un for dans le même pro-
         gramme principal
```

```
19. 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?
     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);
   La valeur de somme affichée est :
     \Box 6
     \square 3
     \square 20
```

 \Box 16

т		-1
	100000	- 1
	acence	

Éléments d'informatique – contrôle continue

	Prénom √o etu	
		□ ne comportent aucune erreur
		□ comportent une erreur qui sera détectée au cours de l'édition de lien
ı,b);		 □ comportent une erreur qui ne sera pas détectée □ comportent une erreur qui sera détectée au cours de l'analyse syntaxique
	9.	Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit- on chercher dans le programme?
		\square un caractère interdit en C
		 □ une directive préprocesseur #include manquante □ une variable non déclarée
		\Box une faute de frappe dans un appel de fonction
	10.	Au début de la fonction main() on place le code :
		<pre>char b = 'A'; b = b + 2; printf("%c\n", b);</pre>
des		Alors l'affichage sera :
		□ A
		□ С
		□В
		□ b
	11.	L'ordonnancement par tourniquet permet :
pro-		\square de ne pas perdre de temps avec la commutation de contexte
pro-		\Box d'entretenir l'illusion que les processus tournent en parallèle
F		\Box de doubler la mémoire disponible
		$\hfill \Box$ d'afficher des ronds colorés à l'écran
	12.	L'écriture $\underline{101}$ en binaire correspond au nombre naturel :
		□ 101

 \square 5

 \Box 4

 \square 3

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
     □ compilé
     □ composé
     □ interprété
2. 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 1 2 1 2 3
     \Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1 0 1
3. Vous utilisez une boucle while quand :
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     \squarevous avez déjà fait un for dans le même pro-
        gramme principal
     □ vous n'avez pas déclaré de fonction
     \Box l'incrément de la variable de boucle n'est pas 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:
     ☐ int saisie_utilisateur();
```

□ void saisie_utilisateur(int n);

□ void saisie_utilisateur(char c);

□ saisie utilisateur(scanf(%d)):

```
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
   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
6. Laquelle des analyses suivantes ne fait pas partie
  étapes de la compilation :
    \square analyse syntaxique
    \square analyse lexicale
    □ analyse sémantique
     \square analyse harmonique
7. Le bus système sert à :
     □ Transférer des données et intructions entre
       cesseur et mémoire
     ☐ Écrire des données sur le dique dur
    □ transporter les processus du tourniquet au
        cesseur
     ☐ Arriver à l'heure en cours
8. Les lignes
  int i;
  int x=0;
  for(i=0,i<5,i=i+1)
  {
     x=x+1;
```

```
18. Le type des réels en C est :
13. Soit la fonction g définie par :
                                                            15. Au début de la fonction main() on place le code :
    int g(int a)
                                                                 char i;
                                                                                                                               \square double
                                                                 for (i = 'A'; i \le 'F'; i = i + 1)
                                                                                                                               \square char
      printf("a = \n", %d);
                                                                                                                               \square int
                                                                   printf("%c", i);
      if (1 > 0)
                                                                                                                               \square real
                                                                 printf("\n");
        return 5;
                                                                                                                         19. Soit la fonction f définie par :
                                                                Alors l'affichage sera:
      return 7;
                                                                  \square ABCDEF
                                                                                                                             int f(int a)
                                                                  \Box i
    Alors l'expression g(0) prendra la valeur :
                                                                                                                               printf("a = \n", %d);
                                                                  □ A
                                                                                                                               if (a > 0)
      \Box 0
                                                                  □ ccccc
      \Box 5
                                                            16. Pour déclarer une fonction factorielle qui prend en
                                                                                                                                 return f(a - 1) + 1;
      \Box 7
                                                                argument un entier et renvoie sa factorielle on écrit :
                                                                  ☐ struct int factorielle(int n);
                                                                                                                               return 4;
14. Soit un programme contenant les lignes suivantes :
                                                                  ☐ int factorielle(double n);
     int i = 0;
                                                                  ☐ int factorielle();
                                                                                                                             Alors l'expression f(1) prendra la valeur :
     int j = 0;
                                                                  \Box int factorielle(int x);
     for (i = 0; i < 3; i = i + 1)
                                                                                                                               \Box 5
                                                            17. Le code suivant :
                                                                                                                               \Box 0
         for (j = 0; j < 5; j = j + 1)
                                                                 int age = 20;
                                                                                                                               \Box 4
                                                                 if (age < 18)
                                                                                                                               \Box 1
                                                                 {
         }
                                                                      printf("Mineur\n");
                                                                                                                         20. Sur un ordinateur avec un seul processeur, habituelle-
     printf("j = %d\n", j);
                                                                                                                             ment les processus sont exécutés :
                                                                 printf("Majeur\n");
                                                                                                                               \square tous ensemble
                                                                affichera:
    qu'est ce qui sera affiché par ce printf?
                                                                  □ Majeur
                                                                                                                               \square chacun son tour, après que le processus précédent
     \Box j = 0
                                                                  ☐ Mineur
                                                                                                                                  a terminé
      \Box j = %d
                                                                  □ Mineur
                                                                                                                               □ tour à tour, un petit peu à chaque fois
     □ j = 4
                                                                     Majeur
                                                                                                                               \square en parallèle, chacun dans un registre
      \Box j = 5
                                                                  \square rien
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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
     \Box j = 4
     \Box i = 5
2. 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)
     □ A && B
     \square !(!A || B) == (A && !B)
3. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ créer un fichier texte
     □ créer un répertoire
     □ ouvrir un fichier texte
4. Un bit est:
     □ la longueur d'un mot mémoire
     ☐ l'instruction qui met fin à un programme
     \square un chiffre binaire (0 ou 1)
```

□ un battement d'horloge processeur

```
5. Un enregistrement permet de grouper plusieurs valeurs
  dans:
    \square ses blocs
    \square ses chants
    □ ses cases
    \square ses champs
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 0
    \square 8
    \square 16
    \Box 4
7. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
    ☐ int afficher_menu(int char);
    □ void afficher_menu();
    ☐ int afficher_menu();
    ☐ char afficher_menu(printf("menu"));
    ☐ double afficher_menu();
8. 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);
    □ void saisie_utilisateur(int n);
    ☐ int saisie_utilisateur();
    □ saisie_utilisateur(scanf(%d));
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=1;i<5;i=i+1)
    \square for(i=1;i<=5;i=i+1)
    ☐ for(i=0:i<=5:i=i+1)
```

```
10. Le code suivant :
     int age = 20;
    if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
    affichera:
     □ Mineur
        Majeur
     □ rien
     □ Majeur
     □ Mineur
11. 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';
12. 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
     \square mettre les blocs en séquence les uns à la suite des
        autres
     □ répéter un bloc tant qu'une condition est vérifée
13. Le type des réels en C est :
     □ char
     □ double
     ☐ real
     □ int
```

14. Le code suivant :	☐ printf("%d", factorielle(n));	\Box il risque d'afficher bonjour à la place de coucou
int age = 18;	16. Pour déclarer une procédure afficher_date qui prend	\square il ne compile pas
if (age < 18) {	en argument un struct date_s et affiche le contenu du struct, on écrit :	$\hfill\Box$ il comporte une boucle infinie
<pre>printf("Mineur\n");</pre>	\square void afficher_date(date_s d);	\square il n'affiche rien
}	\square int afficher_date(date_s d);	
else	☐ struct date_s afficher_date(struct date_s	d),19. Si racine est une fonction prenant en entrée un réel
<pre>1 printf("Majeur\n");</pre>	☐ void afficher_date(struct date_s d);	et renvoyant la racine carree de cet reel, et que x est
}	17. Un fichier source est:	une variable réelle définie et initialisée, il est incorrect d'écrire :
	□ un document qui doit être protégé	
affichera:	\square un document de référence du système	$\square x = racine(x * x) - racine(x);$
☐ Mineur	\square un fichier texte qui sera traduit en instructions	\square x = racine(2/3);
Majeur	processeur	\square x = racine(racine(x)*racine(x));
□ rien	☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur	\square x - 1 = racine(x);
☐ Mineur	un document illisible pour les humains	20. Le langage C est un langage
☐ Majeur 15. Si factorielle est une fonction prenant en entrée un	18. Quel est le problème d'un programme comportant les lignes suivantes?	□ lu, écrit, parlé
entier et renvoyant un entier, il est correct d'écrire :	while (1)	\square interprété
□ n = factorielle();	{	□ compilé
\square n = factorielle(p, q);	<pre>printf("coucou\n");</pre>	□ composé
☐ int factorielle(int 2);	}	□ compose

т	•	-1
	acence	

}

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
1. 004.	

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 blocs \square ses cases \square ses champs \square ses chants 2. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n"); printf("Majeur\n"); affichera: □ rien ☐ Mineur Majeur □ Majeur ☐ Mineur 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(racine(x)*racine(x)); \square x - 1 = racine(x); \square x = racine(2/3); \square x = racine(x * x) - racine(x); 4. Le code suivant : int age = 18; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n");

ique controle continue	LN
affichera:	
☐ Mineur	
☐ Majeur	
□ rien	
☐ Mineur Majeur	
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 :	
\Box dans lequel vous avez déclaré ces fonction	
\Box dans le quel ces fonctions sont appelées dans le main	
\square alphabétique	
\Box un ordre quel conque	
6. L'ordonnancement par tourniquet permet :	
$\hfill \square$ 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	
\Box d'afficher des ronds colorés à l'écran	
7. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :	
□ 8	
\square 3	
□ 111	
\Box 7	
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, int x);	
☐ int pgcd(int y, int x);	
□ void pgcd(int x, int y);	
\Box int pgcd(int x, y);	1

9.	Un programme en langage C doit comporter une et une seule définition de la fonction :
	□ include
	□ begin
	□ init
	□ main
LO.	Quel est le problème d'un programme comportant les lignes suivantes?
	while (1) {
	<pre>printf("coucou\n"); }</pre>
	□ il risque d'afficher bonjour à la place de coucou
	□ il n'affiche rien
	☐ il comporte une boucle infinie
	\Box il ne compile pas
l1.	Quels calculs peut-on programmer en programmation structurée ?
	□ il y 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
	□ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine
	\Box certains programmes sont de vrais plats de spaghetti
12.	Si le code :
	struct toto_s {
	int n;
	<pre>double x; };</pre>
	précède la fonction ${\tt main()},$ alors on peut écrire en début de ${\tt main()}$:
	\Box int toto.n = 3;

```
\square int struct toto_s = {3, -1e10};
     \square toto_s struct z = {3, 0.5};
     \square toto_s n, x;
     □ struct toto_s toto;
13. Le langage C est un langage
     □ composé
     □ lu, écrit, parlé
     □ compilé
     □ interprété
14. 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
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 = 4
      \Box j = 5
      \Box i = %d
      \Box j = 0
16. 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
      \square un désaccord entre la déclaration et la définition
         d'une fonction
      \square une directive préprocesseur \#include manquante
      □ une fonction déclarée mais non définie
17. 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);
      ☐ void exposant(double x^n);
      \square double exposant(double x, int n);
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(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
19. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
     \Box n = pgcd(int p, int q);
     \square n = pgcd(n, 3);
     \square int pgcd(2);
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
     \square 16
```

 \Box 0

 \Box 4

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	1,0111
n etu:	

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

- 1. 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 2. 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(n, 3); \square n = pgcd(int p, int q); 3. 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 □ une unité de calcul spécialisée de l'ordinateur \square un composant qui contient la liste des fichiers du système 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 4
\Box 1
\Box 0
```

```
5. 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 □ A

☐ ABCDEF

6. 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:
```

7. Un programme en langage C doit comporter une et une seule définition de la fonction :

```
□ begin
\square main
\square init
```

□ include

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

9. 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);
\square void pgcd(int x, int y);
\square int pgcd(int x, y);
```

```
10. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
```

```
\square n = mccarthy(p, q);
\square x = mccarthy(n);
\square int mccarthy(int 2);
\square n = mccarthy();
```

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, x} == b{n, x}
□ a == b
\Box a = b
\Box (a.n == b.n) && (a.x == b.x)
```

12. Un fichier source est:

a = a + b;

return a;

	un	$\operatorname{document}$	illis	ible	pour	les	humain
	un	document	qui	doit	être	pro	otégé

□ un document de référence du système

□ un fichie	que l'ont	doit	citer	dans	les	docume	nt
produits sur l'ordinateur							

□ un fichier texte qui sera traduit en instructions processeur

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)
```

```
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. L'ordonnancement par tourniquet permet :
     \Box de doubler la mémoire disponible
     □ d'afficher des ronds colorés à l'écran
     ☐ de ne pas perdre de temps avec la commutation
        de contexte
     □ d'entretenir l'illusion que les processus tournent
        en parallèle
15. 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;
     \Box toto_s struct z = {3, 0.5};
     □ struct toto_s toto;
     \Box int struct toto_s = {3, -1e10};
     \square int toto.n = 3;
```

```
16. 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 6
      \Box 0
17. Sous unix (ou linux), la commande 1s permet de :
      \square compiler un programme
     □ voir des clips musicaux
     \square afficher le contenu d'un fichier texte
     □ afficher la liste de fichiers contenus dans un
         répertoire
18. Le code suivant :
     int age = 18;
     if (age < 18)
     {
         printf("Mineur\n");
     }
```

els { }	ee printf("Majeur\n");
affich	nera:
	Mineur
	Mineur Majeur
	rien
	Majeur
19. Un e dans	nregistrement permet de grouper plusieurs valeurs :
	ses cases
	ses champs
	ses blocs
	ses chants
	déclarer une variable qui sera utilisée comme va- e de boucle on peut utiliser l'instruction
	int %d;
	int k;
	loop i;

 \square int loop n;

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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:
    \square 0 1 2 3
    \Box 01234
    \square 4 3 2 1
    \Box \ 4\ 3\ 2\ 1\ 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
     \square 8
     \Box 0
    \Box 4
3. 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);
    □ void saisie_utilisateur(int n);
    ☐ saisie_utilisateur(scanf(%d));
```

☐ int saisie_utilisateur();

```
4. 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-
     □ Transférer des données et intructions entre pro-
        cesseur et mémoire
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 :
     \square 8
     \square 0
     \Box 4
     \Box 16
6. 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 taille = 3
     \square #define N 3
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
     \square l'incrément de la variable de boucle n'est pas 1
8. 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
```

```
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(x * x) - racine(x):
      \square x = racine(racine(x)*racine(x)):
10. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \square 0.5
      \Box 1
      \Box 0
      \square 1.5
11. Quel est l'opérateur de différence en C :
      □ <>
      \Box !
      □!=
      \square \neq
12. Le langage C est un langage
      □ composé
      □ compilé
      □ lu, écrit, parlé
      □ interprété
13. 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 n, x;
      \square int struct toto_s = {3, -1e10};
      \square toto_s struct z = {3, 0.5};
```

□ struct toto_s toto;

14.	Un enregistrement permet de grouper plusieurs valeurs dans :	
	\square ses champs	
	□ ses cases	
	\square ses chants	
	\square ses blocs	
15.	Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :	
	\Box répéter un bloc tant qu'une condition est vérifée	
	\Box retourner un bloc	
	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition	
	$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres	
16.	Soit la fonction g définie par :	
	<pre>int g(int a) {</pre>	
	<pre>printf("a = \n", %d); if (1 > 0)</pre>	
	{	
	return 5;	
	}	

```
return 7;
    Alors l'expression g(0) prendra la valeur :
      \Box 5
      \Box 0
      \Box 7
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 :
      \square dans lequel ces fonctions sont appelées dans le
         main
      \square un ordre quelconque
      \square dans lequel vous avez déclaré ces fonction
      □ alphabétique
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
      □В
      \square A
      \Box C
```

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

□ int factorielle(double n);
□ int factorielle(int x);
□ int factorielle();
□ struct int factorielle(int n);

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

affichera :
□ 0
□ □</pre>
```

 \Box 6

 \square 42

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
n etu.	

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 : □ afficher le contenu d'un fichier texte \square afficher la liste de fichiers contenus dans un répertoire □ voir des clips musicaux □ compiler un programme 2. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme? □ une directive préprocesseur #include manquante □ une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration □ un désaccord entre la déclaration et la définition d'une fonction 3. 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 "Faux" □ le terminal affiche 5 \square le terminal affiche x = 2 \square le terminal affiche x = 5 4. 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";

```
5. 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 en trop
     ☐ un point-virgule manquant
      \square une accolade manquante
 6. 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
 7. 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 void exposant(double x^n);
     \square exposant(double x, int n, int r);
      \square double exposant(double x, int n);
8. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ ouvrir un fichier texte
     □ créer un répertoire
     □ créer un fichier texte
9. Un enregistrement permet de grouper plusieurs valeurs
    dans:
     □ ses chants
     \square ses blocs
     \square ses cases
     \square ses champs
10. Une variable booléenne est un variable :
     □ NaN (not a number, qui n'est pas un nombre)
     \square jamais nulle
     ☐ réelle positive
      □ qui est vraie ou fausse
```

□ à laquelle une valeur vient d'être affectée

```
11. Le code suivant :
    int age = 20;
    if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     □ Mineur
     □ Majeur
     □ Mineur
        Majeur
     □ rien
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);
     □ void saisie_utilisateur(int n);
     ☐ int saisie_utilisateur();
     ☐ saisie_utilisateur(scanf(%d));
13. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise:
     □ double afficher_menu();
     □ void afficher_menu();
     ☐ int afficher_menu(int char);
     ☐ int afficher_menu();
     ☐ char afficher_menu(printf("menu"));
14. 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);
     ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
```

```
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 4
      \square 3
     \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 x, y);
     \square void pgcd(int x, int y);
     \Box int pgcd(int x, int x);
     \Box int pgcd(int y, int x);
```

```
17. 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
18. 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 = %d
      \Box j = 5
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
    d'écrire :
      \square n = carre(int n);
      \square n = carre(n);
      \square int carre(2);
      \square int n = carre();
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 pgcd(2);
      \Box n = pgcd(int p, int q);
      \square int n = pgcd();
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
11 000.	

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

· r	réponse fausse. Durée : 20 minutes.
1.	L'ordonnancement par tournique t permet :
	\Box de doubler la mémoire disponible
	\Box d'afficher des ronds colorés à l'écran
	\Box d'entretenir l'illusion que les processus tournent en parallèle
	\Box de ne pas perdre de temps avec la commutation de contexte
2.	Sous unix (ou linux), la commande 1s permet de :
	\square compiler un programme
	\square voir des clips musicaux
	\Box afficher la liste de fichiers contenus dans un répertoire
	\Box afficher le contenu d'un fichier texte
3.	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	\square analyse harmonique
	\square analyse syntaxique
	\Box analyse lexicale
	\Box analyse sémantique
4.	Soit la fonction g définie par :
	<pre>int g(int a)</pre>
	<pre>{ printf("a = \n", %d); if (1 > 0) { return 5; }</pre>
	return 7;
	Alors l'expression g(0) prendra la valeur : □ 0 □ 7 □ 5

```
5. 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';
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?
    □ A && B
    \square (!A || B)
    ☐ (A == TRUE) && (B == TRUE)
    \square !(!A || B) == (A && !B)
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=3, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
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
     \Box 01234
     \square 4 3 2 1
     \square 0 1 2 3
 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. Si cet avertissement apparaît à la compilation :
   warning: implicit declaration of function 'max'
   , que doit-on chercher dans le programme?
     \Box\,une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une directive préprocesseur #include manquante
11. Le code suivant :
     int age = 18;
     if (age < 18)
         printf("Mineur\n");
    }
     else
    {
         printf("Majeur\n");
    }
   affichera:
     □ Mineur
        Majeur
     □ rien
     ☐ Mineur
     □ Majeur
```

```
12. Après exécution jusqu'à la ligne 15 du programme C :
                                                            15. Quelle étape de la compilation vient d'échouer lors-
                                                                qu'on a un message comme celui-ci :
 10
       int main() {
                                                                Undefined symbols : "_prinft" ou
 11
            int x = 5;
                                                                référence indéfinie vers « prinft »
 12
            int y;
                                                                  □ l'analyse sémantique
 13
                                                                  □ l'édition de liens
 14
            y = x;
 15
                                                                  □ l'analyse des entrées clavier
  16
                                                                  ☐ l'analyse harmonique
 17
      }
                                                            16. Le code suivant :
      ☐ le programme affiche "Faux"
                                                                 int i:
     \square la variable y vaut 5
                                                                 for (i = 0; i < 7; i = i + 2)
     \square la variable x vaut 0
                                                                      printf("%d ", i);
     \square la variable x vaut 5 et la variable y vaut 0
                                                                 printf("\n");
13. Quel est le problème d'un programme comportant les
                                                                affichera:
   lignes suivantes?
                                                                  \Box 02468
   while (1)
                                                                  \Box 0 1 2 3 4 5 6
                                                                  \Box 0 1 2 3 4 5 6 7
      printf("coucou\n");
                                                                  \Box 0246
                                                            17. Soit un programme contenant les lignes suivantes :
     □ il n'affiche rien
                                                                 int i = 0;
     \square il ne compile pas
                                                                 int j = 0;
     □ il risque d'afficher bonjour à la place de coucou
                                                                 for (i = 0; i < 3; i = i + 1)
     ☐ il comporte une boucle infinie
                                                                      for (j = 0; j < 5; j = j + 1)
14. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
                                                                      }
     \square n = mccarthy();
                                                                 }
                                                                 printf("j = %d\n", j);
      \square x = mccarthy(n);
     \square n = mccarthy(p, q);
                                                                qu'est ce qui sera affiché par ce printf?
      \square int mccarthy(int 2);
                                                                  \Box j = 5
```

```
\Box j = %d
     \Box i = 0
     \Box j = 4
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 1
      \Box 4
      \Box 0
      \Box 5
19. Dans la commande gcc, l'option -Wall signifie :
      ☐ que l'on veut voir tous les avertissements
      □ qu'il faut indenter le fichier source
      □ qu'il faut lancer un déboggueur
      \square qu'on veut changer alétoirement de fond d'écran
20. 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
      □В
```

□ b

т		-1
	100000	- 1
	acence	

Éléments d'informatique - contrôle continue

Prénom : Nom : N° etu :

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 :

```
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 (n<=a) && (n<=b)
     \square (a<=n) && (n<=b)
     \square (a<n) || (n>b)
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 0
    \square 7
    \Box 5
```

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

l'utilisateur, on utilise plutôt :		
\Box un débogueur		
\square printf("Valeur de n ? %g\n", n);		
☐ scanf("%d", &n);		
\square printf("Valeur de n ? %d\n", n);		

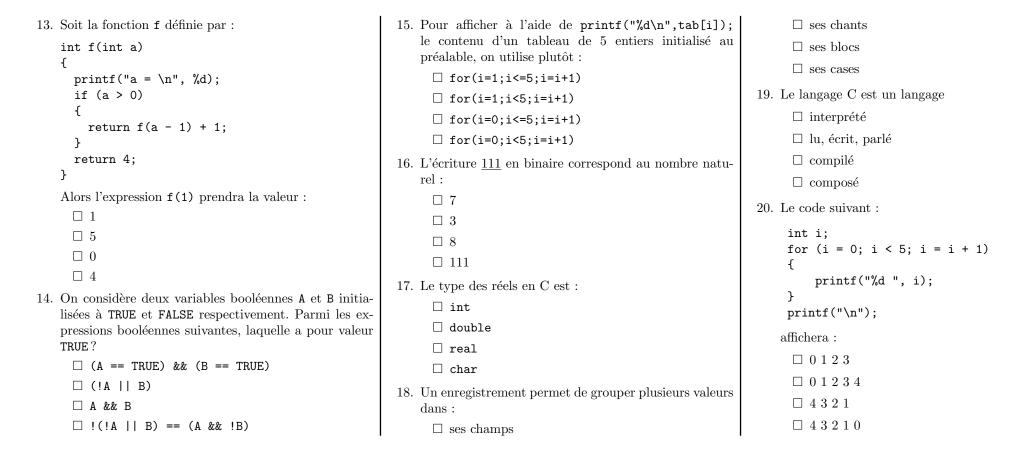
```
4. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
     ☐ que l'on veut voir tous les avertissements
     □ qu'on veut changer alétoirement de fond d'écran
5. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
     \square int n = pgcd();
     \square n = pgcd(n, 3);
     \square n = pgcd(int p, int q);
     \square int pgcd(2);
6. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 4
     \Box 5
     \square 101
     \square 3
7. Sur un ordinateur avec un seul processeur, habituelle-
   ment les processus sont exécutés :
     \square tous ensemble
     □ 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
8. Un bit est:
     □ la longueur d'un mot mémoire
     □ un battement d'horloge processeur
     \square un chiffre binaire (0 ou 1)
     ☐ l'instruction qui met fin à un programme
9. 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

```
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);
   La valeur de somme affichée est :
     \square 3
     \square 16
     \square 20
     \Box 6
11. 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
12. 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 0 1 2
     \Box 0 1 2 0 1 2 3
```

 \Box 0 1 2 3 0 1 2



Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

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 1.5
     \square 0.5
     \Box 1
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 1 2 3 0 1 2
     \Box 0 0 1 1 2 2 3
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. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \square 8
     \Box 7
     □ 111
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 :
     \Box 5
     \square 0
     \square 7
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:
     \Box i
     ☐ ABCDEF
     □ cccccc
     \square A
7. Le langage C est un langage
     □ lu, écrit, parlé
     □ composé
     □ interprété
     □ compilé
```

```
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 0 0 1 1 1
     \Box 1 2 1 2 3
     \Box 0 1 0 1 0 1 0 1
     \Box 0 1 2 0 1 2
 9. 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;
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 exposant(double x, int n, int r);
      \square void exposant(double x^n);
      \square double exposant(double x, int n);
      \square int exposant(double n, int x);
11. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square n = mccarthy(p, q);
      \square n = mccarthy();
      \square int mccarthy(int 2);
      \square x = mccarthy(n);
```

12. Une variable booléenne est un variable : □ à laquelle une valeur vient d'être affectée □ jamais nulle □ NaN (not a number, qui n'est pas un nombre) □ qui est vraie ou fausse	 16. 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;
☐ réelle positive	int n;
13. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : #appart <stdlib.h> #include <studlib.h> #include <studio.h> #include <stdio.h></stdio.h></studio.h></studlib.h></stdlib.h>	<pre>scanf("%d", &n); while(cond) { scanf("%d", &n); } Quelle est la condition cond:</pre>
14. Soit le programme principal suivant :	□ a<=n<=b
<pre>int main() {</pre>	☐ (a <n) (n="" ="">b)</n)>
int a = 3;	☐ (n<=a) && (n<=b)
<pre>int b = 5; printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b); return EXIT_SUCCESS;</pre>	17. Soit un programme contenant les lignes suivantes :
}	<pre>int i = 0; int j = 0;</pre>
appelant la fonction f ainsi définie :	for $(i = 0; i < 0; i = i + 1)$
<pre>int f(int a, int b) { a = a + b; return a; }</pre>	<pre>for (j = 0; j < 5; j = j + 1) { }</pre>
L'affichage dans le main est le suivant :	}
☐ f(a,b)=8, a=3, b=5	<pre>printf("j = %d\n", j);</pre>
☐ f(a,b)=13, a=8, b=5 ☐ f(a,b)=8, a=8, b=5 ☐ f(3,5)=8, a=3, b=5	}
15. Pour déclarer une fonction factorielle qui prend en	qu'est ce qui sera affiché?
argument un entier et renvoie sa factorielle on écrit :	□ j = 0
☐ int factorielle(double n);	□ j = 4
☐ struct int factorielle(int n);	□ j = %d
☐ int factorielle(int x);	□ j = 5
☐ int factorielle();	_ J ~

```
18. Après exécution jusqu'à la ligne 14 du programme C :
       int main() {
  10
  11
            int x = 5;
  12
            printf(" x = %d\n", 2);
  13
  14
  15
      }
  16
      \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
      \square le terminal affiche 5
      \square le terminal affiche x = 5
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=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. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     }
     else
         printf("Majeur\n");
     }
    affichera:
     \square rien
     □ Majeur
     \square Mineur
        Majeur
      ☐ Mineur
```

Éléments d'informatique – contrôle continue

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

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

```
1. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
    {
        printf("%d ", i);
    }
    printf("\n");
    affichera :
        □ 4 3 2 1 0
        □ 4 3 2 1
        □ 0 1 2 3 4</pre>
```

2. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.

```
☐ #define N = 3
☐ #define taille = 3
☐ #define N 3
☐ #define taille = N
```

 \square 1 2 3 4

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 :

```
□ 5
□ 7
□ 0
```

```
4. Après exécution jusqu'à la ligne 14 du programme C :

10 int main() {

11 int x = 5:
```

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

 \Box le terminal affiche x = 5

```
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 :

```
□ 1□ 4□ 5□ 0
```

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);</pre>
```

La valeur affichée est :

```
□ 16
□ 8
□ 0
```

 \Box 4

```
□ int mccarthy(int 2);
□ x = mccarthy(n);
□ n = mccarthy(p, q);
```

8. 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 = 4

\Box j = %d

\Box j = 0
```

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 }
```

 \Box le programme affiche "Faux"

 \square la variable x vaut 5 et la variable y vaut 3

```
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
      □ il comporte une boucle infinie
      □ il n'affiche rien
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:
      \square i
      \square A
      ☐ ABCDEF
      □ cccccc
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
      □ dans lequel ces fonctions sont appelées dans le
         main
     \square un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
13. Avant de faire appel à une fonction il est nécessaire
   de:
      □ l'avoir déclarée et définie
      □ l'avoir déclarée
      □ l'avoir définie
      \square avoir défini une constante symbolique de la taille
         de cette fonction
```

```
14. 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);
15. 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
16. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \Box 7
      □ 111
      \square 3
      \square 8
17. 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';
```

```
18. 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}
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 :
     \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");
     \Box printf("x=%d et y=%d\n",x y);
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)=8, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
```

Éléments d'informatique – contrôle continue

 \Box 4

Prénom: Nom:	
N° etu :	

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
     \square la variable x vaut 0
     \square la variable y vaut 5
     \square la variable x vaut 5 et la variable y vaut 0
     \square le programme affiche "Faux"
2. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \Box 7
     \square 3
     \square 8
3. 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 0 1 2 3 4 5 6 7
     \Box 02468
     \Box 0246
4. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
     □ qu'il faut lancer un déboggueur
     ☐ que l'on veut voir tous les avertissements
     \square qu'on veut changer alétoirement de fond d'écran
```

```
5. 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
    \Box l'incrément de la variable de boucle n'est pas 1
6. Le code suivant :
   int age = 20;
   if (age < 18)
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
  affichera:
    □ Mineur
       Majeur
    ☐ Mineur
    □ Majeur
    □ rien
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 0
    \square 3
```

```
8. Une variable booléenne est un variable :
      □ 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
      \square jamais nulle
 9. 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 variable non déclarée
      □ un caractère interdit en C
      \square une faute de frappe dans un appel de fonction
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:
     □ a == b
     \Box (a.n == b.n) && (a.x == b.x)
     \Box a = b
     \square a{n, x} == b{n, x}
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
      \Box (a<n) || (n>b)
      \square (n<=a) && (n<=b)
      \square (a<=n) && (n<=b)
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 ces fonctions sont appelées dans le
         main
      \square un ordre quelconque
      □ alphabétique
      □ dans lequel vous avez déclaré ces fonction
13. Un bit est:
      \square un chiffre binaire (0 ou 1)
      □ un battement d'horloge processeur
      □ la longueur d'un mot mémoire
      ☐ l'instruction qui met fin à un programme
14. 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 0 1 2 3
      \square 4 3 2 1
```

```
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?
      \square (A == TRUE) && (B == TRUE)
      □ A && B
     \square !(!A \mid | B) == (A \&\& !B)
      \square (!A || B)
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 0 1 2 0 1 2
     \Box 0 0 1 1 2 2
      \Box 1 2 3 1 2
17. 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>
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)=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
19. Pour déclarer une fonction factorielle qui prend en
    argument un entier et renvoie sa factorielle on écrit :
     ☐ int factorielle();
     □ struct int factorielle(int n);
     \Box int factorielle(int x);
     ☐ int factorielle(double n);
20. 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
     □В
```

□ b

int i = 0;

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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 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 i = 4
     \Box i = 5
     \Box i = %d
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
     \Box a = b
     \square (a.n == b.n) && (a.x == b.x)
     \Box a\{n, x\} == b\{n, x\}
3. 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 void exposant(double x^n);
     \square int exposant(double n, int x);
```

```
4. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     □ main
     \square init
     \square include
     □ begin
5. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses chants
     \square ses blocs
     \square ses champs
     □ ses cases
6. Le type des réels en C est :
     □ real
     □ char
     \square int
     ☐ double
7. 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
     □ une fonction déclarée mais non définie
     □ une directive préprocesseur #include manquante
     ☐ une fonction appelée avant sa déclaration
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 5
     \square 0
     \square 1
     \Box 4
```

```
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:
      □ un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
        main
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;
     \square toto_s n, x;
      \square int struct toto_s = {3, -1e10};
      \square toto_s struct z = {3, 0.5};
      □ struct toto_s toto;
11. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \square n = mccarthy();
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
     ☐ int mccarthy(int 2);
12. Sur un ordinateur avec un seul processeur, habituelle-
    ment les processus sont exécutés :
     \square tour à tour, un petit peu à chaque fois
      □ chacun son tour, après que le processus précédent
        a terminé
      \square tous ensemble
      □ en parallèle, chacun dans un registre
```

```
13. Si cette erreur apparaît à la compilation :
   Undefined symbols :"_prinft" ou
   référence indéfinie vers « prinft » que doit-
   on chercher dans le programme?
      \square un caractère interdit en C
     \square une faute de frappe dans un appel de fonction
     □ une directive préprocesseur #include manquante
     \square une variable non déclarée
14. Le code suivant :
     int i;
     for (i = 0; i < 7; i = i + 2)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 02468
     \Box 0246
     \Box 0 1 2 3 4 5 6 7
     \Box 0 1 2 3 4 5 6
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 :
      \square 7
      \Box 0
      \Box 5
16. 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 0
      \Box 16
      \Box 4
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 6
      \square 42
      \Box 1
```

 \Box 0

```
18. 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();
19. Un bit est:
     □ l'instruction qui met fin à un programme
      \square un chiffre binaire (0 ou 1)
     \square un battement d'horloge processeur
      □ la longueur d'un mot mémoire
20. 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

Éléments d'informatique – contrôle continue

Prénom: N	Vom:
N° etu :	

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 :
 - □ 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
 - □ la division du programme en zones homogènes échque
 - □ 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. 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 ccccc

 \square A

☐ ABCDEF

 \square i

3. 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 6

 \Box 10

□ 15

4. Le code suivant :

 $\Box 02468$

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 8 6 4 2
    □ 8 6 4 2 0
    □ 8 2
```

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
- $\Box\,$ une directive préprocesseur $\verb"#include"$ man quante
- $\Box\,$ une fonction déclarée mais non définie
- $\Box\,$ une fonction appelée avant sa déclaration
- 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 7

7. 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<=1)		
$(a \le n)$	&&	(n<=b)	
$(n \le a)$	&&	(n<=b)	
(a <n)< th=""><th>11 (</th><th>(n>b)</th><th></th></n)<>	11 ((n>b)	

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

l'avoir	défir	nie

l'avoir	déclarée	ρt	défini

avoir	défini	une	constante	symbolique	de	la	taill
de ce	tte for	ctio	n				

- □ l'avoir déclarée
- 9. Vous utilisez une boucle while quand:

Ш	l'incr	emen	t de I	a vai	riab.	le de	boucl	e n	est pa	s 1
	vous	avez	déjà	fait	un	for	dans	le	même	pro-

- □ vous n'avez pas déclaré de fonction
- □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance

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	$\mathrm{d}\mathrm{u}$	disque
--	-----	----------	------------------------	--------

gramme principal

	,	
Ш	des	processus

	en	temps	ď,	accès
--	----	-------	----	-------

□ certaines données de la mémoire de travail

```
11. Soit le programme principal suivant :
                                                            14. Le code suivant :
                                                                                                                             Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                             condition:
   int main()
                                                                 int i:
   {
                                                                 for (i = 4; i > 0; i = i - 1)
                                                                                                                               \square a = b
     int a = 3;
                                                                                                                               \square a\{n, x\} == b\{n, x\}
     int b = 5;
                                                                      printf("%d ", i);
                                                                                                                               □ a == b
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
                                                                 printf("\n");
                                                                                                                               \square (a.n == b.n) && (a.x == b.x)
                                                                affichera:
                                                                                                                         18. Un programme en langage C doit comporter une et une
   appelant la fonction f ainsi définie :
                                                                  \Box 01234
                                                                                                                             seule définition de la fonction :
   int f(int a, int b)
                                                                  \square 0 1 2 3
                                                                                                                               \square init
   {
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
                                                                                                                               \square include
      a = a + b;
                                                                  \square 4 3 2 1
      return a;
                                                                                                                               \square main
                                                            15. Le code suivant :
                                                                                                                               □ begin
   L'affichage dans le main est le suivant :
                                                                 int i;
                                                                                                                         19. Pour l'extrait de programme suivant :
      \Box f(a,b)=13, a=8, b=5
                                                                 for (i = 0; i < 5; i = i + 1)
     \Box f(a,b)=8, a=8, b=5
                                                                                                                                int produit = 1;
                                                                      printf("%d ", i);
     \Box f(3,5)=8, a=3, b=5
                                                                                                                                int serie[4] = \{2, 2, 2, 2\};
                                                                 }
                                                                                                                                for (i = 0; i < 4; i = i + 1)
      \Box f(a,b)=8, a=3, b=5
                                                                 printf("\n");
12. Quels calculs peut-on programmer en programmation
                                                                                                                                  produit = produit * serie[i];
                                                                affichera:
   structurée?
                                                                  \square 0 1 2 3
      □ il v a des calculs programmables en programma-
                                                                                                                                printf("produit = %d", produit);
        tion structurée qui ne sont pas programmables en
                                                                  \square 4 3 2 1
                                                                                                                             La valeur affichée est :
        langage machine
                                                                  \Box 01234
      □ en programmation structurée on peut program-
                                                                                                                               \Box 16
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
        mer tous les calculs programmables en langage
                                                                                                                               \Box 0
                                                            16. Si cette erreur apparaît à la compilation :
        machine
                                                                error: expected ';' before '}' token que doit-
                                                                                                                               \square 8
     \square certains programmes sont de vrais plats de spa-
                                                                on chercher dans le programme?
        ghetti
                                                                                                                               \Box 4
                                                                  □ un point-virgule en trop
      ☐ il v a des calculs programmables en langage ma-
                                                                                                                         20. Vous avez déclaré préalablement un ensemble de fonc-
                                                                  \Box une accolade manquante
        chine et qui ne sont pas programmables en pro-
                                                                                                                             tions utilisées par votre programme principal. L'ordre
        grammation structurée
                                                                  ☐ un point-virgule manquant
                                                                                                                             dans lequel vous devez maintenant définir ces fonctions
13. Pour déclarer une fonction pgcd qui calcule et renvoie
                                                                  \square une accolade en trop
                                                                                                                             est l'ordre:
   le plus grand diviseur commun de deux entiers positifs
                                                            17. Si a et b sont deux variables de type:
                                                                                                                               □ un ordre quelconque
   passés en arguments on écrit :
                                                                struct toto_s
                                                                                                                               □ alphabétique
      \square int pgcd(int x, int x);
     \Box int pgcd(int y, int x);
                                                                                                                               □ dans lequel vous avez déclaré ces fonction
                                                                   int n;
     \square void pgcd(int x, int y);
                                                                                                                               □ dans lequel ces fonctions sont appelées dans le
                                                                   double x;
      \square int pgcd(int x, y);
                                                                                                                                  main
                                                                };
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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

1. 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 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 : \square 3 \Box 0 \Box 4 3. Le langage C est un langage □ composé □ compilé □ interprété □ lu, écrit, parlé 4. Un registre du processeur est : □ une gamme de fréquence de fonctionnement du □ 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

```
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 = racine(racine(x)*racine(x)):
     \square x = racine(2/3):
     \square x - 1 = racine(x);
6. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \Box 1
     \Box 0
     \square 0.5
7. 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
     \square une fonction appelée avant sa déclaration
     □ une directive préprocesseur #include manquante
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
     \square la variable y vaut 5
     □ le programme affiche "Faux"
     □ la variable x vaut 3
```

```
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;
     \Box toto_s struct z = {3, 0.5};
10. Le type des réels en C est :
     □ int
      \square real
     □ double
     □ char
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, y);
     \Box int pgcd(int x, int x);
     \Box int pgcd(int y, int x);
     \square void pgcd(int x, int y);
12. Le code suivant :
     int age = 15;
     if (age < 18)
          printf("Mineur\n");
     }
     else
          printf("Majeur\n");
     }
    affichera:
      □ Majeur
```

□ rien
☐ Mineur
□ Mineur Majeur
13. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?
\Box une fonction déclarée mais non définie
\Box un désaccord entre la déclaration et la définition d'une fonction
\Box une fonction appelée avant sa déclaration
\Box une directive préprocesseur $\verb"#include"$ man quante
14. Le code suivant :
<pre>int i; for (i = 4; i > 0; i = i - 1) { printf("%d ", i);</pre>
} }
<pre>printf("\n");</pre>
affichera:
$\square \ 4\ 3\ 2\ 1$
$\Box \ 0\ 1\ 2\ 3$
$\Box \ 0\ 1\ 2\ 3\ 4$
$\Box \ 4\ 3\ 2\ 1\ 0$

```
17. Pour afficher à l'aide de printf("%d\n",tab[i]);
15. Soit un programme contenant les lignes suivantes :
                                                                  le contenu d'un tableau de 5 entiers initialisé au
     int i = 0;
                                                                 préalable, on utilise plutôt :
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
                                                                   \square for(i=1;i<5;i=i+1)
                                                                   \square for(i=1;i<=5;i=i+1)
         for (j = 0; j < 5; j = j + 1)
                                                                   \square for(i=0;i<5;i=i+1)
                                                                   \square for(i=0;i<=5;i=i+1)
          }
                                                              18. Un enregistrement permet de grouper plusieurs valeurs
                                                                  dans:
     printf("j = %d\n", j);
                                                                    \square ses blocs
    qu'est ce qui sera affiché par ce printf?
                                                                    □ ses cases
      \Box j = 5
                                                                    \square ses champs
      \Box j = %d
                                                                    \square ses chants
      \Box j = 4
                                                              19. Laquelle de ces écritures correspond à la déclaration
      \Box j = 0
                                                                  d'une variable de type caractère en langage C?
16. Les lignes
                                                                    \Box char c;
    int i;
    int x=0;
                                                                    ☐ char "c";
    for(i=0,i<5,i=i+1)
                                                                    □ char 'c';
    {
      x=x+1;
                                                                    \square int char;
                                                              20. Un programme en langage C doit comporter une et une
      \square comportent une erreur qui ne sera pas détectée
                                                                  seule définition de la fonction :
      □ comportent une erreur qui sera détectée au cours
                                                                    □ begin
         de l'édition de lien
                                                                    \square init
      □ comportent une erreur qui sera détectée au cours
         de l'analyse syntaxique
                                                                    \square include
      \square ne comportent aucune erreur
                                                                    \square main
```

т	•	-1
	acence	

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

• , 1

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();

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

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)
```

☐ int afficher_menu(int char);

3. 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

4. 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 x
     \square la variable x vaut -\frac{1}{2}
     □ la variable x vaut 16
     □ le programme affiche ****
5. 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
```

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

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(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):
```

8. L' mbre naturel

	_	_	•
écriture <u>111</u> en	binaire	correspond	au noi
l :			
□ 111			
\square 3			
□ 8			
□ 7			

9.	On I	egistre du processeur est :
		une unité de calcul spécialisée de l'ordinateur
		un composant qui contient la liste des fichiers du système
		une gamme de fréquence de fonctionnement du processeur
		una casa mámaira interna au processour qui sora

	in the case memore meetic at processeur	qui	SCIA
	manipulée directement lors des calculs		
10.	Un fichier source est:		

$\hfill\Box$ un fichier que l'ont doit citer dans les document produits sur l'ordinateur
\square un document de référence du système
Un fobian tauta qui gana traduit an instruction

Ш	un neme	er texte	qui	sera	traduit	en	Instructio)11
	processe	ur						
	un docu	ment a	i do	it ôtr	a protég	ó		

un	document	illisible	pour	les	humains

☐ Ecrire des données sur le dique dur
\Box Transférer des données et intructions entre pro
cesseur et mémoire

transporter	${\rm les}$	processus	$\mathrm{d} u$	tourniquet	au	pro
cesseur						

Arriver	à	l'heure	en	cours

12.	Le langage C est un langage
	□ composé

interprété
lu, écrit, parlé
compilé

11. Le bus système sert à :

1 3.	Un programme	en langage C doit	comporter	une et	un
	seule définition	de la fonction :			

	include
	begin
	main
П	init

14.	Un enregistrement permet de grouper plusieurs valeurs dans :	16. Si le code :
	□ ses chants	struct to {
	\square ses blocs	int n;
	□ ses cases	<pre>double };</pre>
	\square ses champs	précède la
15.	Pour l'extrait de programme suivant :	début de m
	<pre>int somme = 0;</pre>	☐ toto_
	int serie[4] = {2, 4, 10, 4};	☐ struc
	for (i = 0; i < 4; i = i + 1)	☐ toto_
	<pre>{ somme = somme + serie[i];</pre>	\Box int s
	}	☐ int t
	<pre>printf("somme = %d",somme);</pre>	17. Après la d
	La valeur de somme affichée est :	correct d'é
	\square 20	□ n = m
	\Box 6	\square int m
	\Box 3	\square n = m
	\Box 16	□ x = m
		I

```
16. Si le code :
    struct toto_s
    {
        int n;
        double x;
    };
    précède la fonction main(), alors on peut écrire en
    début de main() :
        □ toto_s n, x;
        □ struct toto_s toto;
        □ toto_s struct z = {3, 0.5};
        □ int struct toto_s = {3, -1e10};
        □ int toto.n = 3;

17. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
        □ n = mccarthy(p, q);
        □ int mccarthy(int 2);
        □ n = mccarthy();
        □ x = mccarthy(n);
```

18.	Si <i>n</i> est une variable entière pour demander sa valeur
	à l'utilisateur, on utilise plutôt :
	□ scanf("%d", &n);
	\Box un débogueur
	☐ printf("Valeur de n ? %d\n", n);
	☐ printf("Valeur de n ? %g\n", n);
19.	Pour compiler un programme prog.c, on utilise la ligne de commande :
	\square gcc prog.c -o -Wall prog.exe
	\square gcc -Wall prog.c -o prog.exe
	\square gcc -Wall prog.exe -o prog.c
	\square gcc prog.exe -Wall -o prog.c
20.	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 double exposant(double x, int n);
	\square void exposant(double x^n);
	\square int exposant(double n, int x);

Prénom: Nom: N° etu :

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 4 \square 3 \square 0 2. Le code suivant : int i; for (i = 1; i < 5; i = i + 1)printf("%d ", i); printf("\n"); affichera: $\Box 1234$ \square 4 3 2 1 $\Box 01234$ \Box 4 3 2 1 0 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 \square un désaccord entre la déclaration et la définition d'une fonction

une fonction déclarée mais non définie

☐ une directive préprocesseur #include manquante

```
4. 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
5. 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"));
6. 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
     ☐ 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
7. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %g\n", n);
    □ un débogueur
     □ printf("Valeur de n ? %d\n", n);
     □ scanf("%d", &n);
8. Avant de faire appel à une fonction il est nécessaire
   de:
     □ l'avoir définie
    □ l'avoir déclarée et définie
    □ l'avoir déclarée
    □ avoir défini une constante symbolique de la taille
        de cette fonction
```

```
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)
11. 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
12. 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 : $\Box 7$	15. Pour compiler un programme prog.c, on utilise la ligne de commande :	18. Une <i>segmentation fault</i> est une erreur qui survient lorsque:
	☐ gcc -Wall prog.exe -o prog.c	☐ la division du programme en zones homogènes échoue
\Box 5	\square gcc prog.exe -Wall -o prog.c	☐ le programme source a été enregistré sur le disque
13. Dans la commande gcc, l'option -Wall signifie :	☐ gcc prog.c -o -Wall prog.exe	dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur
$\hfill\Box$ qu'il faut indenter le fichier source	☐ gcc -Wall prog.c -o prog.exe	☐ le programme tente d'accèder à une partie de la
\Box qu'il faut lancer un déboggueur	16. Pour déclarer une procédure afficher_date qui prend	mémoire qui ne lui est pas réservée
\Box qu'on veut changer alétoirement de fond d'écran	en argument un struct date_s et affiche le contenu du struct, on écrit :	☐ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre
\square que l'on veut voir tous les avertissements	☐ struct date_s afficher_date(struct date_s	
14. Pour l'extrait de programme suivant :	\square void afficher_date(date_s d);	19. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :
<pre>int somme = 0; int serie[4] = {2, 4, 10, 4};</pre>	\square void afficher_date(struct date_s d);	n = mccarthy();
for $(i = 0; i < 4; i = i + 1)$	☐ int afficher_date(date_s d);	int mccarthy(int 2);
<pre>{ somme = somme + serie[i];</pre>	17. Un fichier source est:	$\square x = mccarthy(n);$
}	□ un document de référence du système	\square n = mccarthy(p, q);
<pre>printf("somme = %d",somme);</pre>	\square un document illisible pour les humains	20. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-
La valeur de somme affichée est :	\Box un fichier que l'ont doit citer dans les documents	on chercher dans le programme?
\square 3	produits sur l'ordinateur	un point-virgule manquant
\Box 6	□ un document qui doit être protégé	\square un point-virgule en trop
\square 20	\square un fichier texte qui sera traduit en instructions	$\hfill\Box$ une accolade man quante
□ 16	processeur	\Box une accolade en trop

т		-1
	acence	

Prénom:	Nom:
N° etu :	

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

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 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);
```

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 vous avez déclaré ces fonction
\square un ordre quelconque
```

□ alphabétique □ dans lequel ces fonctions sont appelées dans le main

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

```
□ struct int factorielle(int n);
\square int factorielle(int x):
☐ int factorielle();
☐ int factorielle(double n);
```

5. 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
    \square le terminal affiche x = 2
    □ le terminal affiche 5
    □ le terminal affiche "Faux"
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
    Πi
    □ A
```

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 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);
```

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 :

4
16
0
8

9. Un enregistrement permet de grouper plusieurs valeurs dans: \square ses champs

\square ses blocs	
\Box ses chants	
\square ses cases	

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

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

□ il n'affiche rien

Ш	il	comporte	une	bouc	le in:	finie

 \square il ne compile pas □ il risque d'afficher bonjour à la place de coucou

11. 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 y vaut 5

□ le programme affiche "Faux"

 \Box la variable x vaut 0

12. 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'ε	analyse	sémantique
-------	---------	------------

□ l'analyse des entrées clavier

□ l'analyse harmoniqu			l'anal	lyse	harmoniqu
-----------------------	--	--	--------	------	-----------

□ l'édition de liens

13. 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; 14. Le code suivant : int age = 20; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); } affichera : □ Mineur	□ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction 16. L'écriture 101 en binaire correspond au nombre naturel: □ 3 □ 4 □ 5 □ 101 17. 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 □ le terminal affiche 8 □ la case mémoire 8 contiendra 0 □ le bus explose □ la case mémoire 8 contiendra 16 18. Un bit est: □ un battement d'horloge processeur □ l'instruction qui met fin à un programme □ un chiffre binaire (0 ou 1)	<pre>19. Le langage C est un langage</pre>
	_ · · · · · · · · · · · · · · · · · · ·	□ (n<=a) && (n<=b) □ (a<=n) && (n<=b)

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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. Un enregistrement permet de grouper plusieurs valeurs dans:

- \square ses champs
- \square ses blocs
- □ ses chants
- □ ses cases

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

4. Le bus système sert à :

\Box Tra	nsférer	${\rm des}$	${\rm donn\acute{e}es}$	et	intructions	entre	pro-
ces	seur et	mén	oire				

- ☐ Écrire des données sur le dique dur
- ☐ Arriver à l'heure en cours
- □ transporter les processus du tourniquet au processeur

5. 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);
```

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

6. 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?

	**	**	**	**	**	**
	***	***	***	* *	***	***
	***	* * *	k***	k *>	***	***
П	**	***	k **	* **	***	* **

7. 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

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

ПВ

 \Box A

9.	Après la déclaration	:	int	mccarthy(int	n);,	il	est
	correct d'écrire :						

ш	п –	mccartny();
	x =	<pre>mccarthy(n);</pre>
	int	<pre>mccarthy(int 2);</pre>

 \square n = mccarthy(p, q);

10. Le langage C est un langage

 \Box = masses+br.().

⊔ compilé	
\Box lu, écrit, parlé	
\square composé	
□ interprété	

11. Quels calculs peut-on programmer en programmation structurée?

\Box 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 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

12. 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
```

 13. 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 des entrées clavier □ l'analyse harmonique □ l'analyse sémantique □ l'édition de liens 14. L'ordonnancement par tourniquet permet : 	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 directive préprocesseur #include manquante ☐ une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration 17. Quel est le problème d'un programme comportant les	 □ n = pgcd(int p, int q); □ n = pgcd(n, 3); □ int n = pgcd(); 19. Quel est l'opérateur de différence en C: □ ≠ □ != □ ! □ <>
 □ d'afficher des ronds colorés à l'écran □ de doubler la mémoire disponible □ de ne pas perdre de temps avec la commutation de contexte □ d'entretenir l'illusion que les processus tournent en parallèle 15. Si x est une variable réelle (de type double) alors 	lignes suivantes? while (1) { printf("coucou\n"); } □ il ne compile pas □ il risque d'afficher bonjour à la place de coucou	20. 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
x = 3/2 lui affecte la valeur : □ 0 □ 1 □ 1.5 □ 0.5	☐ il comporte une boucle infinie ☐ il n'affiche rien 18. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire : ☐ int pgcd(2);	 □ la case mémoire 8 contiendra 0 □ le bus explose □ le terminal affiche 8 □ la case mémoire 8 contiendra 16

Éléments d'informatique – contrôle continue

Prénom : N° etu :	Nom:
	\Box la variable x vaut 0
	\square la variable x vaut 5 et la variable y vaut 0

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 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
- 2. 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 programme principal
 - ☐ l'incrément de la variable de boucle n'est pas 1
 - □ vous n'avez pas déclaré de fonction
- 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)=13, a=8, b=5
- \Box f(a,b)=8, a=3, b=5
- \Box f(a,b)=8, a=8, b=5

```
4. Le code suivant :
   int i:
   for (i = 8; i > 0; i = i - 2)
        printf("%d ", i);
   }
   printf("\n");
  affichera:
    \square 8 2
```

- $\Box 02468$
- \Box 8 6 4 2 0 \Box 8 6 4 2
- 5. 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
- \square 4 3 2 1
- $\Box 01234$
- \Box 4 3 2 1 0
- 6. Une variable booléenne est un variable :
 - □ qui est vraie ou fausse
 - □ à laquelle une valeur vient d'être affectée
 - □ NaN (not a number, qui n'est pas un nombre)
 - □ jamais nulle
 - ☐ réelle positive
- 7. Après exécution jusqu'à la ligne 15 du programme C:

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

- \square la variable y vaut 5
- □ le programme affiche "Faux"
- 8. 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
- ☐ une directive préprocesseur #include manquante
- \square une fonction déclarée mais non définie
- □ un désaccord entre la déclaration et la définition d'une fonction
- 9. 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);
 - □ un débogueur
 - □ scanf("%d", &n);
- 10. 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
- 11. 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
- 12. Un enregistrement permet de grouper plusieurs valeurs dans:
 - \square ses blocs
 - \square ses cases
 - \square ses champs
 - \square ses chants

qu'on a un message comme celui-ci: Undefined symbols:"_prinft" ou référence indéfinie vers « prinft »	 16. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C? □ char "c"; □ char 'c'; □ int char; 17. 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); 18. 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(); □ printf("%d", factorielle(n)); □ n = factorielle(p, q); 	19. L'écriture 101 en binaire correspond au nombre naturel : □ 3 □ 5 □ 101 □ 4 20. Le code suivant : int i; for (i = 1; i < 5; i = i + 1) { printf("%d ", i); } printf("\n"); affichera : □ 4 3 2 1 0 □ 1 2 3 4 □ 0 1 2 3 4 □ 4 3 2 1
---	---	--

Éléments d'informatique – contrôle continue

Prénom:	Nom:
	NOIII .
N° etu :	
11 004.	

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: □ 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 □ 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 2. 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 \square il ne compile pas □ il n'affiche rien 3. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction ☐ int toto[taille=5]; \square int toto[5]; □ char tableau[5]; \square int[] new tableau(5); \square int tab[] = 5; 4. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme? ☐ un point-virgule manquant □ un point-virgule en trop \square une accolade en trop

□ une accolade manquante

5. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : ☐ struct int factorielle(int n); \Box int factorielle(int x): ☐ int factorielle(); ☐ int factorielle(double n): 6. Le code suivant : int age = 20; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); } affichera: □ rien □ Mineur Majeur □ Mineur □ Majeur 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 = b $\square (a.n == b.n) \&\& (a.x == b.x)$ □ a == b \square a{n, x} == b{n, x}

```
8. 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:
     □ (n<=a) && (n<=b)
     □ a<=n<=b
     \square (a<n) || (n>b)
      □ (a<=n) && (n<=b)
 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);
     \square n = factorielle();
     ☐ printf("%d", factorielle(n));
10. 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);
11. 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>
12. Le type des réels en C est :
     □ double
     □ char
      □ real
      □ int
```

```
13. Le code suivant :
     int age = 15;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     \square Mineur
        Majeur
     □ Majeur
     \square rien
     \square Mineur
14. 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 3
```

19
ГЭ
20

8.	Une variable booléenne est un variable :
	$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)
	\Box à laquelle une valeur vient d'être affectée
	\Box qui est vraie ou fausse
	\square réelle positive
	\square jamais nulle
9.	Vous utilisez une boucle while quand :
	\square vous n'avez pas déclaré de fonction
	$\hfill \square$ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
	\square vous avez déjà fait un for dans le même programme principal
	$\hfill \square$ l'incrément de la variable de boucle n'est pas 1
0.	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);
	\square void pgcd(int x, int y);

 \Box int pgcd(int y, int x);

Éléments d'informatique - contrôle continue

Prénom:	Nom:
N° etu:	

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: □ avoir défini une constante symbolique de la taille de cette fonction □ l'avoir déclarée et définie □ l'avoir déclarée □ l'avoir définie 2. Dans la commande gcc, l'option -Wall signifie : □ qu'il faut indenter le fichier source □ qu'il faut lancer un déboggueur ☐ que l'on veut voir tous les avertissements □ qu'on veut changer alétoirement de fond d'écran 3. Le langage C est un langage □ compilé □ lu, écrit, parlé □ interprété □ composé 4. 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 \Box un composant qui contient la liste des fichiers du système 5. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » \square l'édition de liens ☐ l'analyse des entrées clavier \square l'analyse harmonique □ l'analyse sémantique

6. Une variable booléenne est un variable :
☐ qui est vraie ou fausse
□ réelle positive
□ NaN (not a number, qui n'est pas un nombre)
□ à laquelle une valeur vient d'être affectée
☐ jamais nulle
7. L'écriture 111 en binaire correspond au nombre natu-
rel:
□ 8
\Box 7
\square 3
□ 111
 8. 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);
\Box n = pgcd(int p, int q);
\square int n = pgcd();
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 :
$\sqcup 4$
\Box 0
10. Si cette erreur apparaît à la compilation :
error: expected ';' before '}' token que doit- on chercher dans le programme?
un point-virgule manquant
□ une accolade en trop
□ un point-virgule en trop
□ une accolade manquante

```
11. 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
      □ le terminal affiche 8
      \square la case mémoire 8 contiendra 0
      \square le bus explose
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<=b
      □ (a<=n) && (n<=b)
      □ (n<=a) && (n<=b)
      □ (a<n) || (n>b)
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=%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");

14. 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);
☐ void saisie_utilisateur(int n);
☐ saisie_utilisateur(scanf(%d));
☐ int saisie_utilisateur();
15. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :
□ n = mccarthy();
\square n = mccarthy(p, q);
\square x = mccarthy(n);
☐ int mccarthy(int 2);
16. Soit la fonction g définie par :
<pre>int g(int a) {</pre>
<pre>printf("a = \n", %d);</pre>
if (1 > 0)
{ return 5;
}
return 7;
}

```
Alors l'expression g(0) prendra la valeur :
      \Box 5
     \Box 7
                                                                 □ char 'c';
      \Box 0
                                                                 \Box char c;
17. Si a et b sont deux variables de type:
                                                                 □ char "c";
    struct toto_s
                                                                 ☐ int char;
    {
      int n;
      double x;
                                                                 int i = 0;
    };
                                                                 int j = 0;
   Alors pour tester l'égalité de a et de b on utilise la
    condition:
     \square (a.n == b.n) && (a.x == b.x)
                                                                     for (j = 0; j < 2; j = j + 1)
     □ a == b
                                                                     {
                                                                          printf("%d ", i);
     \square a{n, x} == b{n, x}
                                                                     }
     \Box a = b
                                                                 }
18. On considère deux variables booléennes A et B initia-
                                                                 printf("\n");
    lisées à TRUE et FALSE respectivement. Parmi les ex-
   pressions booléennes suivantes, laquelle a pour valeur
                                                               qu'est ce qui sera affiché?
    TRUE?
                                                                 \Box 1 2 3 1 2
     □ A && B
                                                                 \Box 0 0 1 1 2 2
     \square !(!A \mid | B) == (A \&\& !B)
                                                                 \Box 0 1 2 0 1 2
     ☐ (A == TRUE) && (B == TRUE)
                                                                  \Box 0 1 0 1 0 1
     \square (!A || B)
```

```
19. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
20. Pour l'extrait de programme suivant :
     for (i = 0; i < 3; i = i + 1)
```

Éléments d'informatique – contrôle continue

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

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

```
1. 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
2. Au début de la fonction main() on place le code :
    char b = 'A';
   b = b + 2;
    printf("%c\n", b);
  Alors l'affichage sera:
    □В
    □ A
    □ b
    \Box C
3. 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
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 j = 4
    \Box j = %d
    \Box i = 5
     \Box j = 0
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 ces fonctions sont appelées dans le
        main
    □ un ordre quelconque
    \square dans lequel vous avez déclaré ces fonction
     □ alphabétique
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 appelée avant sa déclaration
    □ un désaccord entre la déclaration et la définition
        d'une fonction
    □ une fonction déclarée mais non définie
7. Le type des réels en C est :
     ☐ double
    \square real
    □ char
    \square int
8. 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
```

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
\Box certaines données de la mémoire de travail
\Box les fichiers du disque
\square des processus
Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :
\Box en parallèle, chacun dans un registre
$\hfill \Box$ chacun son tour, après que le processus précédent a terminé
\Box to us ensemble
\Box tour à tour, un petit peu à chaque fois
Le code suivant :
<pre>int age = 20; if (age < 18) {</pre>
<pre>printf("Mineur\n");</pre>
<pre>} printf("Majeur\n");</pre>
affichera:
□ Majeur
☐ Mineur
\Box rien
□ Mineur Majeur
Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :
☐ n = factorielle(p, q);
☐ printf("%d", factorielle(n));
☐ int factorielle(int 2);

 \square n = factorielle();

```
13. Si a et b sont deux variables de type :
                                                                                                                                      printf(" x = %d\n", 2);
                                                                 int f(int a, int b)
                                                                                                                            13
                                                                                                                            14
   struct toto s
                                                                   a = a + b;
                                                                                                                            15
                                                                                                                                 }
                                                                   return a;
                                                                                                                            16
      int n;
      double x;
                                                                                                                                \square le terminal affiche 5
                                                                 L'affichage dans le main est le suivant :
   };
                                                                                                                                □ le terminal affiche "Faux"
                                                                  \Box f(a,b)=8, a=8, b=5
   Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                                \square le terminal affiche x = 2
                                                                  \Box f(a,b)=8, a=3, b=5
   condition:
                                                                                                                                \square le terminal affiche x = 5
                                                                  \Box f(3,5)=8, a=3, b=5
     \Box (a.n == b.n) && (a.x == b.x)
                                                                   \Box f(a,b)=13, a=8, b=5
     □ a == b
                                                                                                                          19. Pour déclarer une fonction pgcd qui calcule et renvoie
                                                            16. Dans la commande gcc, l'option -Wall signifie :
                                                                                                                              le plus grand diviseur commun de deux entiers positifs
     \square a = b
                                                                                                                              passés en arguments on écrit :
                                                                  □ qu'il faut lancer un déboggueur
     \square a{n, x} == b{n, x}
                                                                                                                                \Box int pgcd(int x, int x);
                                                                  □ qu'il faut indenter le fichier source
14. Quel est le problème d'un programme comportant les
                                                                   ☐ que l'on veut voir tous les avertissements
                                                                                                                                \square void pgcd(int x, int y);
   lignes suivantes?
                                                                   \square qu'on veut changer alétoirement de fond d'écran
                                                                                                                                \Box int pgcd(int x, y);
   while (1)
                                                            17. Après exécution du programme :
                                                                                                                                \Box int pgcd(int y, int x);
   {
      printf("coucou\n");
                                                                    lecture 8 r0
                                                                                                                          20. Soit la fonction f définie par :
                                                                    valeur 3 r1
                                                                    mult r1 r0
                                                                                                                              int f(int a)
     □ il risque d'afficher bonjour à la place de coucou
                                                                    valeur 1 r2
     \square il comporte une boucle infinie
                                                                    add r2 r0
                                                                                                                                printf("a = \n", %d);
                                                                    ecriture r0 8
                                                                                                                                if (a > 0)
     □ il n'affiche rien
                                                                    stop
      \square il ne compile pas
                                                                                                                                   return f(a - 1) + 1;
15. Soit le programme principal suivant :
                                                                   \square le terminal affiche 8
                                                                                                                                return 4;
                                                                   □ la case mémoire 8 contiendra 0
   int main()
                                                                   \square le bus explose
                                                                                                                              Alors l'expression f(1) prendra la valeur :
     int a = 3:
                                                                   □ la case mémoire 8 contiendra 16
     int b = 5:
                                                                                                                                \Box 5
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
                                                             18. Après exécution jusqu'à la ligne 14 du programme C:
                                                                                                                                \Box 1
     return EXIT_SUCCESS;
                                                                    int main() {
                                                               10
                                                                                                                                \Box 0
                                                               11
                                                                         int x = 5;
                                                                                                                                \Box 4
   appelant la fonction f ainsi définie :
                                                               12
```

т.	-1
Licence	

Prénom:	Nom:
N° etu :	

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

int a = 0;

int b = 20;

2. 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 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>
```

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

```
printf("%d", somme);
   affichera:
     \Box 1
     \Box 0
     \Box 6
     \Box 42
5. 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
6. 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
7. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \Box 1
     \Box 0
     \square 1.5
8. Un registre du processeur est :
     \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
     □ une gamme de fréquence de fonctionnement du
        processeur
```

9.	Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :			
	☐ double afficher_menu();			
	☐ char afficher_menu(printf("menu"));			
	☐ int afficher_menu(int char);			
	☐ int afficher_menu();			
	☐ void afficher_menu();			
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 :			
	\Box un ordre quel conque			
	\Box dans lequel vous avez déclaré ces fonction			
	\Box dans lequel ces fonctions sont appelées dans le main			
	\square alphabétique			
11.	Vous utilisez une boucle while quand :			
	\square vous n'avez pas déclaré de fonction			
	\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			
	\Box l'incrément de la variable de boucle n'est pas 1			
12.	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			
	□ en programmation structurée on peut programmer tous les calculs programmables en langage machine			
	\Box certains programmes sont de vrais plats de spaghetti			
	☐ il y a des calculs programmables en langage ma-			

chine et qui ne sont pas programmables en pro-

grammation structurée

```
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
     \Box 4
     \square 3
14. Le code suivant :
    int age = 20;
    if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     □ Majeur
     \square Mineur
        Majeur
     \square rien
     ☐ Mineur
```

```
15. 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 6
     \Box 16
     \square 20
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 = 0
     □ j = 5
     \Box j = %d
     \Box j = 4
```

17. Le langage C est un langage
□ interprété
\square compilé
\Box lu, écrit, parlé
\square composé
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 :
\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);
19. 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
$\hfill \square$ certaines données de la mémoire de travail
\Box en temps d'accès
\square des processus
20. Si n 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);

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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 :
 - $\Box \ \ alphabétique$
 - $\Box\,$ un ordre quel conque
 - $\hfill \square$ dans lequel vous avez déclaré ces fonction
 - \square dans lequel ces fonctions sont appelées dans le main
- 2. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :
 - \square n = mccarthy();
 - \square n = mccarthy(p, q);
 - \square int mccarthy(int 2);
 - \square x = mccarthy(n);
- 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=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

- 4. Un bit est:
 - \square la longueur d'un mot mémoire
 - \Box l'instruction qui met fin à un programme
 - \square un chiffre binaire (0 ou 1)
 - \Box un battement d'horloge processeur
- 5. 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
 - \Box gcc -Wall prog.c -o prog.exe
 - \square gcc prog.c -o -Wall prog.exe
- 6. Le langage C est un langage
 - \square composé
 - \square interprété
 - $\Box\,$ lu, écrit, parlé
 - \square compilé
- 7. 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";
- 8. Dans la commande gcc, l'option -Wall signifie :
 - $\hfill \square$ que l'on veut voir tous les avertissements
 - $\hfill\Box$ qu'il faut indenter le fichier source
 - $\hfill \square$ qu'on veut changer alétoirement de fond d'écran
 - □ qu'il faut lancer un déboggueur
- 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)
    {
        ...
}</pre>
```

 $printf("j = %d\n", j);$

- qu'est ce qui sera affiché par ce printf?
 - \Box j = 4
 - $\Box j = 0$
 - $\Box j = %d$ $\Box j = 5$
- 10. Une variable booléenne est un variable :
 - □ qui est vraie ou fausse
 - $\Box\,$ à la quelle une valeur vient d'être affectée
 - $\hfill\Box$ NaN (not a number, qui n'est pas un nombre)
 - \Box jamais nulle
 - \square réelle positive
- 11. 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 6
- \Box 42
- \Box 0
- \Box 1
- 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 :

- □ a == b
- □ a = b
- \square a{n, x} == b{n, x}
- $\square (a.n == b.n) && (a.x == b.x)$

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?	
$\square !(!A \mid B) == (A && !B)$	
☐ (A == TRUE) && (B == TRUE)	
□ A && B	
\square (!A B)	
14. Pour déclarer un tableau d'entiers de taille 5, on peut	
utiliser l'instruction	
☐ int tab[] = 5;	
☐ int toto[taille=5];	
☐ int toto[5];	
☐ int[] new tableau(5);	
☐ char tableau[5];	
15. 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	
\square la case mémoire 8 contiendra 0	
\square le terminal affiche 8	
\Box la case mémoire 8 contiendra 16	
16. Si n est une variable entière pour demander sa valeur	
à l'utilisateur, on utilise plutôt :	
\square un débogueur	
□ scanf("%d", &n);	
☐ printf("Valeur de n ? %d\n", n);	
☐ printf("Valeur de n ? %g\n", n);	

```
17. Le code suivant :
     int age = 20;
    if (age < 18)
     {
         printf("Mineur\n");
     }
    printf("Majeur\n");
   affichera:
     □ Majeur
     \square Mineur
        Majeur
     \square rien
      \square Mineur
18. Le code suivant :
    int age = 20;
    if (age < 18)
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
      \square Mineur
     \square Mineur
        Majeur
     □ Majeur
      \square rien
```

```
19. 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)
     □ (a<=n) && (n<=b)
     ☐ (n<=a) && (n<=b)
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 0
```

 \Box 7

 \Box 5

Éléments d'informatique – contrôle continue

{

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

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 || B) == (A && !B)
☐ A && B
☐ (A == TRUE) && (B == TRUE)
```

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. Le langage C est un langage

```
□ composé
□ compilé
□ interprété
□ lu, écrit, parlé
```

4. 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";
```

5. 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)
    □ (n<=a) && (n<=b)
    □ a<=n<=b
    \square (a<=n) && (n<=b)
6. 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
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);
    ☐ int saisie_utilisateur();
    □ void saisie_utilisateur(int n);
8. 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
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 4
     \square 3
     \Box 0
10. Le code suivant :
    int age = 15:
    if (age < 18)
    {
         printf("Mineur\n");
    }
     else
     {
         printf("Majeur\n");
    }
   affichera:
     □ rien
     □ Majeur
     ☐ Mineur
        Majeur
     □ Mineur
11. 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};
     □ struct toto s toto:
     \Box toto_s struct z = {3, 0.5};
     \square int toto.n = 3;
     \square toto_s n, x;
```

12. Si cette erreur apparaît à la compilation :	☐ une fonction déclarée mais non définie	19. Le code suivant :
erreur: conflicting types for 'max', que doit- on chercher dans le programme?	☐ un désaccord entre la déclaration et la définition d'une fonction	<pre>int i; for (i = 0; i < 7; i = i + 2)</pre>
\Box une fonction appelée avant sa déclaration	\square une fonction appelée avant sa déclaration	{
 □ un désaccord entre la déclaration et la définition d'une fonction □ une fonction déclarée mais non définie 	16. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : □ #include <stdio.h></stdio.h>	<pre>printf("%d ", i); } printf("\n");</pre>
\Box une directive préprocesseur $\# include \ manquante$	☐ #include <studio.h></studio.h>	affichera:
13. L'écriture 101 en binaire correspond au nombre naturel : $\hfill 4$	☐ #appart <stdlib.h> ☐ #include <studlib.h></studlib.h></stdlib.h>	$\begin{array}{c} \square \ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7 \\ \square \ 0\ 2\ 4\ 6\ 8 \\ \square \ 0\ 2\ 4\ 6 \end{array}$
\square 3	17. Le code suivant :	\Box 0 1 2 3 4 5 6
\Box 5	int somme = 0;	20. Soit le programme principal suivant :
□ 101	int i; for (i = 1; i < 4; i = i + 1)	
14. Soit la fonction g définie par :	{	<pre>int main() f</pre>
<pre>int g(int a) { printf("a = \n", %d); if (1 > 0)</pre>	<pre>somme = somme + i; } printf("%d", somme);</pre>	<pre>int a = 3; int b = 5; printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b); return EXIT_SUCCESS;</pre>
{	affichera:	}
<pre>return 5; } return 7;</pre>	□ 1 □ 0	appelant la fonction f ainsi définie : int f(int a, int b)
}	\Box 42	{
Alors l'expression $g(0)$ prendra la valeur : \square 0 \square 5	☐ 6 18. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction	<pre>a = a + b; return a; }</pre>
□ 7	☐ int tab[] = 5;	L'affichage dans le main est le suivant :
15. Si cet avertissement apparaît à la compilation :	☐ int toto[taille=5];	\Box f(3,5)=8, a=3, b=5
warning: implicit declaration of function 'max	, \Box char tableau[5];	\Box f(a,b)=8, a=8, b=5
, que doit-on chercher dans le programme?	☐ int toto[5];	\Box f(a,b)=8, a=3, b=5
$\hfill \square$ une directive préprocesseur #include manquante	\square int[] new tableau(5);	\Box f(a,b)=13, a=8, b=5

Éléments d'informatique – contrôle continue

 $\begin{array}{ll} \text{Pr\'enom}: & \text{Nom}: \\ \text{N$^\circ$ etu}: & \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 <= 'F'; i = i + 1)
{
   printf("%c", i);
}
printf("\n");</pre>
```

Alors l'affichage sera:

- \square ccccc
- ☐ ABCDEF
- \square A
- \Box i

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);</pre>
```

La valeur de somme affichée est :

- \Box 0
- \Box 6
- \Box 15
- \Box 10

3. Vous utilisez une boucle while quand :

- \Box 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 vous n'avez pas déclaré de fonction
- □ l'incrément de la variable de boucle n'est pas 1

- 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 :
 - \square des processus
 - □ certaines données de la mémoire de travail
 - \square les fichiers du disque
 - \square en temps d'accès
- 5. Le bus système sert à :
 - ☐ Arriver à l'heure en cours
 - ☐ Écrire des données sur le dique dur
 - ☐ transporter les processus du tourniquet au processeur
 - ☐ Transférer des données et intructions entre processeur et mémoire
- 6. 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)
 - ☐ for(i=0;i<5;i=i+1)
 - ☐ for(i=1;i<=5;i=i+1)
- 7. Soit la fonction f définie par :
- int f(int a)

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

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

 \square 3

return 4;

- \Box 0
- \Box 4

- 8. Le type des réels en C est :
 - $\ \square \ \ \text{double}$
 - \square real
 - □ char
 - \square int
- 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);
 - $\hfill\Box$ void saisie_utilisateur(int n);
 - □ saisie_utilisateur(scanf(%d));
- 10. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
 - \square 3
 - □ 111
 - □ 8
 - \Box 7
- 11. Si cette erreur apparaît à la compilation :
 - error: expected ';' before '}' token que doiton chercher dans le programme?
 - □ un point-virgule en trop
 - \Box un point-virgule man quant
 - $\Box\,$ une accolade en trop
 - \Box une accolade manquante
- 12. 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);
}</pre>
```

qu'est ce qui sera affiché?

- \square 0 1 2 0 1 2 3
- □ 0 1 2 0 1 2

}

- □ 0 1 2 3 0 1 2
- □ 0 0 1 1 2 2 3

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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

```
1. Afin de représenter la taille d'un tableau, définir une
  constante symbolique N valant 3.
     \square #define N = 3
     ☐ #define N 3
     ☐ #define taille = N
     \square #define taille = 3
2. 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
    \Box 4 3 2 1 0
     \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 en trop
     ☐ un point-virgule manquant
     \square une accolade en trop
     □ une accolade manquante
4. Un bit est:
     □ la longueur d'un mot mémoire
    \Box un battement d'horloge processeur
    ☐ l'instruction qui met fin à un programme
     \square un chiffre binaire (0 ou 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:
     ☐ int saisie utilisateur():
```

□ void saisie_utilisateur(int n);

□ void saisie_utilisateur(char c);
□ saisie_utilisateur(scanf(%d));

```
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(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
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 0
    \Box 5
    \square 7
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 = 5
      \Box j = 4
      \Box j = 0
 9. 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
      \square une fonction appelée avant sa déclaration
      □ un désaccord entre la déclaration et la définition
         d'une fonction
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:
     \Box a = b
     □ a == b
     \square a{n, x} == b{n, x}
     \square (a.n == b.n) & (a.x == b.x)
11. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      \square 8
      \Box 7
      □ 111
      \square 3
```

12. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction	15. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre	<pre>□ exposant(double x, int n, int r);</pre> □ int exposant(double n, int x);
☐ int toto[5];	dans lequel vous devez maintenant définir ces fonctions est l'ordre :	☐ double exposant(double x, int n);
☐ int[] new tableau(5);	un ordre quelconque	19. Le code suivant :
☐ char tableau[5];	\Box dans lequel ces fonctions sont appelées dans le	
☐ int toto[taille=5];	main	int age = 18; if (age < 18)
☐ int tab[] = 5;	☐ dans lequel vous avez déclaré ces fonction☐ alphabétique	{
13. Pour l'extrait de programme suivant :	16. Pour l'extrait de programme suivant :	<pre>printf("Mineur\n"); } else</pre>
int i;	int i = 0;	{
int j;	int j = 0;	<pre>printf("Majeur\n");</pre>
for(i=4;i>0;i=i-1)	for (i = 0; i < 3; i = i + 1)	}
{ for(j=i;j<6;j=j+1) {	{ for (j = 0; j < 2; j = j + 1) {	affichera :
<pre>printf("*");</pre>	printf("%d ", i);	\Box rien
}	}	
<pre>printf(" ");</pre>	}	\square Mineur
,	<pre>printf("\n");</pre>	☐ Mineur
	qu'est ce qui sera affiché?	Majeur
qu'est ce qui sera affiché?	\Box 0 0 1 1 2 2	\square Majeur
_ **** *** ***	□ 0 1 2 0 1 2 -	20. Le code suivant :
<pre> ** ** ** ** **</pre>	□ 1 2 3 1 2	int i;
<pre> **** **** ****</pre>	\Box 0 1 0 1 0 1	for $(i = 0; i < 7; i = i + 2)$
☐ ** *** ****	17. Laquelle de ces écritures correspond à la déclaration	{
☐ ↑↑ ↑↑↑ ↑↑↑↑ ↑↑↑↑↑ 	d'une variable de type caractère en langage C?	<pre>printf("%d ", i);</pre>
14. Pour déclarer une fonction pgcd qui calcule et renvoie	☐ char 'c';	}
le plus grand diviseur commun de deux entiers positifs	☐ int char;	<pre>printf("\n");</pre>
passés en arguments on écrit :	☐ char c;	affichera:
☐ int pgcd(int x, y);	☐ char "c";	$\square \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6$
☐ void pgcd(int x, int y);	18. Pour déclarer une fonction exposant qui prend en ar-	$\Box \ 0\ 2\ 4\ 6\ 8$
☐ int pgcd(int y, int x);	gument un réel x et un entier positif n et renvoie la valeur de x^n on écrit :	$\square \ 0\ 2\ 4\ 6$
☐ int pgcd(int x, int x);	□ void exposant(double x^n);	$\square \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$

	ıce.	

Prénom :	Nom:	
N° etu :		

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. 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
3. Sous unix (ou linux), la commande cd permet de :
     □ changer de répertoire courant
     ☐ récupérer un programme arrêté avec la commande
       ab
```

```
\square détruire un fichier
    □ jouer de la musique
    □ ouvir un bureau partagé (common desktop)
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
     \square 1
    \Box 0
    \Box 4
5. Un bit est:
     \square un battement d'horloge processeur
     \square un chiffre binaire (0 ou 1)
    ☐ l'instruction qui met fin à un programme
    □ la longueur d'un mot mémoire
6. Sous unix (ou linux), la commande 1s permet de :
     □ compiler un programme
    \square afficher la liste de fichiers contenus dans un
        répertoire
    □ voir des clips musicaux
     □ afficher le contenu d'un fichier texte
7. Si n est une variable entière pour demander sa valeur
  à l'utilisateur, on utilise plutôt :
     □ printf("Valeur de n ? %g\n", n);
    □ un débogueur
     □ printf("Valeur de n ? %d\n", n);
```

□ scanf("%d", &n);

<u>:</u>
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 factorielle(int x);
☐ int factorielle(double n);
Le langage C est un langage
\Box interprété
\square composé
□ compilé
\Box lu, écrit, parlé
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";
Un enregistrement permet de grouper plusieurs valeurs dans :
\square ses chants
□ ses blocs
\square ses champs
□ ses cases
Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
☐ mkdir TP4
\square kwrite TP4
□ new TP4
\square yppasswd
Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
\square #include <studio.h></studio.h>
☐ #appart <stdlib.h></stdlib.h>
\square #include <stdio.h></stdio.h>

☐ #include <studlib.h>

 14. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur : □ 1 □ 0 □ 0.5 □ 1.5 15. 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 programme principal □ l'incrément de la variable de boucle n'est pas 1 □ vous n'avez pas déclaré de fonction 16. 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 harmonique ☐ l'édition de liens ☐ l'analyse des entrées clavier 17. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : ☐ analyse lexicale ☐ analyse harmonique ☐ analyse sémantique ☐ analyse syntaxique 18. 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	 19. 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(racine(x)*racine(x)); □ x - 1 = racine(x); □ x = racine(2/3); 20. 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);
--	---	--

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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: \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 □ 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 2. 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 prog.exe -Wall -o prog.c ☐ gcc -Wall prog.c -o prog.exe 3. L'ordonnancement par tourniquet permet : □ 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 ☐ d'entretenir l'illusion que les processus tournent en parallèle 4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : \square analyse harmonique □ analyse sémantique \square analyse syntaxique \square analyse lexicale 5. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3. \square #define N = 3 \square #define taille = N

 \square #define taille = 3

☐ #define N 3

```
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 y, int x);
     \square int pgcd(int x, y);
     \square int pgcd(int x, int x);
    \square void pgcd(int x, int y);
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
     \square 0
8. 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 la variable x vaut -\frac{1}{2}
     \square le programme affiche x
     □ la variable x vaut 16
     ☐ le programme affiche ****
```

```
9. 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)
      \square (n<=a) && (n<=b)

    a <= n <= b
</p>
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 des entrées clavier
      ☐ l'analyse sémantique
      □ l'édition de liens
      \square l'analyse harmonique
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=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)
12. 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?
          **** **** ****
13. 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;
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 5
     \Box 7
```

```
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 0
      \Box 4
      \Box 5
      \Box 1
16. 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 double exposant(double x, int n);
      \square exposant(double x, int n, int r);
      \square int exposant(double n, int x);
17. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \square 8
                                                                    \square un ordre quelconque
      \Box 7
                                                                    \Box dans lequel vous avez déclaré ces fonction
      \Box 111
                                                                    □ alphabétique
      \square 3
```

18. Avant de faire appel à une fonction il est nécessaire de :
□ l'avoir déclarée et définie
□ l'avoir définie
□ l'avoir déclarée
$\hfill \square$ avoir défini une constante symbolique de la taille de cette fonction
19. Au début de la fonction main() on place le code :
<pre>char i; for (i = 'A'; i <= 'F'; i = i + 1) { printf("%c", i); } printf("\n");</pre>
Alors l'affichage sera :
□ ABCDEF
□ A
□i
□ сссссс
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:
\Box dans lequel ces fonctions sont appelées dans le main

Prénom: Nom: N° etu :

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)
  \square (n<=a) && (n<=b)
  □ a<=n<=b
  \square (a<n) || (n>b)
le contenu d'un tableau de 5 entiers initialisé au
préalable, on utilise plutôt :
```

2. Pour afficher à l'aide de printf("%d\n",tab[i]);

 \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) ☐ for(i=1;i<=5;i=i+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. 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) \square !(!A || B) == (A && !B) □ A && B

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 n = carre(n); \square int carre(2);

 \square int n = carre();

6. Le code suivant :

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

affichera:

□ Mineur

☐ Mineur Majeur

□ Majeur

 \square rien

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

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

□ il n'affiche rien

☐ il comporte une boucle infinie

☐ il risque d'afficher bonjour à la place de coucou

 \square il ne compile pas

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 5 \Box 0

 \square 7

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 0 0 0 1 1 1

 \Box 1 2 1 2 3

10. Un bit est:

 \square 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

```
15. Vous avez déclaré préalablement un ensemble de fonc-
11. Le code suivant :
                                                                                                                               ☐ Mineur
                                                                tions utilisées par votre programme principal. L'ordre
                                                                                                                               ☐ Mineur
     int i;
                                                                dans lequel vous devez maintenant définir ces fonctions
                                                                                                                                  Majeur
     for (i = 4; i \ge 0; i = i - 1)
                                                                est l'ordre :
                                                                                                                         18. Au début de la fonction main() on place le code :
                                                                  \Box dans lequel vous avez déclaré ces fonction
         printf("%d ", i);
                                                                                                                              char b = 'A';
                                                                  □ un ordre quelconque
                                                                                                                              b = b + 2;
     printf("\n");
                                                                  □ alphabétique
                                                                                                                              printf("%c\n", b);
    affichera:
                                                                  □ dans lequel ces fonctions sont appelées dans le
                                                                                                                             Alors l'affichage sera:
      \square 4 3 2 1
                                                                     main
                                                                                                                               \square A
      \square 1 2 3 4
                                                            16. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                               \square B
      \Box 01234
                                                                    int main() {
                                                              10
                                                                                                                               \Box C
      \Box 4 3 2 1 0
                                                              11
                                                                         int x = 5;
                                                                                                                               □ b
                                                              12
                                                                         int y;
12. Pour déclarer une fonction factorielle qui prend en
                                                              13
    argument un entier et renvoie sa factorielle on écrit :
                                                                                                                         19. Au début de la fonction main() on place le code :
                                                              14
                                                                        y = x;
      □ struct int factorielle(int n);
                                                              15
      ☐ int factorielle(double n);
                                                                                                                              for (i = 'A'; i \le 'F'; i = i + 1)
                                                              16
                                                                         . . .
                                                              17
                                                                    }
      ☐ int factorielle(int x);
                                                                                                                                 printf("%c", i);
      ☐ int factorielle();
                                                                  \square la variable y vaut 5
13. Sous unix (ou linux), la commande 1s permet de :
                                                                  ☐ le programme affiche "Faux"
                                                                                                                              printf("\n");
      □ voir des clips musicaux
                                                                  \square la variable x vaut 0
                                                                                                                             Alors l'affichage sera:
      □ compiler un programme
                                                                  \square la variable x vaut 5 et la variable y vaut 0
                                                                                                                               ☐ ABCDEF
      □ afficher le contenu d'un fichier texte
                                                            17. Le code suivant :
                                                                                                                               \Box i
      \square afficher la liste de fichiers contenus dans un
                                                                                                                               □ cccccc
                                                                  int age = 20;
         répertoire
                                                                  if (age < 18)
                                                                                                                               □ A
14. Sous unix (ou linux), pour créer un répertoire TP4
                                                                  {
                                                                                                                         20. Un enregistrement permet de grouper plusieurs valeurs
    dans le répertoire courant on peut utiliser la com-
                                                                      printf("Mineur\n");
                                                                                                                             dans:
    mande:
                                                                                                                               \square ses champs
                                                                 printf("Majeur\n");
      ☐ yppasswd
                                                                                                                               \square ses blocs
      ☐ mkdir TP4
                                                                affichera:
                                                                                                                               □ ses chants
      ☐ kwrite TP4
                                                                  □ Majeur
      □ new TP4
                                                                                                                               \square ses cases
                                                                   \square rien
```

т.	-1
Licence	

Prénom :	Nom:	
N° etu :		

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 □ afficher le contenu d'un fichier texte □ voir des clips musicaux \square afficher la liste de fichiers contenus dans 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 \square dans lequel vous avez déclaré ces fonction □ alphabétique □ un ordre quelconque 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 □ il n'affiche rien □ il comporte une boucle infinie 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 1
    \square 0
    \Box 5
    \Box 4
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 -Wall prog.c -o prog.exe
    ☐ gcc prog.exe -Wall -o prog.c
6. Quels calculs peut-on programmer en programmation
  structurée?
    □ en programmation structurée on peut program-
       mer tous les calculs programmables en langage
       machine
    \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
    \square il y a des calculs programmables en langage ma-
       chine et qui ne sont pas programmables en pro-
       grammation structurée
7. Une variable booléenne est un variable :
    □ à laquelle une valeur vient d'être affectée
    ☐ qui est vraie ou fausse
    ☐ réelle positive
    □ NaN (not a number, qui n'est pas un nombre)
    ☐ jamais nulle
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 void exposant(double x^n);
    \square int exposant(double n, int x);
```

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

```
9. Un registre du processeur est :
      □ une unité de calcul spécialisée de l'ordinateur
     □ un composant qui contient la liste des fichiers du
         système
     \square une gamme de fréquence de fonctionnement du
     □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
10. 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 6 4 2
     \square 8 2
11. 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 3
      \square 20
12. L'ordonnancement par tourniquet permet :
      □ 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
```

□ de doubler la mémoire disponible

```
19. Soit un programme contenant les lignes suivantes :
13. Soit la fonction g définie par :
                                                                    int n;
                                                                    double x;
   int g(int a)
                                                                                                                                 int i = 0;
                                                                                                                                 int j = 0;
                                                                  Alors pour tester l'égalité de a et de b on utilise la
      printf("a = \n", %d);
                                                                                                                                 for (i = 0; i < 0; i = i + 1)
                                                                  condition:
      if (1 > 0)
                                                                    \square a{n, x} == b{n, x}
                                                                                                                                      for (j = 0; j < 5; j = j + 1)
        return 5;
                                                                   \square (a.n == b.n) \&\& (a.x == b.x)
                                                                    \Box a = b
      return 7;
                                                                                                                                      }
                                                                    □ a == b
    Alors l'expression g(0) prendra la valeur :
                                                              16. Si racine est une fonction prenant en entrée un réel
                                                                                                                                 printf("j = %d\n", j);
                                                                  et renvoyant la racine carrée de cet réel, et que x est
      \Box 0
                                                                 une variable réelle définie et initialisée, il est incorrect
                                                                                                                                 }
      \Box 7
                                                                  d'écrire :
      \Box 5
                                                                    \square x = racine(racine(x)*racine(x));
                                                                                                                               qu'est ce qui sera affiché?
14. Au début de la fonction main() on place le code :
                                                                   \Box x = racine(x * x) - racine(x);
                                                                    \square x - 1 = racine(x);
     char i:
                                                                                                                                 \Box j = 5
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                                    \square x = racine(2/3);
                                                                                                                                  \Box j = 0
                                                             17. Un enregistrement permet de grouper plusieurs valeurs
       printf("%c", i);
                                                                                                                                 \Box j = 4
                                                                  dans:
     printf("\n");
                                                                    \square ses champs
                                                                                                                                  \Box j = %d
                                                                    \square ses cases
    Alors l'affichage sera :
                                                                                                                            20. Un programme en langage C doit comporter une et une
                                                                    \square ses chants
      □ A
                                                                                                                                seule définition de la fonction :
                                                                    \square ses blocs
      ☐ ABCDEF
                                                                                                                                  \square main
                                                             18. Le langage C est un langage
      □ cccccc
                                                                    □ compilé
      \square i
                                                                                                                                  \square include
                                                                    □ lu, écrit, parlé
15. Si a et b sont deux variables de type :
                                                                                                                                  \square init
                                                                    □ composé
    struct toto_s
                                                                                                                                  □ begin
                                                                    □ interprété
    {
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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? \square 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 2. 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 des entrées clavier ☐ l'analyse harmonique □ l'édition de liens □ l'analyse sémantique 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 (n<=a) && (n<=b) \square (a<=n) && (n<=b) \square (a<n) || (n>b) □ a<=n<=b 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. 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 \mid | B) == (A \&\& !B)
     \square (!A | | B)
6. 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
7. Le code suivant :
    int i:
    for (i = 0; i < 5; i = i + 1)
    {
         printf("%d ", i);
    printf("\n");
   affichera:
     \square 4 3 2 1
     \Box 43210
     \square 0 1 2 3
     \Box 01234
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:
     \Box a = b
     \square (a.n == b.n) \&\& (a.x == b.x)
     \square a\{n, x\} == b\{n, x\}
```

□ a == b

000	•
9.	Une variable booléenne est un variable :
	\Box jamais nulle
	$\hfill\Box$ NaN (not a number, qui n'est pas un nombre)
	\Box à la quelle une valeur vient d'être affectée
	\Box qui est vraie ou fausse
	□ réelle positive
10.	Pour déclarer une procédure afficher_menu sans ar-
	gument et qui ne renvoie rien on utilise :
	<pre>□ int afficher_menu();</pre> <pre>□ void afficher_menu();</pre>
	_ · · · ·
	☐ int afficher_menu(int char);
	☐ char afficher_menu(printf("menu"));
11	☐ double afficher_menu();
11.	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 :
	\Box int exposant(double n, int x);
	□ exposant(double x, int n, int r);
	☐ void exposant(double x^n);
	\Box double exposant(double x, int n);
12.	Sur unix (ou linux), la commande mkdir permet de :
	\square créer un fichier texte
	\Box changer de répertoire courant
	\Box créer un répertoire
	\square ouvrir un fichier texte
13.	Le type des réels en C est :
	☐ double
	□ char
	\square int
	\square real
14.	Le bus système sert à :
	☐ Transférer des données et intructions entre processeur et mémoire
	\Box Écrire des données sur le dique dur
	☐ Arriver à l'heure en cours
	\Box transporter les processus du tourniquet au processeur

15. Soit la fonction g définie par :	
<pre>int g(int a) {</pre>	
$printf("a = \n", %d);$	
if (1 > 0)	
{	
return 5;	
}	
return 7;	
}	
Alors l'expression $g(0)$ prendra la valeur :	
\Box 5	
\Box 0	
□ 7	
16. 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 terminal affiche 8
\square le bus explose
\Box la case mémoire 8 contiendra 0
\Box la case mémoire 8 contiendra 16
17. L'ordonnancement par tourniquet permet :
☐ d'entretenir l'illusion que les processus tournent en parallèle
$\hfill\Box$ de doubler la mémoire disponible
□ d'afficher des ronds colorés à l'écran
\Box de ne pas perdre de temps avec la commutation de contexte
18. Un registre du processeur est :
$\hfill\Box$ une unité de calcul spécialisée de l'ordinateur

 \Box n = factorielle(p, q);

т		-4
	acence	

${\bf\acute{E}} l\acute{e}m\underline{ents}\ d'informatique-contr\^{o}le\ continue$

Prénom:	Nom:
N° etu :	

arème : 1 points par réponse juste (unique) ; -0.5 points ar réponse fausse. Durée : 20 minutes.		
1.	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	
	\Box certaines données de la mémoire de travail	
	\Box les fichiers du disque	
	\square des processus	
2.	Une variable booléenne est un variable :	
	□ jamais nulle	
	□ NaN (not a number, qui n'est pas un nombre)	
	\Box qui est vraie ou fausse	
	□ réelle positive	
	\Box à la quelle une valeur vient d'être affectée	
3.	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 :	
	□ A	
	□ сссссс	
	□i	
	□ ABCDEF	
4.	Un bit est:	
	\Box un battement d'horloge processeur	
	\square un chiffre binaire (0 ou 1)	
	□ la longueur d'un mot mémoire	
	\square l'instruction qui met fin à un programme	
5.	L'écriture <u>101</u> en binaire correspond au nombre natu-	
	rel:	
	□ 4 □ 5	
	⊔ 0	

6. Le type des réels en C est :			
□ char			
☐ double			
□ int			
\square real			
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 :			
$\square x = racine(racine(x)*racine(x));$			
\square x = racine(x * x) - racine(x);			
$\square x - 1 = racine(x);$			
\square x = racine(2/3);			
8. Après exécution jusqu'à la ligne 15 du programme \mathcal{C} :			
10 int main() {			
11 int x = 5;			
12 int y = 3;			
13 14			
14 x = y; 15			
16			
17 }			
\Box la variable y vaut 5			
\Box la variable x vaut 5 et la variable y vaut 3			
\square la variable x vaut 3			
\Box le programme affiche "Faux"			
9. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :			
\square x = mccarthy(n);			
☐ int mccarthy(int 2);			
\square n = mccarthy(p, q);			
\square n = mccarthy();			
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 :			
$\square for(i=1;i \le 5;i=i+1)$			
$\square for(i=0;i<=5;i=i+1)$			
☐ for(i=1;i<5;i=i+1)			

11. Si x est une variable réelle (de type double) alo $x = 3/2$ lui affecte la valeur :	rs
\Box 0	
□ 1	
\square 0.5	
□ 1.5	
12. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doi on chercher dans le programme?	t-
$\hfill\Box$ un désaccord entre la déclaration et la définition d'une fonction	n
\Box une directive préprocesseur #include manquan	te
\Box une fonction déclarée mais non définie	
\Box une fonction appelée avant sa déclaration	
13. Sous unix (ou linux), la commande 1s permet de :	
\Box voir des clips musicaux	
\Box compiler un programme	
\square afficher la liste de fichiers contenus dans u répertoire	ın
$\hfill\Box$ afficher le contenu d'un fichier texte	
14. Avant de faire appel à une fonction il est nécessair de :	re
☐ l'avoir définie	
☐ l'avoir déclarée	
$\hfill\Box$ l'avoir déclarée et définie	
$\hfill \square$ avoir défini une constante symbolique de la tail de cette fonction	le
15. Le langage C est un langage	
\square compilé	
\Box interprété	
$\hfill\Box$ lu, écrit, parlé	
\square composé	

le des
à 4 et btenu

```
return 3;
      return 4;
    Alors l'expression f(0) prendra la valeur :
     \square 3
     \Box 0
     \Box 4
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:
     \square a{n, x} == b{n, x}
     \square (a.n == b.n) && (a.x == b.x)
      \Box a = b
     □ a == b
```

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

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

т		-1
	acence	

int n;

Prénom:	Nom:
N° etu :	

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

r réponse fausse. Durée : 20 minutes.
1. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit- on chercher dans le programme?
\Box une accolade man quante
\Box un point-virgule en trop
\Box une accolade en trop
\Box un point-virgule man quant
2. Un enregistrement permet de grouper plusieurs valeurs dans :
\square ses chants
\square ses blocs
\square ses champs
□ ses cases
3. Un registre du processeur est :
\Box une case mémoire interne au processeur qui sera manipulée directement lors des calculs
\Box un composant qui contient la liste des fichiers du système
\Box une gamme de fréquence de fonctionnement du processeur
\square une unité de calcul spécialisée de l'ordinateur
4. 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>
\square le terminal affiche 8
\Box la case mémoire 8 contiendra 0
\square le bus explose

□ la case mémoire 8 contiendra 16

```
5. Un bit est:
     □ un battement d'horloge processeur
     \square un chiffre binaire (0 ou 1)
     ☐ l'instruction qui met fin à un programme
     □ la longueur d'un mot mémoire
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(2/3);
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
     \Box x = racine(x * x) - racine(x);
7. 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);
     ☐ void exposant(double x^n);
     \square double exposant(double x, int n);
     \square int exposant(double n, int x);
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 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
     □ la division du programme en zones homogènes
        échoue
9. 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;
```

```
scanf("%d", &n);
     while(cond)
       scanf("%d", &n);
   Quelle est la condition cond:
     \square (n<=a) && (n<=b)
     □ (a<=n) && (n<=b)
     □ a<=n<=b
     □ (a<n) || (n>b)
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(struct date_s d);
     □ void afficher_date(date_s d);
     ☐ int afficher_date(date_s d);
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
     □ 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
12. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     □ analyse sémantique
     \square analyse lexicale
     \square analyse harmonique
     □ analyse syntaxique
```

```
18. Si x est une variable réelle (de type double) alors
13. Soit le programme principal suivant :
                                                                   \Box 4
                                                                                                                               x = 3/2 lui affecte la valeur :
                                                                   \Box 0
   int main()
                                                                                                                                 \Box 0
   {
                                                                   \Box 1
     int a = 3;
                                                                   \Box 5
                                                                                                                                 \square 0.5
     int b = 5;
                                                             15. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                                 \square 1.5
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
                                                                     int main() {
                                                               10
                                                                                                                                 \Box 1
    return EXIT_SUCCESS;
                                                               11
                                                                         int x = 5;
                                                                                                                           19. Si cette erreur apparaît à la compilation :
                                                               12
                                                                         int y = 3;
   appelant la fonction f ainsi définie :
                                                                                                                               erreur: conflicting types for 'max', que doit-
                                                               13
                                                                                                                               on chercher dans le programme?
                                                               14
   int f(int a, int b)
                                                                         x = y;
                                                               15
                                                                                                                                 \square une fonction appelée avant sa déclaration
                                                               16
      a = a + b;
                                                                                                                                 ☐ une directive préprocesseur #include manquante
                                                               17
      return a;
                                                                                                                                 \Box un désaccord entre la déclaration et la définition
                                                                   □ la variable y vaut 5
                                                                                                                                    d'une fonction
   L'affichage dans le main est le suivant :
                                                                   \square la variable x vaut 3
                                                                                                                                 \square une fonction déclarée mais non définie
                                                                   \square la variable x vaut 5 et la variable y vaut 3
     \Box f(a,b)=8, a=8, b=5
                                                                                                                           20. Soit la fonction f définie par :
                                                                   □ le programme affiche "Faux"
     \Box f(a,b)=13, a=8, b=5
                                                                                                                               int f(int a)
     \Box f(3,5)=8, a=3, b=5
                                                             16. L'ordonnancement par tourniquet permet :
                                                                                                                               {
     \Box f(a,b)=8, a=3, b=5
                                                                   \square de doubler la mémoire disponible
                                                                                                                                 printf("a = \n", %d);
14. Soit la fonction f définie par :
                                                                   □ d'afficher des ronds colorés à l'écran
                                                                                                                                 if (a > 0)
                                                                                                                                 {
                                                                   \square de ne pas perdre de temps avec la commutation
   int f(int a)
                                                                      de contexte
                                                                                                                                    return 3;
                                                                   ☐ d'entretenir l'illusion que les processus tournent
      printf("a = \n", %d);
                                                                                                                                 return 4;
      if (a > 0)
                                                                      en parallèle
                                                             17. Le langage C est un langage
        return f(a - 1) + 1;
                                                                                                                               Alors l'expression f(0) prendra la valeur :
                                                                   □ compilé
      }
                                                                                                                                 \Box 0
                                                                   □ composé
      return 4;
                                                                                                                                 \square 3
                                                                   □ lu, écrit, parlé
                                                                   □ interprété
                                                                                                                                 \Box 4
   Alors l'expression f(1) prendra la valeur :
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

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

- Dans la commande gcc, l'option -Wall signifie :
 □ qu'on veut changer alétoirement de fond d'écran
 □ qu'il faut indenter le fichier source
 - $\Box\,$ que l'on veut voir tous les avertissements
 - \Box qu'il faut lancer un déboggueur
- 2. 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);
    }
}</pre>
```

qu'est ce qui sera affiché?

- □ 0 1 2 0 1 2
- $\square \ \, 0\ \, 1\ \, 0\ \, 1\ \, 0\ \, 1\ \, 0\ \, 1$
- \square 0 0 0 1 1 1
- \square 1 2 1 2 3
- 3. Le code suivant :

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

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

- 4. Un enregistrement permet de grouper plusieurs valeurs dans :
 - □ ses cases
 - \square ses chants
 - \Box ses champs
 - \square ses blocs
- 5. 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
 - $\Box\,$ une directive préprocesseur $\#\mbox{include}$ man quante
 - un désaccord entre la déclaration et la définition d'une fonction
 - $\Box\,$ une fonction appelée avant sa déclaration
- 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);</pre>
```

La valeur affichée est :

- \square 8
- \Box 0
- \Box 16
- \Box 4
- 7. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :
 - \square x = mccarthy(n);
 - \Box int mccarthy(int 2);
 - \square n = mccarthy();
 - \square n = mccarthy(p, q);

- 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
 - \Box le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée
 - \Box la division du programme en zones homogènes échoue
- 9. 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);
- 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 $\mathfrak{g}(0)$ prendra la valeur :

- \Box 5
- 11. 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 :	14. Le langage C est un langage	19. Le code suivant :
☐ ABCDEF	□ interprété	int 200 - 20:
□i	□ compilé	<pre>int age = 20; if (age < 18)</pre>
	□ lu, écrit, parlé	{
□ A	□ composé	<pre>printf("Mineur\n");</pre>
2. Vous avez déclaré préalablement un ensemble de fonc-	15. Pour déclarer une fonction exposant qui prend en ar-	}
tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions	gument un réel x et un entier positif n et renvoie la	<pre>printf("Majeur\n");</pre>
est l'ordre :	valeur de x^n on écrit :	affichera:
☐ dans lequel ces fonctions sont appelées dans le	\square exposant(double x, int n, int r);	
main	☐ double exposant(double x, int n);	\square Majeur
\square un ordre quelconque	☐ void exposant(double x^n);	☐ Mineur
\square dans lequel vous avez déclaré ces fonction	\Box int exposant(double n, int x);	☐ Mineur
□ alphabétique	16. Afin de représenter la taille d'un tableau, définir une	Majeur
3. Soit un programme contenant les lignes suivantes :	constante symbolique N valant 3.	□ rien
int i = 0;	☐ #define N = 3	□ Hen
int j = 0;	\square #define taille = N	20. Après exécution jusqu'à la ligne 15 du programme C :
for (i = 0; i < 0; i = i + 1)	☐ #define taille = 3	
{ for (j = 0; j < 5; j = j + 1)	□ #define N 3	10 int main() {
{	17. Laquelle de ces écritures correspond à la déclaration	11 int x = 5; 12 int y;
	d'une variable de type caractère en langage C?	12 int y; 13
}	☐ int char;	14 y = x;
}	□ char "c";	15
printf("j = %d\n", j);	□ char c;	16
···· }	☐ char 'c';	17 }
J	18. Pour déclarer une fonction factorielle qui prend en	_ 1
qu'est ce qui sera affiché?	argument un entier et renvoie sa factorielle on écrit :	□ le programme affiche "Faux"
$\Box j = 5$	☐ int factorielle(int x);	\square la variable y vaut 5
_ j = 0	☐ int factorielle(double n);	\Box la variable x vaut 0
□ j = %d	☐ struct int factorielle(int n);	☐ la variable x vaut 5 et la variable y vaut 0
□ j = 4	☐ int factorielle();	□ ia variable x vaut 5 et la variable y vaut 0

т		-1
	acence	

Prénom:	Nom:
N° etu :	

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 \ 4\ 3\ 2\ 1\ 0
     \square 4 3 2 1
     \square 1 2 3 4
     \Box 01234
2. Un enregistrement permet de grouper plusieurs valeurs
  dans:
     \square ses champs
     \square ses chants
     □ ses cases
     \square ses blocs
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);
    }
  qu'est ce qui sera affiché?
     \square j = %d
     \Box i = 5
     \Box j = 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);
    □ void saisie_utilisateur(int n);
    ☐ int saisie_utilisateur();
     □ saisie_utilisateur(scanf(%d));
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=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
6. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
    \square analyse lexicale
    \square analyse harmonique
    □ analyse sémantique
    \square analyse syntaxique
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:
    □ dans lequel vous avez déclaré ces fonction
```

 \square un ordre quelconque

```
□ dans lequel ces fonctions sont appelées dans le
        main
      □ alphabétique
 8. 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"));
 9. 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
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
      ☐ le programme affiche ****
     \square la variable x vaut -\frac{1}{2}
     \square la variable x vaut 16
      \square le programme affiche x
11. Vous utilisez une boucle while quand :
      □ vous n'avez pas déclaré de fonction
      ☐ 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 déià fait un for dans le même pro-
        gramme principal
```

12. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : #include <studio.h> #appart <stdlib.h> #include <studib.h> #include <stdio.h> 13. 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 == TRUE) && (B == TRUE) A && B 14. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : #include <studlib.h> #appart <stdlib.h> #include <stdio.h> #include <stdio.h> 15. Les lignes int i; int x=0; for(i=0,i<5,i=i+1) { x=x+1;</stdio.h></stdio.h></stdlib.h></studlib.h></stdio.h></studib.h></stdlib.h></studio.h>	<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 □ ne comportent aucune erreur 16. 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)); □ n = factorielle(); 17. Une variable booléenne est un variable : □ réelle positive □ NaN (not a number, qui n'est pas un nombre) □ jamais nulle □ à laquelle une valeur vient d'être affectée □ qui est vraie ou fausse 18. 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); } </pre>	□ 0 1 2 0 1 2 □ 0 0 1 1 2 2 3 □ 0 1 2 0 1 2 3 □ 0 1 2 3 0 1 2 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 : □ 4 □ 3 □ 0 20. Un fichier source est : □ 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 □ un document de référence du système
{ x=x+1; } □ comportent une erreur qui ne sera pas détectée		-

Prénom:	Nom:
N° etu :	
1. cca	

Licence 1 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: $\Box 0123$ \square 4 3 2 1 $\Box 01234$ \Box 4 3 2 1 0 2. L'écriture 111 en binaire correspond au nombre naturel: \Box 7 \square 3 \square 8 □ 111 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 3 \Box 16 \Box 6 \square 20 4. 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 exposant(double x, int n, int r); \square int exposant(double n, int x);

 \square void exposant(double x^n);

```
5. Le langage C est un langage
    □ compilé
    □ composé
    □ lu, écrit, parlé
     □ interprété
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)=13, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
7. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
    \square int char:
    ☐ char "c";
    \Box char c;
    □ char 'c';
8. 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
```

```
9. 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
      \square une fonction déclarée mais non définie
      \square une fonction appelée avant sa déclaration
10. 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
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 void pgcd(int x, int y);
     \square int pgcd(int y, int x);
      \Box int pgcd(int x, y);
      \square int pgcd(int x, int x);
12. 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 carre(2);
      \square int n = carre();
13. 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
```

14. Vous utilisez une boucle while quand:	16. Sous unix (ou linux), la commande 1s permet de :	affichera:
\square vous avez déjà fait un for dans le même pro-	□ afficher la liste de fichiers contenus dans un	□ rien
gramme principal	répertoire	\square Majeur
\Box l'incrément de la variable de boucle n'est pas 1	□ compiler un programme □ afficher le contenu d'un fichier texte	\square Mineur
\square vous n'avez pas déclaré de fonction	□ voir des clips musicaux	Majeur
\Box vous ne connaissez pas le nombre d'itérations de	17. Laquelle des analyses suivantes ne fait pas partie des	\square Mineur
la boucle à l'avance	étapes de la compilation :	19. Une variable booléenne est un variable :
15. Pour l'extrait de programme suivant :	\square analyse harmonique	\square réelle positive
<pre>int produit = 0;</pre>	\square analyse lexicale	\Box à la quelle une valeur vient d'être affectée
int serie[4] = {2, 2, 2, 2};	\square analyse sémantique	\square jamais nulle
for $(i = 0; i < 4; i = i + 1)$	□ analyse syntaxique	\Box NaN (not a number, qui n'est pas un nombre)
<pre>t produit = produit * serie[i];</pre>	18. Le code suivant :	\Box qui est vraie ou fausse
}	int age = 18; if (age < 18)	20. Un registre du processeur est :
<pre>printf("produit = %d", produit);</pre>	{	$\hfill\Box$ une case mémoire interne au processeur qui sera
La valeur affichée est :	<pre>printf("Mineur\n");</pre>	manipulée directement lors des calculs
\Box 0	}	\Box une gamme de fréquence de fonctionnement du
\Box 16	else {	processeur
□ 8	<pre>printf("Majeur\n");</pre>	☐ un composant qui contient la liste des fichiers du système
\Box 4	}	$\hfill \square$ une unité de calcul spécialisée de l'ordinateur

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

rı	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 :
	□ alphabétique
	\square un ordre quelconque
	\Box dans lequel vous avez déclaré ces fonction
	\Box dans lequel ces fonctions sont appelées dans le main
2.	Soit la fonction f définie par :
	<pre>int f(int a) {</pre>
	<pre>printf("a = \n", %d); if (a > 0) {</pre>
	return 3;
	}
	return 4;
	Alors l'expression f(0) prendra la valeur :
3.	Sous unix (ou linux), la commande cd permet de :
	□ changer de répertoire courant
	☐ détruire un fichier
	□ jouer de la musique
	□ récupérer un programme arrêté avec la commande ab
	\square ouvir un bureau partagé (common desktop)
4.	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;

```
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)</pre>
         printf("*");
      printf(" ");
   qu'est ce qui sera affiché?
     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 - 1 = racine(x);
    \square x = racine(racine(x)*racine(x));
    \square x = racine(2/3);
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
     \square ses champs
     \square ses cases
     \square ses chants
8. 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);
```

 \Box int pgcd(int x, int x);

 \square int pgcd(int x, y);

```
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 16
     \Box 0
     \Box 4
     \square 8
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. 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);
     \square n = factorielle();
     ☐ printf("%d", factorielle(n));
12. 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 0 1 2 3
      \Box 0 1 2 3 0 1 2
      \Box 0 1 2 0 1 2
13. Un fichier source est:
      \square 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
      □ un fichier texte qui sera traduit en instructions
      □ un document qui doit être protégé
14. Après exécution du programme :
       lecture 8 r0
       valeur 3 r1
       mult r1 r0
  4
       valeur 1 r2
       add r2 r0
       ecriture r0 8
       stop
       5
      □ la case mémoire 8 contiendra 0
      □ la case mémoire 8 contiendra 16
      □ le terminal affiche 8
      \square le bus explose
```

```
15. 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
     ☐ un point-virgule manquant
     \square une accolade en trop
16. 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>
17. 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 double exposant(double x, int n);
     \square int exposant(double n, int x);
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(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
19. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise:
     ☐ int afficher_menu(int char);
     ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu();
     □ void afficher_menu();
     ☐ double afficher_menu();
20. Le code suivant :
    int i;
    for (i = 4; i \ge 0; i = i - 1)
         printf("%d ", i);
    printf("\n");
    affichera:
     \Box 01234
     \square 1 2 3 4
     \square 4 3 2 1
```

 \Box 4 3 2 1 0

Éléments d'informatique – contrôle continue

}

Prénom:	Nom:	
N° etu :		

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

	cponse rausse. Durce: 20 mmutes.
1.	Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction
	☐ int k;
	□ loop i;
	☐ int loop n;
	☐ int %d;
2.	Le type des réels en C est :
	\square double
	□ char
	\square real
	\square int
3.	Le code suivant :
	int age = 18; if (age < 18)
	{
	<pre>printf("Mineur\n"); }</pre>
	else
	{
	<pre>printf("Majeur\n"); }</pre>
	affichera :
	□ Mineur Majeur
	□ rien
	□ Mineur
	□ Majeur
4.	Vous utilisez une boucle while quand :
	□ 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
	\Box l'incrément de la variable de boucle n'est pas 1

```
5. Le langage C est un langage
     □ compilé
     □ interprété
     □ lu, écrit, parlé
     □ composé
6. 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
7. 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
     \Box 1 2 3 4
     \square 4 3 2 1
8. 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
     \Box 0 1 2 0 1 2
 9. 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
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)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
11. 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
      \Box une unité de calcul spécialisée de l'ordinateur
      \square un composant qui contient la liste des fichiers du
```

système

12. Vous avez déclaré préalablement un ensemble de fonc-	Quelle est la condition cond :
tions utilisées par votre programme principal. L'ordre	□ (a<=n) && (n<=b)
dans lequel vous devez maintenant définir ces fonctions	□ (n<=a) && (n<=b)
est l'ordre :	☐ a<=n<=b
☐ dans lequel vous avez déclaré ces fonction	□ (a <n) (n="" ="">b)</n)>
\square alphabétique	16. Si n est une variable entière pour demander sa valeur
\square un ordre quelconque	à l'utilisateur, on utilise plutôt :
\square dans lequel ces fonctions sont appelées dans le	□ scanf("%d", &n);
main	☐ printf("Valeur de n ? %g\n", n);
13. Si cet avertissement apparaît à la compilation :	□ un débogueur
warning: implicit declaration of function 'max, que doit-on chercher dans le programme?	☐ printf("Valeur de n ? %d\n", n);
, que doit-on chercher dans le programme : ☐ une fonction appelée avant sa déclaration	17. Pour l'extrait de programme suivant :
☐ une fonction déclarée mais non définie	int somme = 0;
	int serie[4] = {2, 4, 10, 4};
une directive préprocesseur #include manquante	for $(i = 0; i < 4; i = i + 1)$
☐ un désaccord entre la déclaration et la définition d'une fonction	{
	somme = somme + serie[i];
14. Pour compiler un programme prog.c, on utilise la ligne de commande :	<pre>printf("somme = %d",somme);</pre>
☐ gcc prog.c -o -Wall prog.exe	La valeur de somme affichée est :
☐ gcc -Wall prog.c -o prog.exe	
☐ gcc wall plog.c o plog.exe ☐ gcc prog.exe -Wall -o prog.c	
☐ gcc -Wall prog.exe -o prog.c	
15. On souhaite faire une boucle de contrôle de saisie : tant que l'entier \mathbf{n} n'appartient pas à l'intervalle $[ab]$, on	
recommence la saisie de n. Soit le programme suivant :	18. Soit la fonction f définie par :
int a = 0;	<pre>int f(int a) {</pre>
int $b = 20$;	printf("a = \n", %d);
int n;	if (a > 0)
scanf("%d", &n);	{
while(cond)	return f(a - 1) + 1;
{	}
scanf("%d", &n); }	return 4; }
,	· ·

```
Alors l'expression f(1) prendra la valeur :
      \Box 1
      \Box 0
      \Box 4
      \Box 5
19. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
          printf("%d ", i);
     printf("\n");
    affichera :
      \square 0 1 2 3
      \square 4 3 2 1
      \Box \ 0\ 1\ 2\ 3\ 4
      \Box 4 3 2 1 0
20. 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
      }
      \Box la variable x vaut -\frac{1}{2}
      \Box le programme affiche x
      \square la variable x vaut 16
      \square le programme affiche ****
```

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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: $\Box 02468$ \Box 0 1 2 3 4 5 6 7 $\Box 0123456$ $\Box 0246$ 2. 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" □ la variable v vaut 5 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'analyse syntaxique □ ne comportent aucune erreur \square comportent une erreur qui sera détectée au cours de l'édition de lien □ comportent une erreur qui ne sera pas détectée

```
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 1
     \Box 5
     \Box 4
     \square 0
5. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \square 101
     \Box 4
     \square 5
     \square 3
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. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses blocs
     \square ses cases
     □ ses chants
```

```
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
     □ un point-virgule manquant
      \square une accolade en trop
 9. Une variable booléenne est un variable :
      \square jamais nulle
      □ à laquelle une valeur vient d'être affectée
      \square qui est vraie ou fausse
      ☐ réelle positive
     □ NaN (not a number, qui n'est pas un nombre)
10. Le bus système sert à :
     ☐ Arriver à l'heure en cours
     □ Transférer des données et intructions entre pro-
         cesseur et mémoire
     ☐ Écrire des données sur le dique dur
      □ transporter les processus du tourniquet au pro-
         cesseur
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
     \Box 1
     \square 42
      \Box 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 : □ 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"); Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu	<pre>int age = 18; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); }</pre>
	<pre>du struct, on écrit :</pre>	affichera: ☐ Mineur d); Majeur ☐ Mineur
14.	Sur unix (ou linux), la commande mkdir permet de : ouvrir un fichier texte créer un répertoire créer un fichier texte	□ Majeur □ rien 17. Soit la fonction f définie par : int f(int a)
15.	<pre>□ changer de répertoire courant</pre> Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire : □ n = factorielle(); □ n = factorielle(p, q); □ printf("%d", factorielle(n)); □ int factorielle(int 2);	<pre>{ printf("a = \n", %d); if (a > 0) { return 3; } return 4; }</pre>
	, , , , , , , , , , , , , , , , , , ,	Alors l'expression f(0) prendra la valeur :

	\Box 4
	\square 0
	\square 3
18.	Si carre est une fonction prenant en entrée un en tier et renvoyant le carré de cet entier, et que n es une variable entière définie et initialisée, il est correc d'écrire :
	☐ int carre(2);
	☐ int n = carre();
	\square n = carre(n);
	\square n = carre(int n);
19.	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 :
	\square dans lequel ces fonctions sont appelées dans le main
	\Box alphabétique
	\Box un ordre quel conque
	\Box dans lequel vous avez déclaré ces fonction
20.	Pour déclarer une fonction factorielle qui prend et argument un entier et renvoie sa factorielle on écrit :
	☐ struct int factorielle(int n);
	☐ int factorielle();
	☐ int factorielle(int x);
	☐ int factorielle(double n);

Éléments d'informatique – contrôle continue

10

16

17

}

Prénom:	Nom:
N° etu :	

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

- 1. Le langage C est un langage
 - \square interprété
 - □ lu, écrit, parlé
 - □ compilé
 - \square composé
- 2. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
 - \square #define N = 3
 - \square #define taille = 3
 - \square #define taille = N
 - □ #define N 3
- 3. 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?

- \square 0 1 0 1 0 1
- \square 0 0 1 1 2 2
- \Box 0 1 2 0 1 2
- \square 1 2 3 1 2
- 4. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
 - □ char 'c';
 - \square char c;
 - ☐ char "c";
 - ☐ int char;

5. Après exécution jusqu'à la ligne 15 du programme \mathcal{C} :

```
11 int main() {
12 int x = 5;
13
14 x = 3 * x + 1;
15
```

- \square la variable x vaut 16
- $\Box\,$ le programme affiche x
- \Box le programme affiche ****
- \Box la variable x vaut $-\frac{1}{2}$
- 6. Le code suivant :

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

affichera:

- □ Majeur
- \square Mineur
- □ Mineur Majeur
- \Box rien
- 7. 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
- \Box 6
- \square 42 \square 1

16

}

8. 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 ...
```

- \square le terminal affiche x = 2
- \Box le terminal affiche x = 5
- \square le terminal affiche 5
- □ le terminal affiche "Faux"
- 9. 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 :
 - $\hfill\Box$ exposant(double x, int n, int r);
 - \square void exposant(double x^n);
 - \square double exposant(double x, int n);
 - \square int exposant(double n, int x);
- 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 :

- $\Box (a.n == b.n) && (a.x == b.x)$ $\Box a = b$
- $\square a\{n, x\} == b\{n, x\}$
- □ a == b

```
17. Pour déclarer une fonction saisie_utilisateur qui
11. Soit un programme contenant les lignes suivantes :
                                                                  return 4;
                                                                                                                             demande à l'utilisateur d'entrer un entier au clavier et
     int i = 0;
                                                                                                                            renvoie cet entier on écrit :
     int j = 0;
                                                                Alors l'expression f(0) prendra la valeur :
                                                                                                                              ☐ int saisie_utilisateur();
     for (i = 0; i < 0; i = i + 1)
                                                                  \square 3
                                                                                                                              ☐ void saisie_utilisateur(char c);
                                                                  \Box 0
         for (j = 0; j < 5; j = j + 1)
                                                                                                                              □ void saisie_utilisateur(int n);
                                                                  \Box 4
                                                                                                                              ☐ saisie_utilisateur(scanf(%d));
            . . .
                                                            14. L'écriture 111 en binaire correspond au nombre natu-
                                                                                                                        18. Un programme en langage C doit comporter une et une
                                                                rel:
                                                                                                                             seule définition de la fonction :
     printf("j = %d\n", j);
                                                                                                                              \square init
                                                                  \square 3
                                                                                                                              □ main
                                                                  \square 111
                                                                                                                              \square include
                                                                  \square 8
                                                                                                                               □ begin
   qu'est ce qui sera affiché?
                                                                  \square 7
     \Box j = 5
                                                                                                                        19. Au début de la fonction main() on place le code :
                                                            15. Si cette erreur apparaît à la compilation :
     \Box j = %d
                                                                                                                              char i:
                                                                erreur: conflicting types for 'max', que doit-
      \Box j = 4
                                                                                                                              for (i = 'A'; i \le 'F'; i = i + 1)
                                                                on chercher dans le programme?
     \Box i = 0
                                                                  □ un désaccord entre la déclaration et la définition
                                                                                                                                printf("%c", i);
12. Le bus système sert à :
                                                                     d'une fonction
      □ Transférer des données et intructions entre pro-
                                                                  \square une fonction déclarée mais non définie
                                                                                                                              printf("\n");
        cesseur et mémoire
                                                                  ☐ une fonction appelée avant sa déclaration
                                                                                                                             Alors l'affichage sera:
      ☐ Arriver à l'heure en cours
                                                                  □ une directive préprocesseur #include manquante
                                                                                                                              □ cccccc
     □ transporter les processus du tourniquet au pro-
                                                                                                                              □ ABCDEF
                                                            16. Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                              \Box i
                                                                tions utilisées par votre programme principal. L'ordre
     ☐ Écrire des données sur le dique dur
                                                                dans lequel vous devez maintenant définir ces fonctions
                                                                                                                              \Box A
13. Soit la fonction f définie par :
                                                                est l'ordre :
                                                                                                                        20. Pour déclarer une fonction factorielle qui prend en
   int f(int a)
                                                                  □ dans lequel vous avez déclaré ces fonction
                                                                                                                            argument un entier et renvoie sa factorielle on écrit :
   {
                                                                                                                              ☐ struct int factorielle(int n);
                                                                  □ un ordre quelconque
      printf("a = \n", %d);
      if (a > 0)
                                                                                                                              ☐ int factorielle(double n);
                                                                  □ alphabétique
                                                                                                                               ☐ int factorielle();
                                                                  \square dans lequel ces fonctions sont appelées dans le
        return 3;
                                                                     main
                                                                                                                              ☐ int factorielle(int x);
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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
    □ Majeur
    \square rien
2. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
     ☐ #include <studlib.h>
    ☐ #appart <stdlib.h>
    ☐ #include <stdio.h>
    ☐ #include <studio.h>
3. Sous unix (ou linux), pour créer un répertoire TP4
  dans le répertoire courant on peut utiliser la com-
  mande:
    ☐ kwrite TP4
    □ mkdir TP4
    □ new TP4
    □ yppasswd
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 == TRUE) && (B == TRUE)
    \square !(!A || B) == (A && !B)
```

□ A && B

 \square (!A | | B)

```
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 42
     \Box 1
    \Box 0
     \Box 6
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:
     □В
     \Box C
     \square A
     □ b
7. Après exécution jusqu'à la ligne 14 du programme C:
 10
      int main() {
           int x = 5;
 11
 12
           printf(" x = %d\n", 2);
 13
 14
 15
           . . .
      }
 16
     □ le terminal affiche 5
     □ le terminal affiche "Faux'
     \square le terminal affiche x = 5
     \square le terminal affiche x = 2
```

```
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 0
     \square 3
 9. Un enregistrement permet de grouper plusieurs valeurs
    dans:
     □ ses cases
     \square ses blocs
     \square ses champs
      \square ses chants
10. 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(struct date_s d);
     □ void afficher_date(date_s d);
11. Dans la commande gcc, l'option -Wall signifie :
     \Box qu'il faut lancer un déboggueur
     ☐ qu'il faut indenter le fichier source
     ☐ que l'on veut voir tous les avertissements
     \square qu'on veut changer alétoirement de fond d'écran
12. 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 :
                                                           16. Après la déclaration : int mccarthy(int n);, il est
                                                                                                                          L'affichage dans le main est le suivant :
                                                               correct d'écrire :
     \square 8
                                                                                                                            \Box f(a,b)=8, a=3, b=5
                                                                 \square x = mccarthy(n);
     \Box 16
                                                                                                                            \Box f(a,b)=8, a=8, b=5
                                                                 \square n = mccarthy(p, q);
     \Box 0
                                                                                                                            \Box f(a,b)=13, a=8, b=5
                                                                 \square n = mccarthy();
     \Box 4
                                                                                                                            \Box f(3,5)=8, a=3, b=5
                                                                 \square int mccarthy(int 2);
13. Si n est une variable entière pour demander sa valeur
                                                                                                                       19. Lorsqu'un programme utilise printf ou scanf il faut
                                                           17. Si cette erreur apparaît à la compilation :
   à l'utilisateur, on utilise plutôt :
                                                                                                                          qu'il contienne l'instruction préprocesseur :
                                                               error: expected ';' before '}' token que doit-
     □ scanf("%d", &n);
                                                               on chercher dans le programme?
                                                                                                                            ☐ #appart <stdlib.h>
     □ printf("Valeur de n ? %g\n", n);
                                                                 □ un point-virgule manquant
                                                                                                                            ☐ #include <stdio.h>
     \square un débogueur
                                                                 \square une accolade en trop
                                                                                                                            ☐ #include <studio.h>
     □ printf("Valeur de n ? %d\n", n);
                                                                 \square une accolade manquante
                                                                                                                            ☐ #include <studlib.h>
                                                                 \square un point-virgule en trop
14. Pour déclarer une procédure afficher_menu sans ar-
                                                                                                                       20. Pour l'extrait de programme suivant :
   gument et qui ne renvoie rien on utilise :
                                                           18. Soit le programme principal suivant :
                                                                                                                             int somme = 0;
     ☐ double afficher_menu();
                                                               int main()
                                                                                                                             int serie[4] = \{2, 4, 10, 4\};
                                                               {
     ☐ char afficher_menu(printf("menu"));
                                                                                                                             for (i = 0; i < 4; i = i + 1)
                                                                int a = 3;
     ☐ int afficher_menu();
                                                                int b = 5;
     □ void afficher_menu();
                                                                                                                               somme = somme + serie[i];
                                                                printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     ☐ int afficher_menu(int char);
                                                                return EXIT_SUCCESS;
                                                                                                                             printf("somme = %d",somme);
15. Laquelle de ces écritures correspond à la déclaration
                                                                                                                          La valeur de somme affichée est :
   d'une variable de type caractère en langage C?
                                                               appelant la fonction f ainsi définie :
                                                                                                                            \Box 16
     \square int char;
                                                               int f(int a, int b)
                                                                                                                            \square 3
     \Box char c;
                                                                 a = a + b;
     ☐ char "c";
                                                                                                                             \square 20
                                                                 return a;
     ☐ char 'c';
                                                                                                                             \Box 6
```

т	•	-1
	acence	

Prénom:	Nom:
N° etu :	

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 en-
   tier et renvoyant le carré de cet entier, et que n est
   une variable entière définie et initialisée, il est correct
   d'écrire :
     \Box int carre(2);
     \square n = carre(int n);
     \square n = carre(n):
     \square int n = carre();
2. 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 ne sera pas détectée
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
     \square comportent une erreur qui sera détectée au cours
        de l'analyse syntaxique
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
     \square 5
     \Box 1
     \square 0
```

```
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. 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
    ☐ une fonction appelée avant sa déclaration
    □ un désaccord entre la déclaration et la définition
       d'une fonction
6. 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
    □ il comporte une boucle infinie
     \square il ne compile pas
7. 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 5
    \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
```

```
8. 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);
 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 :
      \square 16
      \Box 0
      \Box 4
     \square 8
10. 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
11. 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);
12. Une variable booléenne est un variable :
     ☐ réelle positive
     ☐ jamais nulle
      □ NaN (not a number, qui n'est pas un nombre)
     □ à laquelle une valeur vient d'être affectée
      \square qui est vraie ou fausse
```

13. Laquelle de ces écritures correspond à la déclaration	16. Après exécution jusqu'à la ligne 15 du programme C :	\square la longueur d'un mot mémoire
d'une variable de type caractère en langage C?	10	\Box l'instruction qui met fin à un programme
\square char c;	11 int main() {	\Box un chiffre binaire (0 ou 1)
□ char 'c';	12 int x = 5;	10.0
☐ int char;	$\begin{bmatrix} 13 \\ 14 \\ x = 3 * x + 1; \end{bmatrix}$	19. On considère deux variables booléennes A et B initialisées à TRUE et FALSE respectivement. Parmi les ex-
☐ char "c";	15	pressions booléennes suivantes, laquelle a pour valeur
14. Sur unix (ou linux), la commande mkdir permet de :	16	TRUE?
$\hfill \Box$ changer de répertoire courant	17 }	□ (!A B)
\square ouvrir un fichier texte	\square le programme affiche x	□ A && B
\square créer un fichier texte	\Box la variable x vaut $-\frac{1}{2}$	☐ (A == TRUE) && (B == TRUE)
□ créer un répertoire	☐ le programme affiche ****	☐ !(!A B) == (A && !B)
15. Au début de la fonction main() on place le code :	□ la variable x vaut 16	
char b = 'A';	17. Un enregistrement permet de grouper plusieurs valeurs	20. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la pro-
b = b + 2;	dans :	grammation structurée:
<pre>printf("%c\n", b);</pre>	□ ses blocs	\Box mettre les blocs en séquence les uns à la suite des
Alors l'affichage sera :	□ ses champs	autres
□ b	□ ses chants	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condi-
□В	☐ ses cases	tion
□ A	18. Un bit est:	\Box répéter un bloc tant qu'une condition est vérifée
□ С	\square un battement d'horloge processeur	\Box retourner un bloc

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	1,0111
n etu:	

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 \square une accolade en trop □ un point-virgule manquant \square une accolade manquante 2. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée : \square retourner un bloc □ répéter un bloc tant qu'une condition est vérifée □ sélectionner entre deux blocs à l'aide d'une condition \square mettre les blocs en séquence les uns à la suite des autres 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. Le bus système sert à : □ transporter les processus du tourniquet au pro-☐ Transférer des données et intructions entre processeur et mémoire ☐ Arriver à l'heure en cours

☐ Écrire des données sur le dique dur

```
5. Le code suivant :
    int i:
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
    \Box 01234
    \square 1 2 3 4
    \square 4 3 2 1
    \Box \ 4\ 3\ 2\ 1\ 0
6. Le code suivant :
    int age = 18;
    if (age < 18)
    {
        printf("Mineur\n");
    else
    {
        printf("Majeur\n");
  affichera:
    □ Majeur
    □ Mineur
    □ Mineur
       Majeur
     \square rien
7. 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>
```

```
8. 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
      \square 0
 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 = b
      □ a == b
      \square a{n, x} == b{n, x}
10. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
      ☐ char "c";
      □ char 'c';
      \Box char c;
      ☐ int char;
11. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 3
      \Box 5
      \Box 4
      \square 101
12. 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
```

```
13. Pour l'extrait de programme suivant :
                                                                   \Box int struct toto_s = {3, -1e10};
                                                                                                                                \Box f(a,b)=8, a=3, b=5
                                                                   \square int toto.n = 3;
                                                                                                                                \Box f(a,b)=8, a=8, b=5
      int somme = 0;
      int serie[4] = \{2, 4, 10, 4\};
                                                                   □ struct toto_s toto;
      for (i = 0; i < 4; i = i + 1)
                                                             16. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                                \Box f(3,5)=8, a=3, b=5
                                                                    int main() {
        somme = somme + serie[i];
                                                                                                                          18. Le code suivant :
                                                               11
                                                                         int x = 5;
                                                                                                                               int i;
                                                               12
      printf("somme = %d",somme);
                                                                         int y;
                                                               13
   La valeur de somme affichée est :
                                                               14
                                                                         y = x;
      \square 20
                                                                                                                                    printf("%d ", i);
                                                               15
     \Box 6
                                                               16
                                                                                                                               printf("\n");
                                                                    }
                                                               17
      \square 3
                                                                                                                              affichera:
      \Box 16
                                                                   □ le programme affiche "Faux"
                                                                                                                                \Box 0 1 2 3 4 5 6 7
                                                                   \square la variable y vaut 5
14. Si cet avertissement apparaît à la compilation :
                                                                                                                                \Box 02468
   warning: implicit declaration of function 'max'
                                                                   \square la variable x vaut 5 et la variable y vaut 0
   , que doit-on chercher dans le programme?
                                                                                                                                \square 0 2 4 6
                                                                   \Box la variable x vaut 0
      □ un désaccord entre la déclaration et la définition
                                                                                                                                \Box 0 1 2 3 4 5 6
                                                             17. Soit le programme principal suivant :
         d'une fonction
     ☐ une fonction appelée avant sa déclaration
                                                                 int main()
     □ une directive préprocesseur #include manquante
                                                                                                                                \square #define N = 3
                                                                  int a = 3;
     □ une fonction déclarée mais non définie
                                                                  int b = 5;
                                                                                                                                \square #define taille = 3
15. Si le code :
                                                                  printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
                                                                                                                                \square #define N 3
                                                                  return EXIT_SUCCESS;
   struct toto s
                                                                                                                                \square #define taille = N
   {
      int n;
                                                                 appelant la fonction f ainsi définie :
      double x;
                                                                                                                              et renvoyant un entier, il est correct d'écrire :
                                                                 int f(int a, int b)
   };
                                                                 {
                                                                                                                                \square int n = pgcd();
   précède la fonction main(), alors on peut écrire en
                                                                   a = a + b:
                                                                                                                                \square int pgcd(2);
   début de main():
                                                                   return a;
                                                                                                                                \square n = pgcd(n, 3);
                                                                 }
      \square toto_s struct z = {3, 0.5};
                                                                                                                                \square n = pgcd(int p, int q);
                                                                 L'affichage dans le main est le suivant :
      \square toto_s n, x;
```

```
\Box f(a,b)=13, a=8, b=5
     for (i = 0; i < 7; i = i + 2)
19. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
20. Si pgcd est une fonction prenant en entrée deux entiers
```

Éléments d'informatique – contrôle continue

Prénom:	Vom:
N° etu :	

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: □ avoir défini une constante symbolique de la taille de cette fonction □ l'avoir définie □ l'avoir déclarée et définie □ l'avoir déclarée 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 □ de doubler la mémoire disponible ☐ d'entretenir l'illusion que les processus tournent en parallèle 3. 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 \square 4 3 2 1 $\Box 01234$ 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(2/3);

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

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

```
5. 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);
    \Box n = pgcd(int p, int q);
    \square int n = pgcd();
6. 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'édition de liens
    ☐ l'analyse des entrées clavier
    ☐ l'analyse sémantique
7. Vous utilisez une boucle while quand :
     □ 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
    □ vous ne connaissez pas le nombre d'itérations de
       la boucle à l'avance
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;
    \square int toto.n = 3;
    \square toto_s struct z = {3, 0.5};
    \square int struct toto_s = {3, -1e10};
     □ struct toto_s toto;
```

```
9. 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>
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 :
      □ certaines données de la mémoire de travail
     \square en temps d'accès
     \square des processus
      \square les fichiers du disque
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(struct date_s d);
     ☐ int afficher_date(date_s d);
     □ struct date_s afficher_date(struct date_s d);
      □ void afficher_date(date_s d);
12. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      □ 111
      \square 3
     \square 8
     \square 7
13. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
     \square analyse harmonique
      □ analyse sémantique
     \square analyse lexicale
      □ analyse syntaxique
14. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
      ☐ char afficher_menu(printf("menu"));
     ☐ int afficher_menu();
      □ void afficher_menu();
      ☐ int afficher_menu(int char);
      ☐ double afficher_menu();
```

```
15. 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 prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
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 5
     \Box 0
```

```
17. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
     ☐ char "c";
     \Box char c;
     □ char 'c';
     ☐ int char;
18. 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
19. Lorsqu'un programme utilise printf ou scanf il faut
    qu'il contienne l'instruction préprocesseur :
     ☐ #include <studio.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studlib.h>
      ☐ #include <stdio.h>
```

```
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?
    □ j = 4
    □ j = 5</pre>
```

 \Box j = %d

 \Box j = 0

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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

2. 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 n'affiche rien

 $\Box\,$ il comporte une boucle infinie

 $\Box\,$ il ne compile pas

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 :

□ B□ C□ A

ПЪ

4. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.

```
    #define N = 3

    #define taille = 3

    #define N 3

    #define taille = N
```

 $5.\ \, {\rm Soit}$ 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?

 $\Box j = 5$ $\Box j = %d$ $\Box j = 0$

□ j = 4

6. 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

 \Box un ordre quel
conque

□ alphabétique

 \Box dans lequel vous avez déclaré ces fonction

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();

 \Box int afficher_menu();

☐ int afficher_menu(int char);

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 = b

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

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

□ a == b

9. Le code suivant :

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

□ 0 2 4 6 8

□ 8 6 4 2 0

 \square 8 6 4 2

 \square 8 2

10. Si le code:

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

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

```
\square 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};

```
14. Une segmentation fault est une erreur qui survient
                                                                                                                           17. Vous utilisez une boucle while quand:
        return 3:
      }
                                                                 lorsque:
                                                                                                                                 □ vous ne connaissez pas le nombre d'itérations de
      return 4;
                                                                    □ la division du programme en zones homogènes
                                                                                                                                     la boucle à l'avance
   }
                                                                       échoue
                                                                                                                                  □ l'incrément de la variable de boucle n'est pas 1
                                                                   □ le programme source a été enregistré sur le disque
   Alors l'expression f(0) prendra la valeur :
                                                                       dur en plusieurs morceaux et l'un d'entre eux ne
                                                                                                                                  □ vous n'avez pas déclaré de fonction
      \Box 0
                                                                       peut pas être chargé par le compilateur
                                                                                                                                  □ vous avez déjà fait un for dans le même pro-
      \square 3
                                                                    \square le programme tente d'afficher des caractères sur
                                                                                                                                     gramme principal
                                                                       une ligne qui va au delà de la largeur de la fenêtre
      \Box 4
                                                                                                                            18. On considère deux variables booléennes A et B initia-
                                                                       du terminal
12. Laquelle des analyses suivantes ne fait pas partie des
                                                                                                                               lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                    \square le programme tente d'accèder à une partie de la
   étapes de la compilation :
                                                                                                                               pressions booléennes suivantes, laquelle a pour valeur
                                                                       mémoire qui ne lui est pas réservée
                                                                                                                               TRUE?
      \square analyse harmonique
                                                             15. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                                  \square (!A || B)
                                                               10
      \square analyse syntaxique
                                                                     int main() {
                                                               11
                                                                                                                                  □ A && B
      \square analyse lexicale
                                                               12
                                                                          int x = 5;
                                                                                                                                  \square !(!A || B) == (A && !B)
      \square analyse sémantique
                                                               13
                                                                                                                                  \square (A == TRUE) && (B == TRUE)
                                                               14
                                                                          x = 3 * x + 1:
13. Soit la fonction g définie par :
                                                                15
                                                                                                                           19. Un programme en langage C doit comporter une et une
   int g(int a)
                                                                16
                                                                                                                                seule définition de la fonction :
                                                                    }
                                                                17
      printf("a = \n", %d);
                                                                                                                                  \square include
                                                                    \square le programme affiche x
      if (1 > 0)
                                                                    □ la variable x vaut 16
                                                                                                                                 □ init
      {
                                                                    □ le programme affiche ****
        return 5;
                                                                                                                                  □ begin
                                                                   \square la variable x vaut -\frac{1}{2}
                                                                                                                                  □ main
      return 7;
                                                             16. Pour déclarer une procédure afficher_date qui prend
                                                                                                                           20. Quel est l'opérateur de différence en C :
                                                                  en argument un struct date_s et affiche le contenu
                                                                  du struct, on écrit :
                                                                                                                                 \square \neq
   Alors l'expression g(0) prendra la valeur :
                                                                    ☐ int afficher_date(date_s d);
                                                                                                                                  \Box !
      \square 7
                                                                    □ void afficher_date(date_s d);
      \Box 0
                                                                                                                                  □ !=
                                                                   ☐ struct date_s afficher_date(struct date_s d);
      \Box 5
                                                                                                                                  □ <>
                                                                    □ void afficher_date(struct date_s d);
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

ar 1	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;
	☐ char "c";
	☐ int char;
	□ char 'c';
2.	Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
	☐ kwrite TP4
	□ new TP4
	☐ mkdir TP4
	\square yppasswd
3.	Un enregistrement permet de grouper plusieurs valeurs dans :
	\square ses blocs
	\square ses chants
	☐ ses cases
	\square ses champs
4.	Soit un programme contenant les lignes suivantes :
	<pre>int i = 0; int j = 0; for (i = 0; i < 3; i = i + 1)</pre>
	{
	for (j = 0; j < 5; j = j + 1) {
	 } }
	<pre>printf("j = %d\n", j);</pre>
	qu'est ce qui sera affiché par ce printf?

 \Box j = 0

 \Box j = 4

 \Box j = %d

```
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 :
     \Box 7
     \Box 5
     \Box 0
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(p, q);
     \square n = factorielle();
     ☐ int factorielle(int 2);
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 0
     \square 3
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 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);

```
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
     \square 8 6 4 2
     \square 8 2
     \Box 02468
10. 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 1 0 1 0 1 0 1
     \Box 1 2 1 2 3
     \Box 0 0 0 1 1 1
11. Sur unix (ou linux), la commande mkdir permet de :
      □ changer de répertoire courant
      \square ouvrir un fichier texte
     □ créer un répertoire
```

 \square créer un fichier texte

```
12. 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
      \square 4 3 2 1
     \Box 01234
13. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
      ☐ #include <studlib.h>
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
      ☐ #appart <stdlib.h>
14. 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
```

```
15. Sous unix (ou linux), la commande cd permet de :
      \Box jouer de la musique
      □ récupérer un programme arrêté avec la commande
      □ détruire un fichier
      □ changer de répertoire courant
      □ ouvir un bureau partagé (common desktop)
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)=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
17. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \Box 0
      \Box 1
      \square 0.5
      \square 1.5
```

```
18. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square x = mccarthy(n);
      \square n = mccarthy();
     \Box int mccarthy(int 2);
     \square n = mccarthy(p, q);
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(char c);
      ☐ int saisie_utilisateur();
      □ void saisie_utilisateur(int n);
20. 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 x vaut 3
```

 \square la variable y vaut 5

Éléments d'informatique – contrôle continue

Prénom:	Nom:
	110111
N° etu :	
ii cuu.	

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

	eponse lausse. Durce: 20 minutes.
1.	Un fichier source est :
	\Box un document qui doit être protégé
	\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
2.	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 :
	□ A
	□ b
	□ С
	□В
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);
	☐ int afficher_date(date_s d);
	☐ void afficher_date(struct date_s d);
	\Box struct date_s afficher_date(struct date_s
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 :
	$\hfill \square$ certaines données de la mémoire de travail
	\Box les fichiers du disque
	\square des processus
	□ en temps d'accès

```
5. Lorsqu'un programme utilise printf ou scanf il faut
      qu'il contienne l'instruction préprocesseur :
        ☐ #include <studio.h>
        ☐ #appart <stdlib.h>
        ☐ #include <stdio.h>
        ☐ #include <studlib.h>
   6. 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 0 1 2 3
        \Box 01234
        \square 4 3 2 1
   7. Quels calculs peut-on programmer en programmation
      structurée?
        ☐ 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
        ☐ 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
d);
           machine
   8. 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)
        □ A && B
        \square (A == TRUE) && (B == TRUE)
```

 \square (!A || B)

```
9. L'écriture 111 en binaire correspond au nombre natu-
   rel:
      \square 7
     □ 111
     \square 8
     \square 3
10. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \square x = mccarthy(n);
     \square n = mccarthy(p, q);
     \Box int mccarthy(int 2);
     \square n = mccarthy();
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 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
12. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ char afficher_menu(printf("menu"));
      ☐ int afficher_menu(int char);
      ☐ double afficher_menu();
      ☐ int afficher_menu();
```

□ void afficher_menu();

 13. 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 □ une directive préprocesseur #include manquante 	 16. 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 □ dans lequel vous avez déclaré ces fonction 	<pre>19. Soit la fonction f définie par : int f(int a) { printf("a = \n", %d); if (a > 0) { return 3; } }</pre>
☐ une faute de frappe dans un appel de fonction	□ un ordre quelconque □ alphabétique	return 4;
14. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : □ analyse syntaxique □ analyse sémantique □ analyse harmonique □ analyse lexicale 15. Dans la commande gcc, l'option -Wall signifie : □ qu'il faut lancer un déboggueur □ que l'on veut voir tous les avertissements □ qu'il faut indenter le fichier source	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. 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	Alors l'expression f(0) prendra la valeur : 0 3 4 20. Vous utilisez une boucle while quand : vous avez déjà fait un for dans le même programme principal vous n'avez pas déclaré de fonction vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
$\hfill \square$ qu'on veut changer alétoirement de fond d'écran	☐ Écrire des données sur le dique dur	\Box l'incrément de la variable de boucle n'est pas 1

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
N° etu:	

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? ☐ une directive préprocesseur #include manquante ☐ une faute de frappe dans un appel de fonction □ un caractère interdit en C □ une variable non déclarée 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 : \Box 6 \square 3 \square 20 \Box 16 3. 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 y, int x); \square int pgcd(int x, y); 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 : \square en temps d'accès \square des processus

□ certaines données de la mémoire de travail

 \square les fichiers du disque

```
5. 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 4 3 2 1
     \square 0 1 2 3
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=%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);
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. 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
 9. 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 le bus explose
     □ le terminal affiche 8
     □ la case mémoire 8 contiendra 0
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 n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
11. 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
      \square 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
```

```
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. 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
14. 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};
      □ struct toto_s toto;
```

```
\Box int struct toto_s = {3, -1e10};
      \square int toto.n = 3:
      \square toto_s n, x;
15. 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 n = carre(n):
      \square int n = carre();
16. Un bit est:
      \square un battement d'horloge processeur
      \square un chiffre binaire (0 ou 1)
      □ l'instruction qui met fin à un programme
      □ la longueur d'un mot mémoire
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 0 1 0 1 0 1
      \Box 0 1 2 0 1 2
      \Box 0 0 1 1 2 2
      \Box 1 2 3 1 2
```

```
18. 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
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 5
     \Box 0
     \Box 4
     \Box 1
20. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
     \Box int mccarthy(int 2);
     \square n = mccarthy();
     \square n = mccarthy(p, q);
```

 \square x = mccarthy(n);

Prénom: Nom: N° etu :

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 < 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 j = 0
    \Box i = %d
    \Box i = 5
2. Quels calculs peut-on programmer en programmation
  structurée?
       machine
       langage machine
       ghetti
```

```
□ en programmation structurée on peut program-
       mer tous les calculs programmables en langage
    ☐ il y a des calculs programmables en programma-
       tion structurée qui ne sont pas programmables en
    □ certains programmes sont de vrais plats de spa-
    ☐ il v a des calculs programmables en langage ma-
       chine et qui ne sont pas programmables en pro-
       grammation structurée
3. 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
```

```
4. 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
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
     \square l'incrément de la variable de boucle n'est pas 1
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 :
     \Box 0
     \Box 5
     \square 7
6. 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 1 2 3 4
     \square 4 3 2 1
7. 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
      \square le terminal affiche x = 2
      □ le terminal affiche 5
      □ le terminal affiche "Faux"
 8. 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
 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(char c);
     ☐ saisie_utilisateur(scanf(%d));
      ☐ int saisie_utilisateur();
      □ void saisie_utilisateur(int n);
10. Quel est l'opérateur de différence en C :
      \square \neq
      \Box !
      □!=
     □ <>
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 : \square 4	15. 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 :	18. Quel est le problème d'un programme comportant les lignes suivantes?
□ 0 □ 1 □ 5	<pre>int a = 0; int b = 20; int n;</pre>	<pre>while (1) { printf("coucou\n"); }</pre>
 12. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C? □ int char; □ char "c"; □ char c; 	<pre>scanf("%d", &n); while(cond) { scanf("%d", &n); }</pre> Quelle est la condition cond:	 □ il risque d'afficher bonjour à la place de coucou □ il n'affiche rien □ il comporte une boucle infinie □ il ne compile pas
□ char 'c'; 13. Un fichier source est : □ un document de référence du système	□ (a<=n) && (n<=b) □ a<=n<=b □ (a <n) (n="" ="">b)</n)>	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 :
 □ 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 □ un fichier texte qui sera traduit en instructions processeur 	☐ (n<=a) && (n<=b) 16. 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	☐ for(i=1;i<5;i=i+1) ☐ for(i=0;i<=5;i=i+1) ☐ for(i=1;i<=5;i=i+1) ☐ for(i=0;i<5;i=i+1) 20. Vous avez déclaré préalablement un ensemble de fonc-
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 ☐ un désaccord entre la déclaration et la définition d'une fonction ☐ une fonction déclarée mais non définie	 □ une variable non déclarée □ un caractère interdit en C □ une directive préprocesseur #include manquante 17. Un bit est : □ un chiffre binaire (0 ou 1) □ un battement d'horloge processeur □ la longueur d'un mot mémoire 	tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions est l'ordre : \[\subseteq \text{dans lequel ces fonctions sont appelées dans le main} \] \[\subseteq \text{dans lequel vous avez déclaré ces fonction} \] \[\subseteq \text{un ordre quelconque} \]
☐ une fonction appelée avant sa déclaration	☐ l'instruction qui met fin à un programme	□ alphabétique

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Vous utilisez une boucle while quand: □ vous n'avez pas déclaré de fonction \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 □ vous avez déjà fait un for dans le même programme principal 2. Sous unix (ou linux), la commande 1s permet de : \square afficher la liste de fichiers contenus dans un répertoire □ compiler un programme □ voir des clips musicaux \square afficher le contenu d'un fichier texte 3. 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 4. Sur unix (ou linux), la commande mkdir permet de : □ créer un fichier texte \square ouvrir un fichier texte □ créer un répertoire □ changer de répertoire courant 5. 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":

```
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 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
7. Si pgcd est une fonction prenant en entrée deux entiers
   et renvoyant un entier, il est correct d'écrire :
    \square int n = pgcd();
    \Box n = pgcd(int p, int q);
    \square n = pgcd(n, 3);
     \square int pgcd(2);
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 :
     \Box 4
     \Box 0
     \square 8
     \Box 16
9. Le langage C est un langage
     \square compilé
     □ composé
     □ lu, écrit, parlé
     □ interprété
```

```
10. 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 programma-
        tion structurée qui ne sont pas 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
11. 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);
12. 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));
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 void exposant(double x^n);
     \square double exposant(double x, int n);
14. 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):
```

15.	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
	\square l'analyse des entrées clavier
	\Box l'édition de liens
16.	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);
	☐ int carre(2);
	□ n = carre(int n);
	\square int n = carre();
17.	L'ordonnancement par tourniquet permet :
	\Box de doubler la mémoire disponible
	\Box de ne pas perdre de temps avec la commutation de contexte

```
\square d'entretenir l'illusion que les processus tournent
         en parallèle
      □ d'afficher des ronds colorés à l'écran
18. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
     printf("\n");
    affichera:
     \square 8 6 4 2
     \Box 8 6 4 2 0
     \square 8 2
      \Box 02468
19. 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
     \Box (a<n) || (n>b)
      \square (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
20. 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
     \square dans lequel ces fonctions sont appelées dans le
        main
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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

ır ré	ponse fausse. Durée : 20 minutes.
1. U	In fichier source est:
	\Box un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
	\Box un document qui doit être protégé
	\square un document illisible pour les humains
	\Box un fichier texte qui sera traduit en instructions processeur
	\Box un document de référence du système
2. I	Le bus système sert à :
	\Box transporter les processus du tourniquet au processeur
	\Box Transférer des données et intructions entre processeur et mémoire
	\Box Écrire des données sur le dique dur
	\Box Arriver à l'heure en cours
d	Sous unix (ou linux), pour créer un répertoire TP4 lans le répertoire courant on peut utiliser la comnande :
	□ new TP4
	\square yppasswd
	☐ mkdir TP4
	☐ kwrite TP4
	Quel est le problème d'un programme comportant les ignes suivantes?
w -{	hile (1)
}	<pre>printf("coucou\n");</pre>
	\Box il ne compile pas
	\square il comporte une boucle infinie
	□ il risque d'afficher bonjour à la place de coucou
	□ il n'affiche rien

```
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?
     □ A && B
     \square (!A || B)
     \square (A == TRUE) && (B == TRUE)
     \square !(!A \mid | B) == (A \&\& !B)
6. Le type des réels en C est :
     □ char
     \square real
     \square double
     \square int
7. 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
     \Box l'incrément de la variable de boucle n'est pas 1
     □ vous n'avez pas déclaré de fonction
8. 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

```
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 = %d
     \Box j = 5
     \Box j = 4
     \Box j = 0
10. Sous unix (ou linux), la commande cd permet de :
     ☐ récupérer un programme arrêté avec la commande
     □ jouer de la musique
     □ changer de répertoire courant
     ☐ détruire un fichier
     □ ouvir un bureau partagé (common desktop)
11. 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
     \square 0 2 4 6
```

 $\Box 0123456$

12. Soit la fonction g définie par :	$\square x = racine(racine(x)*racine(x));$	\square qui est vraie ou fausse
int g(int a)	\square x = racine(x * x) - racine(x);	☐ réelle positive
{	$\square x - 1 = racine(x);$	□ NaN (not a number, qui n'est pas un nombre)
<pre>printf("a = \n", %d); if (1 > 0) {</pre>	15. Une segmentation fault est une erreur qui survient lorsque :	18. Si cette erreur apparaît à la compilation : Undefined symbols :"_prinft" ou référence indéfinie vers « prinft » que doit-
return 5;	☐ le programme source a été enregistré sur le disque dur en plusieurs morceaux et l'un d'entre eux ne	on chercher dans le programme?
return 7;	peut pas être chargé par le compilateur	$\hfill\Box$ une variable non déclarée
}	☐ la division du programme en zones homogènes	\square une directive préprocesseur #include manquante
Alors l'expression g(0) prendra la valeur :	échoue	$\hfill\Box$ une faute de frappe dans un appel de fonction
\Box 5	☐ le programme tente d'accèder à une partie de la	$\hfill\Box$ un caractère interdit en C
□ 7 □ 0	mémoire qui ne lui est pas réservée □ le programme tente d'afficher des caractères sur	19. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :
13. Si n est une variable entière pour demander sa valeur	une ligne qui va au delà de la largeur de la fenêtre du terminal	☐ double afficher_menu();
à l'utilisateur, on utilise plutôt :	16. Pour déclarer une fonction factorielle qui prend en	☐ int afficher_menu(int char);
☐ printf("Valeur de n ? %d\n", n);	argument un entier et renvoie sa factorielle on écrit :	\square void afficher_menu();
□ scanf("%d", &n);	☐ int factorielle(double n);	☐ char afficher_menu(printf("menu"));
\Box un débogueur	☐ int factorielle();	☐ int afficher_menu();
\square printf("Valeur de n ? %g\n", n);	☐ int factorielle(int x);	20. Le langage C est un langage
14. Si racine est une fonction prenant en entrée un réel	☐ struct int factorielle(int n);	\square composé
et renvoyant la racine carrée de cet réel, et que x est	17. Une variable booléenne est un variable :	\square compilé
une variable réelle définie et initialisée, il est incorrect d'écrire :	□ jamais nulle	\Box interprété
\Box x = racine(2/3);	☐ à laquelle une valeur vient d'être affectée	\Box lu, écrit, parlé

Éléments d'informatique – contrôle continue

int x=0;

x=x+1;

{

for(i=0.i<5.i=i+1)

Prénom:	Nom:
N° etu :	

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

- pressions booléennes suivantes, laquelle a pour valeur TRUE?

 (!A || B)

 A && B
 - ☐ !(!A || B) == (A && !B)
 ☐ (A == TRUE) && (B == TRUE)
- 3. Le code suivant :

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 0 2 4 6 8
```

 \Box 8 6 4 2 0

_ 86420

 $\square 8642$ $\square 82$

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(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
5. Quel est l'opérateur de différence en C :
     □!
    □ !=
    \square \neq
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. 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
    \Box une gamme de fréquence de fonctionnement du
    \square un composant qui contient la liste des fichiers du
       système
8. Les lignes
  int i;
```

```
□ 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
      □ comportent une erreur qui ne sera pas détectée
 9. Un enregistrement permet de grouper plusieurs valeurs
   dans:
      □ ses cases
      \square ses champs
      □ ses chants
      \square ses blocs
10. Le code suivant :
     int age = 18;
     if (age < 18)
          printf("Mineur\n");
     else
     {
          printf("Majeur\n");
    affichera:
      □ Mineur
      □ Mineur
         Majeur
      □ Majeur
      \square rien
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 - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
      \square x = racine(x * x) - racine(x):
```

12. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	Alors l'expression $f(0)$ prendra la valeur :	17. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs
int char;		passés en arguments on écrit :
☐ char "c";	\Box 3	☐ int pgcd(int x, int x);
□ char c;	15. Après exécution jusqu'à la ligne 15 du programme C :	☐ int pgcd(int x, y);
☐ char 'c';	10 11 int main() {	☐ int pgcd(int y, int x);
13. Une segmentation fault est une erreur qui survient	11 int main() { 12 int x = 5;	□ void pgcd(int x, int y);
lorsque :	13	18. Le type des réels en C est :
□ le programme tente d'afficher des caractères sur	14 x = 3 * x + 1; 15	\Box char
une ligne qui va au delà de la largeur de la fenêtre du terminal	16	\square double
\square le programme tente d'accèder à une partie de la	17 }	\square real
mémoire qui ne lui est pas réservée	☐ le programme affiche ****	\square int
☐ la division du programme en zones homogènes échoue	☐ le programme affiche x☐ la variable x vaut 16☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	19. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
☐ le programme source a été enregistré sur le disque	\Box la variable x vaut $-\frac{1}{2}$	□ n = pgcd(int p, int q);
dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur	16. Le code suivant : int i;	☐ int pgcd(2);
14. Soit la fonction f définie par :	for (i = 1; i < 5; i = i + 1)	$\square \ n = pgcd(n, 3);$
int f(int a)	<pre>{ printf("%d ", i);</pre>	☐ int n = pgcd();
<pre>{ printf("a = \n", %d);</pre>	} printf("\n");	20. Un programme en langage C doit comporter une et une seule définition de la fonction :
if (a > 0)	affichera:	\Box init
return 3;		\Box begin
} return 4;	$\square \ 4\ 3\ 2\ 1$ $\square \ 4\ 3\ 2\ 1$	\Box include
}		□ main

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 = 4
    \Box j = %d
     \Box j = 0
     \Box i = 5
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.n == b.n) && (a.x == b.x)
    □ a == b
    \Box a = b
    \square a{n, x} == b{n, x}
3. 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
```

```
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 :
     \Box 0
     \square 7
     \square 5
5. 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, y);
     \square int pgcd(int x, int x);
     \square int pgcd(int y, int x);
6. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \square 5
     \square 3
     \Box 4
     \square 101
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. 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 carre(2);
      \square int n = carre():
 9. 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
      ☐ le programme affiche "Faux"
      \square la variable x vaut 0
      \square la variable y vaut 5
10. Le type des réels en C est :
      \square real
      \square int
      □ double
      □ char
11. Une de ces manière de composer les blocs de pro-
    grammes ne fait pas partie des opérations de la pro-
    grammation structurée :
      \square retourner un bloc
      □ répéter un bloc tant qu'une condition est vérifée
      \square sélectionner entre deux blocs à l'aide d'une condi-
         tion
      \square mettre les blocs en séquence les uns à la suite des
         autres
12. Sur unix (ou linux), la commande mkdir permet de :
      □ changer de répertoire courant
      □ créer un répertoire
      \square ouvrir un fichier texte
      \square créer un fichier texte
```

13.	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
	\Box 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
14.	Sous unix (ou linux), pour créer un répertoire TP4
	dans le répertoire courant on peut utiliser la com-
	mande:
	☐ kwrite TP4
	☐ mkdir TP4
	□ new TP4
	\square yppasswd
15.	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);
	☐ int pgcd(2);
	\square int n = pgcd();
	\square n = pgcd(n, 3);
16.	Pour compiler un programme prog.c, on utilise la
	ligne de commande :
	☐ gcc prog.exe -Wall -o prog.c
	\square gcc prog.c -o -Wall prog.exe
	☐ gcc -Wall prog.c -o prog.exe
	□ gcc -Wall prog.exe -o prog.c

```
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 :
     \Box 0
     \Box 4
     \square 3
18. 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();
19. 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 #include manquante
      \square une fonction déclarée mais non définie
     \square un désaccord entre la déclaration et la définition
        d'une fonction
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(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

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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
```

□ a == b

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. 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");
```

```
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 0 1 1 2 2 3
```

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

 \Box 0 1 2 3 0 1 2

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

5. Le code suivant :

```
for (i = 1; i < 5; i = i + 1)
    printf("%d ", i);
printf("\n");
affichera:
```

```
\Box 01234
```

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

```
\Box 1234
```

 \square 4 3 2 1

6. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction

```
\square int %d:
□ loop i;
```

 \square int k:

 \square int loop n;

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 :

 \square un ordre quelconque

□ alphabétique

□ dans lequel vous avez déclaré ces fonction

□ dans lequel ces fonctions sont appelées dans le main

```
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 :

	;	3

 \Box 4

 \square 0

9. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :

	${\tt void}$	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)		int	afficher	date	(date_s	d)
-------------------------------	--	-----	----------	------	---------	----

10. 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>

11. Un registre du processeur est :

□ une case mémoire interne au processeur	qui	ser
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

```
15. Soit le programme principal suivant :
12. Pour l'extrait de programme suivant :
                                                                                                                              \Box C
      int somme = 0;
                                                                                                                              □ b
                                                                int main()
      for (i = 0; i < 5; i = i + 1)
                                                                                                                              □В
                                                                 int a = 3;
                                                                                                                        18. Pour déclarer une fonction saisie_utilisateur qui
        somme = somme + i;
                                                                 int b = 5;
                                                                                                                            demande à l'utilisateur d'entrer un entier au clavier et
        i = i + 1; /* attention ! */
                                                                 printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                            renvoie cet entier on écrit :
                                                                return EXIT_SUCCESS;
      printf("somme = %d",somme);
                                                                                                                             □ void saisie_utilisateur(int n);
   La valeur de somme affichée est :
                                                               appelant la fonction f ainsi définie :
                                                                                                                             ☐ int saisie_utilisateur();
      \Box 6
                                                               int f(int a, int b)
                                                                                                                              □ void saisie_utilisateur(char c);
     \Box 0
                                                                                                                              ☐ saisie_utilisateur(scanf(%d));
                                                                  a = a + b;
     \Box 10
                                                                                                                        19. Soit la fonction g définie par :
                                                                  return a;
     \square 15
13. Un bit est:
                                                                                                                            int g(int a)
                                                               L'affichage dans le main est le suivant :
     \square un battement d'horloge processeur
                                                                 \Box f(a,b)=8, a=8, b=5
                                                                                                                              printf("a = \n", %d);
     \square un chiffre binaire (0 ou 1)
                                                                                                                              if (1 > 0)
                                                                 \Box f(3,5)=8, a=3, b=5
     ☐ l'instruction qui met fin à un programme
                                                                                                                              {
                                                                 \Box f(a,b)=13, a=8, b=5
     \square la longueur d'un mot mémoire
                                                                                                                                return 5;
                                                                 \Box f(a,b)=8, a=3, b=5
14. Le code suivant :
                                                                                                                              return 7;
                                                            16. Un enregistrement permet de grouper plusieurs valeurs
     int somme = 0;
                                                                dans:
     int i;
                                                                                                                            Alors l'expression g(0) prendra la valeur :
                                                                 \square ses cases
     for (i = 1; i < 4; i = i + 1)
                                                                                                                              \Box 0
                                                                 \square ses champs
       somme = somme + i;
                                                                 \square ses chants
                                                                                                                              \Box 7
                                                                 \square ses blocs
                                                                                                                              \Box 5
     printf("%d", somme);
                                                           17. Au début de la fonction main() on place le code :
                                                                                                                        20. Le langage C est un langage
   affichera:
                                                                 char b = 'A';
                                                                                                                              □ interprété
     \Box 6
                                                                 b = b + 2;
                                                                                                                              □ compilé
                                                                 printf("%c\n", b);
     \Box 1
                                                                                                                             □ composé
     \square 42
                                                                Alors l'affichage sera :
                                                                                                                              □ lu, écrit, parlé
     \Box 0
                                                                 □ A
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
n etu.	

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:
    ☐ 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);
2. Le code suivant :
    int age = 18;
    if (age < 18)
        printf("Mineur\n");
    else
        printf("Majeur\n");
   }
  affichera:
    □ Majeur
    ☐ Mineur
       Majeur
    ☐ Mineur
    □ rien
3. Après exécution jusqu'à la ligne 14 du programme C:
10
      int main() {
          int x = 5;
11
12
          printf(" x = %d\n", 2);
13
14
15
 16
     }
    \square le terminal affiche x = 5
    □ le terminal affiche 5
    \square le terminal affiche x = 2
    □ le terminal affiche "Faux"
```

```
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=%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);
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 int carre(2);
    \square n = carre(int n);
    \square n = carre(n);
     \square int n = carre();
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):
    \square x = racine(racine(x)*racine(x)):
    \square x = racine(x * x) - racine(x);
     \square x = racine(2/3);
7. 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);
    ☐ printf("Valeur de n ? %g\n", n);
     □ scanf("%d", &n);
8. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     □ créer un fichier texte
     □ ouvrir un fichier texte
     □ changer de répertoire courant
```

```
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:
      ☐ ABCDEF
     \square A
     \Box i
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:
     ☐ a<=n<=b
     □ (a<=n) && (n<=b)
      □ (n<=a) && (n<=b)
      □ (a<n) || (n>b)
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 n = pgcd(int p, int q);
     \square int n = pgcd();
```

 \square int pgcd(2);

```
12. Pour l'extrait de programme suivant :
                                                                  \Box f(a,b)=8, a=8, b=5
                                                                  \Box f(a,b)=13, a=8, b=5
     int i = 0;
                                                                                                                               int i;
     int j = 0;
                                                                  \Box f(a,b)=8, a=3, b=5
     for (i = 0; i < 3; i = i + 1)
                                                                  \Box f(3,5)=8, a=3, b=5
                                                            15. Après exécution jusqu'à la ligne 15 du programme C:
         for (j = 0; j < 2; j = j + 1)
                                                                    int main() {
              printf("%d ", i);
                                                              11
                                                                         int x = 5;
         }
                                                              12
                                                                         int y;
                                                              13
     printf("\n");
                                                                                                                               \square 8 2
                                                              14
                                                                        y = x;
                                                              15
   qu'est ce qui sera affiché?
                                                              16
                                                                         . . .
     \Box 0 1 0 1 0 1
                                                                   }
                                                              17
     \Box 0 0 1 1 2 2
                                                                  \Boxla variable x vaut 5 et la variable y vaut 0
     \Box 0 1 2 0 1 2
                                                                  \Box la variable x vaut 0
      \Box 1 2 3 1 2
                                                                  \square la variable y vaut 5
13. Pour déclarer une fonction pgcd qui calcule et renvoie
                                                                                                                              ₹
                                                                  □ le programme affiche "Faux"
   le plus grand diviseur commun de deux entiers positifs
   passés en arguments on écrit :
                                                                                                                              }
                                                            16. Un bit est:
     \square void pgcd(int x, int y);
                                                                                                                               else
                                                                  □ la longueur d'un mot mémoire
                                                                                                                              {
     \square int pgcd(int y, int x);
                                                                  \square un battement d'horloge processeur
     \Box int pgcd(int x, int x);
                                                                                                                              }
                                                                  \square un chiffre binaire (0 ou 1)
     \Box int pgcd(int x, y);
                                                                  ☐ l'instruction qui met fin à un programme
14. Soit le programme principal suivant :
                                                            17. Soit la fonction f définie par :
   int main()
                                                                int f(int a)
   {
     int a = 3;
                                                                   printf("a = \n", %d);
     int b = 5;
                                                                   if (a > 0)
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                   {
    return EXIT_SUCCESS;
                                                                                                                               \square rien
                                                                     return 3;
   appelant la fonction f ainsi définie :
                                                                   return 4;
   int f(int a, int b)
                                                                Alors l'expression f(0) prendra la valeur :
      a = a + b;
                                                                  \Box 0
      return a;
                                                                  \Box 4
                                                                                                                               \square for(i=1;i<5;i=i+1)
                                                                   \square 3
   L'affichage dans le main est le suivant :
```

```
18. Le code suivant :
    for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
    printf("\n");
   affichera:
     \Box 02468
     \square 8 6 4 2
     \Box 8 6 4 2 0
19. Le code suivant :
     int age = 15;
    if (age < 18)
         printf("Mineur\n");
         printf("Majeur\n");
   affichera:
     □ Mineur
        Majeur
     □ Mineur
     □ Majeur
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)
     \square for(i=0;i<5;i=i+1)
     \square for(i=0;i<=5;i=i+1)
```

 \Box char c;

☐ char "c";

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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

d'une variable de type caractère en langage C?

1. Laquelle de ces écritures correspond à la déclaration

	☐ int char;	
	☐ char 'c';	
2.	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 afficher_menu(int char);	
	☐ void afficher_menu();	
3.	Sur unix (ou linux), la commande mkdir permet de :	
	\Box créer un fichier texte	
	\square ouvrir un fichier texte	
	\Box changer de répertoire courant	
	□ créer un répertoire	
4.	Le code suivant :	
	int i;	
	for (i = 8; i > 0; i = i - 2)	
	{	
	<pre>printf("%d ", i); }</pre>	
	<pre>printf("\n");</pre>	
	affichera:	
	\Box 0 2 4 6 8	
	\Box 8 6 4 2 0	
	$\Box \ 8 \ 6 \ 4 \ 2$	
	□ 8 2	
5	Si cette erreur apparaît à la compilation :	
υ.	erreur: conflicting types for 'max', que doit-	
	on chercher dans le programme?	
	\square un désaccord entre la déclaration et la définition	
	d'une fonction	
	\Box une fonction appelée avant sa déclaration	
	\Box une directive préprocesseur $\verb"#include"$ man quante	
	\Box une fonction déclarée mais non définie	

```
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
      □ dans lequel vous avez déclaré ces fonction
      □ alphabétique
      □ un ordre quelconque
 7. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 101
      \square 3
      \Box 4
      \Box 5
 8. 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);
 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
      \square (!A || B)
      \square !(!A \mid | B) == (A \&\& !B)
      \square (A == TRUE) && (B == TRUE)
10. Le langage C est un langage
      □ compilé
      □ composé
      □ interprété
```

□ lu, écrit, parlé

```
11. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      □ ses cases
      \square ses blocs
      \square ses champs
      \square ses chants
12. 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 déclarée
      □ l'avoir définie
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 :
      \Box 0
      \square 7
      \Box 5
14. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
          printf("%d ", i);
     printf("\n");
    affichera:
      \Box 0123
      \Box \ 4\ 3\ 2\ 1\ 0
      \Box 01234
      \square 4 3 2 1
```

15. Si x est une variable réelle (de type double) alors	<pre>printf("%d ", i);</pre>	☐ int toto[taille=5];
x = 3/2 lui affecte la valeur :	}	☐ int[] new tableau(5);
\Box 0	<pre>printf("\n");</pre>	☐ int toto[5];
\square 0.5	affichera:	☐ char tableau[5];
\square 1.5	$\Box \ 0\ 1\ 2\ 3\ 4$	
□ 1	\square 1 2 3 4	20. Une segmentation fault est une erreur qui survient
16. Laquelle des analyses suivantes ne fait pas partie des	$\Box \ 4\ 3\ 2\ 1\ 0$	lorsque:
étapes de la compilation :	\square 4 3 2 1	☐ la division du programme en zones homogènes échoue
\square analyse harmonique	18. Le type des réels en C est :	
$\hfill\Box$ analyse lexicale	□ char	☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée
\square analyse sémantique	\square double	☐ le programme tente d'afficher des caractères sur
\square analyse syntaxique	\square real	une ligne qui va au delà de la largeur de la fenêtre
17. Le code suivant :	\Box int	du terminal
int i;	19. Pour déclarer un tableau d'entiers de taille 5, on peut	\Box le programme source a été enregistré sur le disque
for $(i = 4; i \ge 0; i = i - 1)$	utiliser l'instruction	dur en plusieurs morceaux et l'un d'entre eux ne
{	☐ int tab[] = 5;	peut pas être chargé par le compilateur

Éléments d'informatique – contrôle continue

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

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)
```

- □ a == b
- \Box a = b

2. Le code suivant :

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 0 2 4 6 8
```

 \square 8 6 4 2

 \Box 8 6 4 2 0

 \square 8 2

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

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

☐ char tableau[5];

```
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 4

 \Box 5

 \Box 1

 \Box 0

5. Quel est l'opérateur de différence en C :

 \Box !

□ !=

□ <>

 $\square \neq$

6. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction

```
\square loop i;
```

 \square int loop n;

 \square int k;

☐ int %d;

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

```
\square char "c";
```

☐ char 'c';

 \square int char;

 \square char c;

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 0

□ 5

9. 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();

 \square n = carre(n);

 \square int carre(2);

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 :

 \square 3

 \Box 4

 \Box 0

11. Sur unix (ou linux), la commande mkdir permet de :	14. Quels calculs peut-on programmer en programmation	Alors l'affichage sera :
\Box changer de répertoire courant	structurée?	□ С
\Box créer un fichier texte	☐ en programmation structurée on peut programmer tous les calculs programmables en langage	□ В
□ créer un répertoire	machine	□ A
□ ouvrir un fichier texte	☐ il y a des calculs programmables en programma- tion structurée qui ne sont pas programmables en	□ъ
12. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	chine et qui ne sont pas programmables en pro-	18. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :
\square une directive préprocesseur #include manquante	grammation structurée □ certains programmes sont de vrais plats de spa-	\square struct date_s afficher_date(struct date_s d);
\square une fonction appelée avant sa déclaration	ghetti	☐ void afficher_date(struct date_s d);
\square une fonction déclarée mais non définie	15. Après la déclaration : int mccarthy(int n);, il est	☐ int afficher_date(date_s d);
un désaccord entre la déclaration et la définition	correct d'écrire :	☐ void afficher_date(date_s d);
d'une fonction	☐ int mccarthy(int 2);	19. Un programme en langage C doit comporter une et une
13. Les lignes	□ x = mccarthy(n);	seule définition de la fonction :
int i;	n = mccarthy();	☐ main
int x=0; for(i=0,i<5,i=i+1)	□ n = mccarthy(p, q);	□ include
{	16. Soient deux variables entières x et y initialisées à 4 et 5 respectivement. L'affichage x=4 et y=5 est obtenu	
x=x+1;	avec la commande :	□ init
}	\Box printf("x=%d et y=%d\n",x y);	\Box begin
\square comportent une erreur qui sera détectée au cours	\Box printf("x=%x et y=%y\n");	20. L'écriture <u>101</u> en binaire correspond au nombre natu-
de l'édition de lien	\Box printf("x=%d et y=%d\n,x,y");	rel:
\Box ne comportent aucune erreur	\Box printf("x=%d et y=%d\n",x,y);	□ 5
$\hfill \square$ comportent une erreur qui sera détectée au cours	17. Au début de la fonction main() on place le code :	□ 101
de l'analyse syntaxique	char b = 'A';	\square 4
\Box comportent une erreur qui ne sera pas détectée	b = b + 2; printf("%c\n", b);	

Éléments d'informatique – contrôle continue

Prénom : Nom : No etu :

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);</pre>
```

qu'est ce qui sera affiché par ce printf?

```
\Box j = 0
```

```
\Box j = %d
```

□ j = 5

2. 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);
    }
}</pre>
```

qu'est ce qui sera affiché?

```
□ 0 1 2 0 1 2
```

 \square 0 1 0 1 0 1 0 1

 $\square \ 0 \ 0 \ 0 \ 1 \ 1 \ 1$

□ 1 2 1 2 3

3. Vous utilisez une boucle while quand :

```
\hfill \square vous n'avez pas déclaré de fonction
```

 \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
```

- □ vous avez déjà fait un for dans le même programme principal
- 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}
```

□ a == b

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

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);</pre>
```

La valeur de somme affichée est :

- \Box 10
- \Box 6
- \Box 15
- \Box 0

6. Un enregistrement permet de grouper plusieurs valeurs dans :

	ses	champs
ш	SCS	champs

П	ses	cases
ш	סכס	Cases

```
7. Le bus système sert à :
```

Transférer	des	données	et	intructions	entre	pro
cesseur et	mén	noire				

- □ transporter les processus du tourniquet au processeur
- ☐ Arriver à l'heure en cours
- ☐ Écrire des données sur le dique dur
- 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};
- \Box int toto.n = 3;
- \Box toto_s struct z = {3, 0.5};

9. 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

10. 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
- \square gcc prog.exe -Wall -o prog.c
- $\hfill\Box$ gcc -Wall prog.c -o prog.exe

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

	int	<pre>factorielle()</pre>	;
--	-----	--------------------------	---

- ☐ int factorielle(double n);
- □ struct int factorielle(int n);
- \square int factorielle(int x);

12. Soit la fonction f définie par :	\Box printf("x=%d et y=%d\n,x,y");	18. Quel est l'opérateur de différence en C :
int f(int a)	\Box printf("x=%x et y=%y\n");	□≠
{	\Box printf("x=%d et y=%d\n",x y);	_ !
printf("a = \n", %d);	\square printf("x=%d et y=%d\n",x,y);	□ !=
if (a > 0) {	15. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu	
return f(a - 1) + 1;	du struct, on écrit :	19. Le langage C est un langage
return 4;	\square struct date_s afficher_date(struct date_s	d);
}	☐ void afficher_date(struct date_s d);	\square composé
Alors l'expression f(1) prendra la valeur :	☐ int afficher_date(date_s d);	□ interprété
\Box 0	☐ void afficher_date(date_s d);	□ lu, écrit, parlé
□ 5 	16. Si x est une variable réelle (de type double) alors $x = 3/2$ lui affecte la valeur :	20. Après exécution jusqu'à la ligne 15 du programme C :
\Box 4	\square 0.5	10
□ 1	□ 1.5	11 int main() {
13. Pour déclarer une fonction exposant qui prend en ar-	□ 1	12 int x = 5; 13
gument un réel x et un entier positif n et renvoie la valeur de x^n on écrit :	\Box 0	14 x = 3 * x + 1;
	17. Sur un ordinateur avec un seul processeur, habituelle-	15
□ exposant(double x, int n, int r);	ment les processus sont exécutés :	16
□ void exposant(double x^n);	☐ chacun son tour, après que le processus précédent	17 }
\square double exposant(double x, int n);	a terminé	□ le programme affiche ****
\Box int exposant(double n, int x);	\square en parallèle, chacun dans un registre	\square le programme affiche x
14. Soient deux variables entières x et y initialisées à 4 et	\Box tous ensemble	\Box la variable x vaut $-\frac{1}{2}$
5 respectivement. L'affichage $x=4$ et $y=5$ est obtenu avec la commande :	\Box tour à tour, un petit peu à chaque fois	□ la variable x vaut 16

т		-1
	acence	

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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 □ le terminal affiche 8 \square la case mémoire 8 contiendra 0 \square le bus explose □ la case mémoire 8 contiendra 16 2. 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 3. Après exécution jusqu'à la ligne 15 du programme C: int main() { int x = 5; 11 12 int y; 13 14 y = x;15 16 17 \square la variable x vaut 0 \square la variable x vaut 5 et la variable y vaut 0 ☐ le programme affiche "Faux"

 \square la variable y vaut 5

```
4. Une variable booléenne est un variable :
     \square 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)
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
     \square il risque d'afficher bonjour à la place de coucou
6. Un bit est:
    \square un chiffre binaire (0 ou 1)
                                                             1
    □ l'instruction qui met fin à un programme
    □ la longueur d'un mot mémoire
     \square un battement d'horloge processeur
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 :
     \square 8
     \Box 0
     \Box 4
     \square 16
```

Cuu	•
8.	Sous unix (ou linux), la commande 1s permet de :
	\square compiler un programme
	\square voir des clips musicaux
	□ afficher la liste de fichiers contenus dans un répertoire
	\square afficher le contenu d'un fichier texte
9.	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 une accolade en trop
	\square un point-virgule manquant
	\square une accolade manquante
10.	$L'ordonnancement\ par\ tourniquet\ permet:$
	\Box 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
	\Box d'afficher des ronds colorés à l'écran
11.	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.exe -Wall -o prog.c
	☐ gcc prog.c -o -Wall prog.exe
12.	Un enregistrement permet de grouper plusieurs valeurs
	dans:
	\square ses champs
	\square ses blocs
	□ ses cases
	\square ses chants
13.	Le type des réels en C est :
	\square double
	\square int
	\square real
	□ char

14.	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";
15.	Vous utilisez une boucle while quand:
	\square vous n'avez pas déclaré de fonction
	\Box l'incrément de la variable de boucle n'est pas 1
	\square vous avez déjà fait un for dans le même pro-
	gramme principal
	\Box vous ne connaissez pas le nombre d'itérations de
	la boucle à l'avance
16.	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();
	\square void saisie_utilisateur(char c);
	\square void saisie_utilisateur(int n);
17.	Le langage C est un langage
	\square compilé
	\square composé
	□ interprété
	\square lu, écrit, parlé

```
18. 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
19. 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
     □ (a<=n) && (n<=b)
     \square (n<=a) && (n<=b)
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);
   La valeur de somme affichée est :
     \square 20
     \Box 16
     \Box 6
```

 \square 3

т		-1
	100000	- 1
	acence	

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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

```
    Après exécution du programme :
    lecture 8 r0
    valeur 3 r1
```

3 mult r1 r0 4 valeur 1 r2 5 add r2 r0

6 ecriture r0 8

7 stop 8 5

 \square la case mémoire 8 contiendra 0

 \Box le bus explose

 \square le terminal affiche 8

 \Box la case mémoire 8 contiendra 16

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)
```

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:
```

□ b
□ A
□ B

```
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. 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 | | B)
    \square !(!A || B) == (A && !B)
    \square (A == TRUE) && (B == TRUE)
6. 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 ****
    □ la variable x vaut 16
```

 \square le programme affiche x

 \square la variable x vaut $-\frac{1}{2}$

```
7. Un registre du processeur est :
      \square 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
      un composant qui contient la liste des fichiers du
        système
 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 n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
 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(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);
10. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     □ ouvrir un fichier texte
      □ changer de répertoire courant
      \square créer un fichier texte
11. Le langage C est un langage
     □ composé
     □ lu, écrit, parlé
     □ interprété
      □ compilé
12. 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
      ☐ #define taille = N
```

```
15. Quel est le problème d'un programme comportant les
13. Soit un programme contenant les lignes suivantes :
                                                                                                                           18. Le code suivant :
                                                                 lignes suivantes?
     int i = 0;
                                                                                                                                int i:
                                                                 while (1)
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
                                                                   printf("coucou\n");
                                                                                                                                     printf("%d ", i);
         for (j = 0; j < 5; j = j + 1)
                                                                                                                               printf("\n");
                                                                   \square il ne compile pas
                                                                                                                              affichera:
                                                                   \square il n'affiche rien
                                                                                                                                 \square 8 2
                                                                   \square il comporte une boucle infinie
     printf("j = %d\n", j);
                                                                                                                                 \square 8 6 4 2
                                                                   □ il risque d'afficher bonjour à la place de coucou
                                                                                                                                \Box 02468
                                                             16. Pour déclarer une procédure afficher_menu sans ar-
    qu'est ce qui sera affiché par ce printf?
                                                                                                                                \Box 8 6 4 2 0
                                                                 gument et qui ne renvoie rien on utilise:
      \Box j = 4
                                                                                                                           19. Le bus système sert à :
                                                                   ☐ int afficher_menu();
      \Box j = 0
                                                                   □ void afficher_menu();
      \Box j = 5
                                                                                                                                    cesseur
                                                                   ☐ char afficher_menu(printf("menu"));
      \Box j = %d
                                                                   ☐ int afficher_menu(int char);
14. Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                                    cesseur et mémoire
                                                                   ☐ double afficher_menu();
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
                                                             17. Pour déclarer une fonction exposant qui prend en ar-
                                                                 gument un réel x et un entier positif n et renvoie la
    est l'ordre :
                                                                 valeur de x^n on écrit :
      □ alphabétique
                                                                   \square int exposant(double n, int x);
                                                                                                                                 \Box char
      \square dans lequel vous avez déclaré ces fonction
                                                                   \square void exposant(double x^n);
                                                                                                                                 \square int
      \square dans lequel ces fonctions sont appelées dans le
                                                                   \square exposant(double x, int n, int r);
         main
                                                                                                                                 ☐ double
      \square un ordre quelconque
                                                                   \square double exposant(double x, int n);
                                                                                                                                 \square real
```

```
for (i = 8; i > 0; i = i - 2)
      □ transporter les processus du tourniquet au pro-
      \square Transférer des données et intructions entre pro-
      ☐ Arriver à l'heure en cours
     ☐ Écrire des données sur le dique dur
20. Le type des réels en C est :
```

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 struct toto_s = {3, -1e10}; \square int toto.n = 3: □ struct toto s toto: \square toto_s struct z = {3, 0.5}; \square toto_s n, x; 2. 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 □ 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 □ certains programmes sont de vrais plats de spaghetti 3. 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); ☐ int saisie_utilisateur();

□ void saisie_utilisateur(int n);

```
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. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
  mande:
     ☐ yppasswd
    □ mkdir TP4
    ☐ kwrite TP4
    □ new TP4
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 carre(2);
    \square n = carre(int n);
    \square n = carre(n);
    \square int n = carre();
7. 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
```

```
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
 9. Un bit est:
      \square 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
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. 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 un désaccord entre la déclaration et la définition
         d'une fonction
      ☐ une directive préprocesseur #include manquante
      une fonction déclarée mais non définie
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 6
      \Box 0
      \square 10
      \square 15
```

```
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 :
      \Box 5
      \Box 0
      \square 7
14. Laquelle de ces écritures correspond à la déclaration
    d'une variable de type caractère en langage C?
      \square int char;
      ☐ char "c";
      \Box char c;
      □ char 'c';
15. 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
       }
      \Boxle programme affiche "Faux"
      \square la variable y vaut 5
      \Box la variable x vaut 0
      \square la variable x vaut 5 et la variable y vaut 0
```

```
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 = 5
     \Box j = %d
17. 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 1 0 1 0 1 0 1
     \Box 0 1 2 0 1 2
```

 \Box 0 0 0 1 1 1

```
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 :
      □ (a<=n) && (n<=b)
      \square (n<=a) && (n<=b)
      \square (a<n) || (n>b)
      □ a<=n<=b
19. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée
      □ l'avoir déclarée et définie
      ☐ l'avoir définie
      □ avoir défini une constante symbolique de la taille
         de cette fonction
20. L'écriture 111 en binaire correspond au nombre natu-
```

rel:

□ 8

 \Box 7

 \square 3

□ 111

Éléments d'informatique – contrôle continue

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

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
```

5

□ le terminal affiche 8

```
\Box la case mémoire 8 contiendra 0
```

 $\Box\,$ le bus explose

 \Box la case mémoire 8 contiendra 16

2. Le langage C est un langage

```
\Box\,lu, écrit, parlé
```

 \Box composé

 $\Box \ \ {\rm compil\acute{e}}$

 \square interprété

3. 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);
    }
}</pre>
```

qu'est ce qui sera affiché?

```
□ 1 2 1 2 3□ 0 1 2 0 1 2□ 0 1 0 1 0 1 0 1□ 0 0 0 1 1 1
```

```
4. Soit la fonction {\sf 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

5. Le bus système sert à :

☐ Arriver à l'heure en cours

☐ Transférer des données et intructions entre processeur et mémoire

 \Box transporter les processus du tourniquet au processeur

6. Vous utilisez une boucle while quand :

\square vous ne connaissez pas	le nombre	d'itérations	de
la boucle à l'avance			

 $\square\,$ vous n'avez pas déclaré de fonction

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

□ vous avez déjà fait un for dans le même programme principal

7. Quels calculs peut-on programmer en programmation 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 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 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:

 \Box dans lequel ces fonctions sont appelées dans le main

 \square dans lequel vous avez déclaré ces fonction

□ alphabétique□ un ordre quelconque

9. 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 :

 \Box A

☐ ABCDEF

 \square ccccc

 \square i

10. Au début de la fonction ${\tt main}$ () on place le code :

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

 $Alors\ l'affichage\ sera:$

□b

□ B□ A

 \Box C

11. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :

```
\square n = mccarthy();

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

 \square int mccarthy(int 2);

 \square x = mccarthy(n);

```
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:
      \square a = b
      □ a == b
      \square a{n, x} == b{n, x}
      \square (a.n == b.n) && (a.x == b.x)
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 int n = pgcd();
      \square n = pgcd(n, 3);
14. 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
      \square il risque d'afficher bonjour à la place de coucou
```

```
15. Si racine est une fonction prenant en entrée un réel
                                                                  □ l'analyse harmonique
    et renvoyant la racine carrée de cet réel, et que x est
                                                                 \square l'analyse des entrées clavier
    une variable réelle définie et initialisée, il est incorrect
                                                                  \square l'édition de liens
    d'écrire :
     \square x = racine(x * x) - racine(x);
                                                            18. Pour compiler un programme prog.c, on utilise la
                                                                ligne de commande :
      \square x = racine(racine(x)*racine(x));
                                                                  ☐ gcc -Wall prog.c -o prog.exe
      \square x - 1 = racine(x);
                                                                  ☐ gcc -Wall prog.exe -o prog.c
      \square x = racine(2/3);
                                                                 ☐ gcc prog.exe -Wall -o prog.c
16. Soit un programme contenant les lignes suivantes :
                                                                 ☐ gcc prog.c -o -Wall prog.exe
     int i = 0;
     int j = 0;
                                                            19. Après exécution jusqu'à la ligne 15 du programme C:
     for (i = 0: i < 0: i = i + 1)
                                                                   int main() {
                                                             11
                                                                        int x = 5;
         for (j = 0; j < 5; j = j + 1)
                                                              12
                                                                        int y;
                                                              13
                                                              14
                                                                        y = x;
                                                              15
                                                              16
     printf("j = %d\n", j);
                                                              17
                                                                  }
     }
                                                                  \square la variable y vaut 5
                                                                  □ le programme affiche "Faux"
    qu'est ce qui sera affiché?
                                                                  \Box la variable x vaut 0
      \Box j = 5
                                                                  \square la variable x vaut 5 et la variable y vaut 0
      \Box i = 0
     \Box j = 4
                                                            20. Laquelle de ces écritures correspond à la déclaration
                                                                d'une variable de type caractère en langage C?
      \Box j = %d
17. Quelle étape de la compilation vient d'échouer lors-
                                                                  ☐ char "c";
    qu'on a un message comme celui-ci :
                                                                 ☐ int char;
    Undefined symbols : "_prinft" ou
                                                                  \Box char c;
    référence indéfinie vers « prinft »
                                                                  □ char 'c';
      □ l'analyse sémantique
```

16

}

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 :
     \square 15
     \Box 6
     \Box 0
     \Box 10
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 1
    \Box 5
    \Box 0
     \Box 4
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
```

```
\square le terminal affiche x = 5
     ☐ le terminal affiche "Faux"
     \square le terminal affiche x = 2
     \square le terminal affiche 5
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 == TRUE) && (B == TRUE)
     □ A && B
     \square (!A || B)
     \square !(!A | | B) == (A && !B)
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 :
     \square 8
     \Box 16
     \Box 4
     \square 0
6. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
    {
        printf("%d ", i);
    printf("\n");
   affichera:
     \Box 02468
    \Box 0246
     \Box 0 1 2 3 4 5 6
```

 \Box 0 1 2 3 4 5 6 7

```
7. 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);
 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:
     \Box C
     □В
     □ b
     □ A
9. 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>
10. 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);
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 :	15. Dans la commande gcc, l'option -Wall signifie :	\square Mineur
\Box 4	□ qu'il faut indenter le fichier source	Majeur
\square 3	□ qu'il faut lancer un déboggueur	\square rien
\Box 0	☐ qu'on veut changer alétoirement de fond d'écran	\square Majeur
12. Un enregistrement permet de grouper plusieurs valeurs	\square que l'on veut voir tous les avertissements	☐ Mineur
dans: \Box ses chants	16. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :	19. Soit le programme principal suivant :
□ ses blocs	int factorielle(double n);	int main()
□ ses cases	☐ int factorielle();	<pre>int a = 3;</pre>
\square ses champs	☐ struct int factorielle(int n);	int b = 5;
13. Un programme en langage C doit comporter une et une seule définition de la fonction :	☐ int factorielle(int x);	<pre>printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b) return EXIT_SUCCESS;</pre>
	17. Vous avez déclaré préalablement un ensemble de fonc-	}
init	tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions	appelant la fonction f ainsi définie :
\square main	est l'ordre :	<pre>int f(int a, int b)</pre>
\square include	□ alphabétique	{
14. Pour l'extrait de programme suivant :	\Box dans lequel ces fonctions sont appelées dans le	a = a + b; return a;
int i = 0;	main	}
int j = 0;	\Box dans lequel vous avez déclaré ces fonction	L'affichame dang la main est la suivent.
for (i = 0; i < 3; i = i + 1)	\square un ordre quelconque	L'affichage dans le main est le suivant :
for (j = 0; j < 2; j = j + 1)	18. Le code suivant :	$\Box f(a,b)=8, a=8, b=5$
{	int age = 18;	\Box f(a,b)=8, a=3, b=5
<pre>printf("%d ", i);</pre>	if (age < 18)	\Box f(3,5)=8, a=3, b=5
}	{	\Box f(a,b)=13, a=8, b=5
<pre>} printf("\n");</pre>	<pre>printf("Mineur\n"); }</pre>	20. Après la déclaration : int mccarthy(int n);, il est
-	else	correct d'écrire :
qu'est ce qui sera affiché?	{	\square n = mccarthy();
	<pre>printf("Majeur\n");</pre>	\square n = mccarthy(p, q);
	}	☐ int mccarthy(int 2);
	, m	$\square x = mccarthy(n);$
\Box 0 1 0 1 0 1	affichera:	Z A mood ony (n),

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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; 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 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 \Box a = b \square (a.n == b.n) && (a.x == b.x) \square a{n, x} == b{n, x} 3. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : \square int factorielle(int x); □ struct int factorielle(int n); ☐ int factorielle(); ☐ int factorielle(double n); 4. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise: ☐ double afficher_menu(); □ void afficher menu(): ☐ int afficher_menu(); ☐ char afficher_menu(printf("menu")); ☐ int afficher menu(int char):

```
5. 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 j = 0
     \Box j = %d
     \Box j = 5
     \Box j = 4
6. 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
7. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     □ begin
     \square include
     \square main
     \square init
8. Une variable booléenne est un variable :
     \square jamais nulle
     □ à laquelle une valeur vient d'être affectée
     ☐ réelle positive
     \square qui est vraie ou fausse
```

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

```
9. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses blocs
      \square ses chants
      \square ses champs
      □ ses cases
10. 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);
     \square n = factorielle();
      ☐ printf("%d", factorielle(n));
11. 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;
      \square toto_s struct z = {3, 0.5};
      \square int toto.n = 3;
      □ struct toto_s toto;
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 0
      \Box 4
```

```
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
     \Box 0 0 1 1 2 2 3
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 3 0 1 2
14. Le langage C est un langage
      □ interprété
     □ compilé
     □ lu, écrit, parlé
      □ composé
15. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square n = mccarthy();
     \square n = mccarthy(p, q);
     \square x = mccarthy(n);
      \square int mccarthy(int 2);
```

```
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 :
     \square printf("x=%d et y=%d\n",x y);
     \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");
17. Le code suivant :
    int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     \square rien
     □ Mineur
        Majeur
     □ Majeur
     □ Mineur
18. 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 8 6 4 2
     \Box 02468
     \square 8 2
19. 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<=b
     □ (a<=n) && (n<=b)
     \square (a<n) || (n>b)
20. Un registre du processeur est :
     □ un composant qui contient la liste des fichiers du
        système
     \Box une unité de calcul spécialisée de l'ordinateur
     \square une gamme de fréquence de fonctionnement du
         processeur
     □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
```

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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 comporte une boucle infinie
    □ il n'affiche rien
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il ne compile pas
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. Le code suivant :
    int age = 20;
    if (age < 18)
    ₹
        printf("Mineur\n");
    printf("Majeur\n");
  affichera:
     \square rien
     □ Majeur
     ☐ Mineur
     ☐ Mineur
```

Majeur

```
4. 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 struct toto_s = {3, -1e10};
    \square toto_s n, x;
    \square int toto.n = 3;
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 0
    \square 8 2
    \Box 8 6 4 2
6. Pour compiler un programme prog.c, on utilise la
  ligne de commande:
    ☐ gcc prog.exe -Wall -o prog.c
    ☐ gcc -Wall prog.c -o prog.exe
    ☐ gcc -Wall prog.exe -o prog.c
    ☐ gcc prog.c -o -Wall prog.exe
7. Laquelle des analyses suivantes ne fait pas partie des
  étapes de la compilation :
    \square analyse lexicale
    □ analyse sémantique
```

 \square analyse harmonique

 \square analyse syntaxique

```
8. 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 "Faux"
      \square le terminal affiche x = 5
      \square le terminal affiche 5
      \square le terminal affiche x = 2
 9. 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 sera détectée au cours
        de l'édition de lien
      \square ne comportent aucune erreur
      □ comportent une erreur qui ne sera pas détectée
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();
     □ void saisie_utilisateur(char c);
     □ void saisie_utilisateur(int n);
      □ saisie_utilisateur(scanf(%d));
11. 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
      □ un document illisible pour les humains
      \square un fichier texte qui sera traduit en instructions
        processeur
```

12. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : ☐ int factorielle(int x);	16. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme?	5 add r2 r0 6 ecriture r0 8 7 stop 8 5
☐ int factorielle(double n); ☐ int factorielle(); ☐ struct int factorielle(int n); 13. 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 ☐ un désaccord entre la déclaration et la définition d'une fonction	□ 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 17. Après exécution jusqu'à la ligne 15 du programme C: 10 int main() { 11 int x = 5; 12 int y; 13	□ la case mémoire 8 contiendra 16 □ le terminal affiche 8 □ le bus explose □ la case mémoire 8 contiendra 0 19. Si carre est une fonction prenant en entrée un er tier et renvoyant le carré de cet entier, et que n es une variable entière définie et initialisée, il est correc d'écrire :
 □ une fonction appelée avant sa déclaration 14. Le bus système sert à : □ Écrire des données sur le dique dur □ Transférer des données et intructions entre processeur et mémoire □ transporter les processus du tourniquet au processeur □ Arriver à l'heure en cours 	13 14	 □ n = carre(n); □ int n = carre(); □ int carre(2); □ n = carre(int n); 20. Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordr dans lequel vous devez maintenant définir ces fonctions.
<pre>15. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire : □ n = factorielle(); □ int factorielle(int 2); □ printf("%d", factorielle(n)); □ n = factorielle(p, q);</pre>	□ la variable x vaut 5 et la variable y vaut 0 18. Après exécution du programme : 1 lecture 8 r0 2 valeur 3 r1 3 mult r1 r0 4 valeur 1 r2	est l'ordre : □ alphabétique □ un ordre quelconque □ dans lequel vous avez déclaré ces fonction □ dans lequel ces fonctions sont appelées dans main

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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); □ void afficher_menu(); ☐ double afficher_menu(); ☐ int afficher_menu(); ☐ char afficher_menu(printf("menu")); 2. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <studio.h> ☐ #appart <stdlib.h> ☐ #include <studlib.h> ☐ #include <stdio.h> 3. Le code suivant : int i: for $(i = 4; i \ge 0; i = i - 1)$ printf("%d ", i); printf("\n"); affichera: $\Box 01234$ \square 1 2 3 4 \square 4 3 2 1 \Box 4 3 2 1 0 4. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3. \square #define taille = 3 \square #define N = 3 \square #define taille = N

□ #define N 3

```
5. 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 directive préprocesseur #include manquante
     \square une fonction déclarée mais non définie
     □ un désaccord entre la déclaration et la définition
        d'une fonction
6. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     □ analyse syntaxique
     \square analyse lexicale
     □ analyse sémantique
     \square analyse harmonique
7. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square x = mccarthy(n);
    \square n = mccarthy();
     \square int mccarthy(int 2);
     \square n = mccarthy(p, q);
8. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
     \square int loop n;
    □ loop i;
    \square int k:
     □ int %d:
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(a,b)=8, a=3, b=5
      \Box f(3,5)=8, a=3, b=5
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 variable non déclarée
      \square une faute de frappe dans un appel de fonction
      □ un caractère interdit en C
11. 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
12. 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
      \square 1
13. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses blocs
      \square ses champs
      \square ses cases
      □ ses chants
```

```
□ il risque d'afficher bonjour à la place de coucou
14. Après exécution du programme :
                                                                                                                              }
                                                                                                                              return 7;
                                                                 □ il n'affiche rien
       lecture 8 r0
       valeur 3 r1
                                                                 \square il ne compile pas
       mult r1 r0
                                                                                                                           Alors l'expression g(0) prendra la valeur :
                                                                 \square il comporte une boucle infinie
  4
       valeur 1 r2
                                                                                                                             \Box 0
       add r2 r0
                                                           17. Pour déclarer une fonction saisie_utilisateur qui
  6
       ecriture r0 8
                                                                demande à l'utilisateur d'entrer un entier au clavier et
                                                                                                                             \Box 5
       stop
                                                               renvoie cet entier on écrit :
                                                                                                                             \Box 7
  8
       5
                                                                 □ void saisie_utilisateur(int n);
                                                                                                                       20. Soit un programme contenant les lignes suivantes :
     □ la case mémoire 8 contiendra 0
                                                                 □ void saisie_utilisateur(char c);
     \square le bus explose
                                                                 ☐ int saisie_utilisateur();
                                                                                                                             int i = 0;
     □ le terminal affiche 8
                                                                                                                             int j = 0;
                                                                 ☐ saisie_utilisateur(scanf(%d));
      □ la case mémoire 8 contiendra 16
                                                                                                                            for (i = 0; i < 2; i = i + 1)
                                                            18. Une variable booléenne est un variable :
15. Si cette erreur apparaît à la compilation :
                                                                  □ qui est vraie ou fausse
                                                                                                                                 for (j = 0; j < 3; j = j + 1)
   error: expected ';' before '}' token que doit-
                                                                 \Box à la
quelle une valeur vient d'être affectée
   on chercher dans le programme?
                                                                                                                                      printf("%d ", i);
                                                                 □ NaN (not a number, qui n'est pas un nombre)
     \Box une accolade manquante
                                                                                                                                 }
                                                                 □ jamais nulle
     \square un point-virgule manquant
                                                                                                                             }
                                                                 ☐ réelle positive
     \square une accolade en trop
     \square un point-virgule en trop
                                                           19. Soit la fonction g définie par :
                                                                                                                           qu'est ce qui sera affiché?
16. Quel est le problème d'un programme comportant les
                                                                int g(int a)
                                                                                                                             \Box 0 1 2 0 1 2
   lignes suivantes?
                                                               {
                                                                                                                             \Box 1 2 1 2 3
   while (1)
                                                                  printf("a = \n", %d);
                                                                  if (1 > 0)
                                                                                                                             \Box 0 0 0 1 1 1
      printf("coucou\n");
                                                                  {
                                                                                                                              \Box 0 1 0 1 0 1 0 1
                                                                    return 5;
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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 ☐ une fonction appelée avant sa déclaration \square une fonction déclarée mais non définie ☐ une directive préprocesseur #include manquante 2. 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 void exposant(double x^n); \square double exposant(double x, int n); \square int exposant(double n, int x); \square exposant(double x, int n, int r); 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 : \square 3 \Box 4 \square 0 4. Sous unix (ou linux), la commande 1s permet de : \Box afficher la liste de fichiers contenus dans un répertoire □ compiler un programme □ voir des clips musicaux \square afficher le contenu d'un fichier texte

```
5. 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
6. Quel est l'opérateur de différence en C :
     \Box !
    □ !=
    \square \neq
     □ <>
7. Sur unix (ou linux), la commande mkdir permet de :
    □ créer un répertoire
    □ ouvrir un fichier texte
    □ créer un fichier texte
     □ changer de répertoire courant
8. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
    ☐ Mineur
        Majeur
    \square rien
     □ Majeur
```

☐ Mineur

```
9. Le code suivant :
     int age = 15;
     if (age < 18)
          printf("Mineur\n");
     }
     else
          printf("Majeur\n");
    affichera:
      □ Majeur
      □ Mineur
         Majeur
      ☐ Mineur
      \square rien
10. Une variable booléenne est un variable :
      ☐ réelle positive
      □ à laquelle une valeur vient d'être affectée
      □ NaN (not a number, qui n'est pas un nombre)
      \square qui est vraie ou fausse
      \square jamais nulle
11. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses chants
      □ ses cases
      \square ses blocs
      \square ses champs
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 (n<=a) && (n<=b)
      □ a<=n<=b
      ☐ (a<=n) && (n<=b)
      \square (a<n) || (n>b)
13. 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)
14. 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 6
      \square 3
      \Box 16
```

```
15. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
     \square int factorielle(int x);
     ☐ struct int factorielle(int n);
     ☐ int factorielle(double n);
     ☐ int factorielle();
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 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
17. Le code suivant :
     int i;
    for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
    }
                                                              ☐ double afficher_menu();
    printf("\n");
```

affichera:
\square 1 2 3 4
\square 4 3 2 1
$\Box \ 4\ 3\ 2\ 1\ 0$
$\Box \ 0 \ 1 \ 2 \ 3 \ 4$
18. L'écriture <u>111</u> en binaire correspond au nombre naturel :
\Box 7
□ 111
□ 8
19. Le bus système sert à :
\Box Écrire des données sur le dique dur
\Box transporter les processus du tourniquet au processeur
☐ Transférer des données et intructions entre processeur et mémoire
\Box Arriver à l'heure en cours
20. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :
☐ char afficher_menu(printf("menu"));
☐ void afficher_menu();
☐ int afficher_menu(int char);
☐ int afficher menu():

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

9. Quelle étape de la compilation vient d'échouer lors-

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 □ créer un répertoire □ changer de répertoire courant □ créer un fichier texte 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 \square (a<=n) && (n<=b) \square (n<=a) && (n<=b) 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. 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 □ alphabétique

□ dans lequel vous avez déclaré ces fonction

main

□ dans lequel ces fonctions sont appelées dans le

```
5. 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 01234
    \square 4 3 2 1
    \Box 1234
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 :
    \square 7
    \Box 5
     \Box 0
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
    \square ses chants
    \square ses champs
    \square ses blocs
     □ ses cases
8. 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);
```

```
qu'on a un message comme celui-ci :
                                                         Undefined symbols : "_prinft" ou
                                                         référence indéfinie vers « prinft »
                                                            □ l'analyse des entrées clavier
                                                           □ l'édition de liens
                                                           ☐ l'analyse sémantique
                                                           ☐ l'analyse harmonique
                                                      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);
                                                         La valeur de somme affichée est :
                                                           \square 20
                                                           \Box 6
                                                           \square 16
                                                           \square 3
                                                      11. 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
                                                            \square la variable v vaut 5
                                                           \Box la variable x vaut 0
                                                            ☐ le programme affiche "Faux"
□ struct date_s afficher_date(struct date_s d);
```

```
12. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
    }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     □ Majeur
     ☐ Mineur
     □ rien
     ☐ Mineur
        Majeur
13. Vous utilisez une boucle while quand :
     \square 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 pro-
        gramme principal
     \square vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
14. Le code suivant :
     int i;
    for (i = 0; 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 0 1 2 3
15. 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
      \Box qu'il faut lancer un déboggueur
      ☐ qu'il faut indenter le fichier source
16. 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 :
      \Box x - 1 = racine(x):
      \square x = racine(2/3);
      \square x = racine(x * x) - racine(x);
      \square x = racine(racine(x)*racine(x));
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 :
      □В
      □ b
      □ A
      \Box 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. 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
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 pgcd(2);
```

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

 \square int n = pgcd();

т		-1
	acence	- 1

int i;

int j;

for(i=4;i>0;i=i-1)

for(j=i;j<6;j=j+1)

printf("*");

Éléments d'informatique – contrôle continue

Prénom:	Nom:
	NOIII .
N° etu :	
11 004.	

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 \square 3 \Box 5 \Box 4 2. 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(int n); □ void saisie_utilisateur(char c); 3. 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 n = pgcd(int p, int q); \square int pgcd(2); \square int n = pgcd(); 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 int carre(2); \square n = carre(int n): \square n = carre(n); 5. Pour l'extrait de programme suivant :

```
}
      printf(" ");
    }
   qu'est ce qui sera affiché?
         ***** *** ***
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)
     □ A && B
     \square (A == TRUE) && (B == TRUE)
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 :
     \square des processus
     □ certaines données de la mémoire de travail
     \square en temps d'accès
     □ les fichiers du disque
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 void pgcd(int x, int y);
    \Box int pgcd(int x, int x);
    \Box int pgcd(int y, int x);
    \Box int pgcd(int x, y);
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=%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);
```

```
10. 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
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 4
      \square 5
      \Box 0
12. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \square 3
      \square 7
      \square 8
```

□ 111

<pre>13. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit- on chercher dans le programme? □ une accolade en trop □ un point-virgule manquant □ un point-virgule en trop □ une accolade manquante 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) {</pre>	☐ créer un répertoire ☐ changer de répertoire courant 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 ☐ il ne compile pas ☐ il risfiche rien ☐ il comporte une boucle infinie	19. 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 □ le bus explose □ la case mémoire 8 contiendra 16
<pre>produit = produit * serie[i]; } printf("produit = %d", produit);</pre>	17. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?□ char "c";	□ le terminal affiche 8 $□$ la case mémoire 8 contiendra 0
La valeur affichée est : B 4 D 16 15. Sur unix (ou linux), la commande mkdir permet de : Créer un fichier texte Ouvrir un fichier texte	☐ int char; ☐ char 'c'; ☐ char c; 18. Un programme en langage C doit comporter une et une seule définition de la fonction : ☐ begin ☐ init ☐ main ☐ include	 20. 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();

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
N° etu :	
n etu.	

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:
     \square 42
     \Box 0
     \Box 6
     \Box 1
2. 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 int n = pgcd();
     \square n = pgcd(n, 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 16
     \square 8
     \Box 4
     \square 0
```

```
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. 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 x, int x);
     \square int pgcd(int y, int x);
6. Un fichier source est:
     □ un document de référence du système
     □ 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
7. 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
```

8.	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);
	☐ saisie_utilisateur(scanf(%d));
	☐ int saisie_utilisateur();
9.	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
	$\hfill\Box$ un point-virgule man quant
	\Box un point-virgule en trop
10.	Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
	☐ #include <stdio.h></stdio.h>
	\square #include <studio.h></studio.h>
	☐ #appart <stdlib.h></stdlib.h>
	\square #include <studlib.h></studlib.h>
11.	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
	□В
	□ A
	□ C
12.	Dans la commande gcc, l'option -Wall signifie :
	\Box qu'on veut changer alétoirement de fond d'écran
	$\hfill\Box$ qu'il faut indenter le fichier source
	$\hfill\Box$ qu'il faut lancer un déboggueur
	$\hfill\Box$ que l'on veut voir tous les avertissements

3. Vous utilisez une boucle while quand:	\Box f(a,b)=8, a=3, b=5	□ comportent une erreur qui sera détectée au cours
□ vous avez déjà fait un for dans le même pro-	\Box f(a,b)=8, a=8, b=5	de l'analyse syntaxique
gramme principal	\Box f(a,b)=13, a=8, b=5	\Box comportent une erreur qui ne sera pas détectée
$\hfill\Box$ l'incrément de la variable de boucle n'est pas 1	16. Soit la fonction g définie par :	$\hfill\Box$ ne comportent aucune erreur
\square vous n'avez pas déclaré de fonction	int g(int a)	□ comportent une erreur qui sera détectée au cours
\Box vous ne connaissez pas le nombre d'itérations de	{ printf("a = \n", %d);	de l'édition de lien
la boucle à l'avance	$ \begin{array}{ll} \text{if } (1 > 0) \end{array} $	19. Soit la fonction f définie par :
4. L'écriture <u>111</u> en binaire correspond au nombre natu-	{	19. Soit is fonction 1 definite par :
rel:	return 5;	int f(int a)
□ 8	}	{
\square 3	return 7;	<pre>printf("a = \n", %d); if (a > 0)</pre>
□ 111	}	{
□ 7	Alors l'expression g(0) prendra la valeur :	return f(a - 1) + 1;
5. Soit le programme principal suivant :	□ 7	}
int main()	\Box 5 \Box 0	return 4;
Int main() {	-	}
int a = 3;	17. Sous unix (ou linux), la commande cd permet de : □ ouvir un bureau partagé (common desktop)	Alors l'expression f(1) prendra la valeur :
int b = 5;	☐ détruire un fichier	\Box 4
printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);	☐ récupérer un programme arrêté avec la commande	\square 5
return EXIT_SUCCESS;	ab	□ 1
,	□ jouer de la musique	
appelant la fonction f ainsi définie :	☐ changer de répertoire courant	
int f(int a, int b)	18. Les lignes	20. Un programme en langage C doit comporter une et un
{ a = a + b;	int i;	seule définition de la fonction :
return a;	int x=0;	\square main
}	for(i=0,i<5,i=i+1)	\Box init
L'affichage dans le main est le suivant :	{	\square begin
☐ f(3,5)=8, a=3, b=5	x=x+1; }	\Box include
□ 1(0,0/-0, d-0, D-0	J	1

т		-1
	100000	- 1
	acence	

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

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 f(a - 1) + 1;
     return 4;
  Alors l'expression f(1) prendra la valeur :
     \Box 1
     \Box 5
    \Box 0
     \Box 4
2. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
        printf("Majeur\n");
    }
  affichera:
    □ rien
    □ Majeur
     ☐ Mineur
     □ Mineur
       Majeur
```

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 vous avez déclaré ces fonction
```

```
□ dans lequel ces fonctions sont appelées dans le
        main
     \square un ordre quelconque
     □ alphabétique
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(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
5. 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
     ☐ un désaccord entre la déclaration et la définition
        d'une fonction
     □ une fonction déclarée mais non définie
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 carre(2);
    \square int n = carre();
     \square n = carre(int n);
```

 \square n = carre(n);

```
7. Le code suivant :
     int age = 15;
     if (age < 18)
    {
         printf("Mineur\n");
    else
    {
          printf("Majeur\n");
   affichera:
     □ Mineur
        Majeur
     □ Mineur
     \square rien
     □ Majeur
 8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 8
     \Box 7
     \square 3
     □ 111
 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. 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);
     \Box x = racine(x * x) - racine(x);
```

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

11. Le langage C est un langage	affichera:	18. L'écriture <u>101</u> en binaire correspond au nombre nate
\square composé	$\Box \ 8 \ 6 \ 4 \ 2 \ 0$	rel:
□ lu, écrit, parlé	\square 8 6 4 2	
\Box interprété	\square 8 2	
\square compilé	$\Box \ 0\ 2\ 4\ 6\ 8$	
12. Le bus système sert à :	15. Sur unix (ou linux), la commande mkdir permet de :	
\Box Écrire des données sur le dique dur	□ changer de répertoire courant	19. Si x est une variable réelle (de type double) alor
\Box Transférer des données et intructions entre pro-	-	x = 3/2 lui affecte la valeur :
cesseur et mémoire	□ créer un répertoire	
□ transporter les processus du tourniquet au pro-	□ créer un fichier texte	
cesseur ☐ Arriver à l'heure en cours	\square ouvrir un fichier texte	
	16. Un bit est:	
13. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs	☐ l'instruction qui met fin à un programme	20. Après exécution jusqu'à la ligne 15 du programme C
passés en arguments on écrit :	□ la longueur d'un mot mémoire	10
□ void pgcd(int x, int y);	\square un battement d'horloge processeur	11 int main() {
☐ int pgcd(int x, y);	\square un chiffre binaire (0 ou 1)	12 int x = 5;
☐ int pgcd(int x, int x);	17. Si cet avertissement apparaît à la compilation :	14 x = 3 * x + 1;
☐ int pgcd(int y, int x);	warning: implicit declaration of function 'max	
14. Le code suivant :	, que doit-on chercher dans le programme?	16 17 }
int i;	☐ une directive préprocesseur #include manquante	
for $(i = 8; i > 0; i = i - 2)$	$\hfill\Box$ une fonction déclarée mais non définie	\square le programme affiche x
{	☐ un désaccord entre la déclaration et la définition	☐ le programme affiche ****
<pre>printf("%d ", i); }</pre>	d'une fonction	\square la variable x vaut $-\frac{1}{2}$
<pre>printf("\n");</pre>	\square une fonction appelée avant sa déclaration	☐ la variable x vaut 16
- -		•

т		-1
	acence	

 \Box 4

Prénom:	Nom:
N° etu:	1,0111
n etu:	

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=%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);
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(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
3. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
     ☐ int afficher_menu();
    ☐ double afficher_menu();
    ☐ char afficher_menu(printf("menu"));
```

□ void afficher_menu();

☐ int afficher_menu(int char);

```
4. L'écriture 111 en binaire correspond au nombre natu-
  rel:
     \Box 7
     \square 8
     \square 3
     □ 111
5. Une variable booléenne est un variable :
     ☐ réelle positive
     □ 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
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 0
     \Box 1
     \Box 4
     \square 5
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
```

etu	:	•
8.	Une	segmentation fault est une erreur qui survient que :
		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
		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
9.	L'écrel :	riture $\underline{101}$ en binaire correspond au nombre natu-
		4
		3
		5
		101
10.	warr	et avertissement apparaît à la compilation : ning: implicit declaration of function 'max' e 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
11.		s unix (ou linux), pour créer un répertoire TP4 s le répertoire courant on peut utiliser la comde :
		mkdir TP4
		new TP4
	П	unnaggud

☐ kwrite TP4

```
15. Dans la commande gcc, l'option -Wall signifie :
12. Le code suivant :
                                                                   ☐ que l'on veut voir tous les avertissements
     int i;
     for (i = 4; i > 0; i = i - 1)
                                                                   □ qu'il faut indenter le fichier source
                                                                   □ qu'on veut changer alétoirement de fond d'écran
         printf("%d ", i);
                                                                   □ qu'il faut lancer un déboggueur
     }
     printf("\n");
                                                             16. Le code suivant :
    affichera:
                                                                   int i;
                                                                  for (i = 0; i < 5; i = i + 1)
      \Box 01234
     \square 4 3 2 1
                                                                       printf("%d ", i);
      \square 0 1 2 3
                                                                  printf("\n");
      \Box 4 3 2 1 0
                                                                 affichera:
13. Pour déclarer un tableau d'entiers de taille 5, on peut
                                                                   \square 4 3 2 1
    utiliser l'instruction
                                                                   \square 0 1 2 3
     \Box int tab[] = 5;
                                                                   \Box \ 0\ 1\ 2\ 3\ 4
     □ char tableau[5];
                                                                   \Box \ 4\ 3\ 2\ 1\ 0
     ☐ int toto[taille=5];
                                                             17. Le code suivant :
     \square int toto[5];
                                                                   int age = 20;
     ☐ int[] new tableau(5);
                                                                  if (age < 18)
14. Vous avez déclaré préalablement un ensemble de fonc-
                                                                   {
    tions utilisées par votre programme principal. L'ordre
                                                                       printf("Mineur\n");
    dans lequel vous devez maintenant définir ces fonctions
                                                                  }
    est l'ordre :
                                                                   else
                                                                   {
      \Box dans lequel vous avez déclaré ces fonction
                                                                       printf("Majeur\n");
      \square dans lequel ces fonctions sont appelées dans le
                                                                  }
         main
     \square un ordre quelconque
                                                                 affichera:
     □ alphabétique
                                                                    ☐ Mineur
```

 \square il n'affiche rien

Prénom: Nom: N° etu :

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)=13, a=8, b=5
     \Box f(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
2. 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 en trop
     ☐ un point-virgule manquant
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 :
     □ en temps d'accès
     \square les fichiers du disque
```

□ certaines données de la mémoire de travail

 \square des processus

```
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=%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);
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:
    \Box (a.n == b.n) && (a.x == b.x)
    \square a{n, x} == b{n, x}
    \Box a = b
    □ a == b
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 vous avez déclaré ces fonction
    □ dans lequel ces fonctions sont appelées dans le
        main
    □ un ordre quelconque
    □ alphabétique
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 n = carre(n);
    \square int n = carre();
```

```
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 1
      \Box 5
      \Box 0
      \Box 4
 9. 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';
10. Le code suivant :
     int somme = 0;
     int i;
     for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
     printf("%d", somme);
    affichera:
      \Box 42
      \square 1
      \Box 0
      \Box 6
11. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
      \square 1.5
      \Box 1
      \Box 0
```

 \square 0.5

2. Vous utilisez une boucle while quand:	15. L'écriture <u>111</u> en binaire correspond au nombre natu-	18. Si le code :
 □ l'incrément de la variable de boucle n'est pas 1 □ vous avez déjà fait un for dans le même programme principal 	rel:	struct toto_s
	□ 7	{
	□ 111	int n;
□ vous n'avez pas déclaré de fonction		double x;
□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance		};
3. Le code suivant :	16. Pour l'extrait de programme suivant :	précède la fonction main(), alors on peut écrire en début de main():
int age = 20;	. ~	☐ int struct toto_s = {3, -1e10};
if (age < 18)	int produit = 0;	
{	int serie[4] = {2, 2, 2, 2};	☐ struct toto_s toto;
<pre>printf("Mineur\n");</pre>	for (i = 0; i < 4; i = i + 1)	\Box toto_s struct z = {3, 0.5};
}	<pre>produit = produit * serie[i];</pre>	☐ toto_s n, x;
else	}	\Box int toto.n = 3;
<pre>{ printf("Majeur\n");</pre>	<pre>printf("produit = %d", produit);</pre>	 19. Pour afficher à l'aide de printf("%d\n",tab[i]);
}	La valeur affichée est :	le contenu d'un tableau de 5 entiers initialisé au
		préalable, on utilise plutôt :
affichera:		☐ for(i=0;i<=5;i=i+1)
□ rien		☐ for(i=0;i<5;i=i+1)
☐ Majeur	\Box 4	☐ for(i=1;i<=5;i=i+1)
☐ Mineur	□ 8	
☐ Mineur	17. L'ordonnancement par tourniquet permet :	☐ for(i=1;i<5;i=i+1)
Majeur	17. L'ordonnancement par tourinquet permet.	20. Sous unix (ou linux), la commande cd permet de :
4. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs	\Box de ne pas perdre de temps avec la commutation de contexte	□ récupérer un programme arrêté avec la commande ab
passés en arguments on écrit :	☐ d'afficher des ronds colorés à l'écran	☐ détruire un fichier
□ void pgcd(int x, int y);	☐ de doubler la mémoire disponible	\Box changer de répertoire courant
☐ int pgcd(int y, int x);	☐ d'entretenir l'illusion que les processus tournent	□ jouer de la musique
☐ int pgcd(int x, y);	en parallèle	
\Box int pgcd(int x, int x);		□ ouvir un bureau partagé (common desktop)

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

1. 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 2. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n"); □ il n'affiche rien □ il comporte une boucle infinie \square il ne compile pas \square il risque d'afficher bonjour à la place de coucou 3. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme? □ une fonction déclarée mais non définie □ un désaccord entre la déclaration et la définition d'une fonction \square une fonction appelée avant sa déclaration ☐ une directive préprocesseur #include manquante 4. 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>

```
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 :
    \Box 1
    \Box 4
    \Box 0
    \Box 5
6. Le code suivant :
    int i:
    for (i = 0; i < 5; i = i + 1)
        printf("%d ", i);
   printf("\n");
   affichera:
    \Box 01234
    \square 0 1 2 3
    \Box 4 3 2 1 0
    \square 4 3 2 1
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
```

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

□ vous n'avez pas déclaré de fonction

```
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
      \square 7
      \Box 0
 9. Sous unix (ou linux), la commande 1s permet de :
      \square afficher la liste de fichiers contenus dans un
         répertoire
      □ compiler un programme
      □ voir des clips musicaux
      □ afficher le contenu d'un fichier texte
10. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \Box 5
      \Box 4
      \square 3
      \square 101
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
      \square a{n, x} == b{n, x}
```

 \square a = b

12. Quel est l'opérateur de différence en C :	16. Soit le programme principal suivant :	18. Sous unix (ou linux), la commande cd permet de :
□ !=	<pre>int main()</pre>	□ changer de répertoire courant
$\Box \Leftrightarrow$	{	☐ détruire un fichier
□ !	int a = 3; int b = 5;	□ ouvir un bureau partagé (common desktop)
□≠	<pre>printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b); return EXIT_SUCCESS;</pre>	□ jouer de la musique
13. Un bit est:		-
□ la longueur d'un mot mémoire	}	☐ récupérer un programme arrêté avec la commande ab
☐ l'instruction qui met fin à un programme	appelant la fonction f ainsi définie :	
un chiffre binaire (0 ou 1)	int f(int a, int b)	19. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :
un battement d'horloge processeur	1 a = a + b:	ment les processus sont executes.
<u> </u>	return a;	\Box tour à tour, un petit peu à chaque fois
14. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-	}	\Box en parallèle, chacun dans un registre
on chercher dans le programme?	L'affichage dans le main est le suivant :	\Box tous ensemble
\Box une accolade en trop	\Box f(3,5)=8, a=3, b=5	☐ chacun son tour, après que le processus précédent
□ un point-virgule en trop	$\Box f(a,b)=8, a=8, b=5$	a terminé
une accolade manquante	☐ f(a,b)=13, a=8, b=5	20 0: 4 4: 4 2: 4 2: 4
un point-virgule manquant	☐ f(a,b)=8, a=3, b=5 17. Si racine est une fonction prenant en entrée un réel	20. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max'
15. Pour déclarer une fonction factorielle qui prend en	et renvoyant la racine carrée de cet réel, et que x est	, que doit-on chercher dans le programme?
argument un entier et renvoie sa factorielle on écrit :	une variable réelle définie et initialisée, il est incorrect	une directive préprocesseur #include manquante
□ struct int factorielle(int n);	<pre>d'écrire :</pre>	☐ une fonction appelée avant sa déclaration
int factorielle();		
☐ int factorielle(int x);		☐ un désaccord entre la déclaration et la définition d'une fonction
☐ int factorielle(double n);		☐ une fonction déclarée mais non définie
☐ Int lactorierie(double n),	\square x = racine(2/3);	une ionetion declaree mais non denine

 \square 0

Éléments d'informatique – contrôle continue

Pré	nom	:	Nom:
N°	etu	:	
		_	
ı	0	т	- 1 C+ 1

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

1. Le bus système sert à : ☐ Transférer des données et intructions entre processeur et mémoire ☐ Écrire des données sur le dique dur □ transporter les processus du tourniquet au processeur ☐ Arriver à l'heure en cours 2. L'écriture 101 en binaire correspond au nombre naturel: \Box 5 \square 3 \Box 4 \square 101 3. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit : ☐ struct int factorielle(int n); \Box int factorielle(int x); ☐ int factorielle(); ☐ int factorielle(double n); 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 4 \Box 5 \Box 1

```
5. 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 1 2 3 4
    \square 4 3 2 1
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 0
    \square 8
    \square 16
    \Box 4
7. 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 4 3 2 1
    \Box 01234
```

 \square 0 1 2 3

```
8. Le langage C est un langage
     □ compilé
      □ lu, écrit, parlé
      □ interprété
      □ composé
 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 :
      \square 5
      \Box 7
      \Box 0
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. 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
      \square un désaccord entre la déclaration et la définition
         d'une fonction
      ☐ une fonction appelée avant sa déclaration
```

12. Laquelle de ces écritures correspond à la déclaration	15. Si le code :	Alors l'expression f(0) prendra la valeur :
d'une variable de type caractère en langage C?	struct toto_s	\Box 3
☐ char 'c';	{	\Box 4
☐ int char;	<pre>int n; double x;</pre>	
☐ char "c";	};	18. Un registre du processeur est :
□ char c;	précède la fonction main(), alors on peut écrire en début de main():	☐ une case mémoire interne au processeur qui sera manipulée directement lors des calculs
13. Pour déclarer une procédure afficher_date qui prend	☐ int struct toto_s = {3, -1e10};	☐ une unité de calcul spécialisée de l'ordinateur
en argument un struct date_s et affiche le contenu du struct, on écrit :	☐ struct toto_s toto;	\Box un composant qui contient la liste des fichiers du
,	\Box toto_s struct z = {3, 0.5};	système
☐ void afficher_date(date_s d);	□ toto_s n, x;	\Box une gamme de fréquence de fonctionnement du
☐ int afficher_date(date_s d);	☐ int toto.n = 3;	processeur
☐ void afficher_date(struct date_s d);	16. Si cet avertissement apparaît à la compilation :	19. Une de ces manière de composer les blocs de pro-
\square struct date_s afficher_date(struct date_s	d); warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	grammes ne fait pas partie des opérations de la programmation structurée :
14. Les lignes int i;	$\hfill\Box$ un désaccord entre la déclaration et la définition d'une fonction	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition
int x=0;	\Box une directive préprocesseur $\verb"#include"$ manquante	\Box retourner un bloc
for(i=0,i<5,i=i+1)	\Box une fonction appelée avant sa déclaration	☐ répéter un bloc tant qu'une condition est vérifée
{	\Box une fonction déclarée mais non définie	☐ mettre les blocs en séquence les uns à la suite des
x=x+1;	17. Soit la fonction f définie par :	autres
,	int f(int a)	20. Vous utilisez une boucle while quand:
\square comportent une erreur qui ne sera pas détectée	<pre>printf("a = \n", %d);</pre>	□ vous n'avez pas déclaré de fonction
□ comportent une erreur qui sera détectée au cours	if (a > 0)	□ vous ne connaissez pas le nombre d'itérations de
de l'analyse syntaxique	{	la boucle à l'avance
\square ne comportent aucune erreur	return 3;	\Box l'incrément de la variable de boucle n'est pas 1
☐ comportent une erreur qui sera détectée au cours de l'édition de lien	} return 4;	□ vous avez déjà fait un for dans le même programme principal

Éléments d'informatique – contrôle continue

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

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

```
1. 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 8 6 4 2 0
     \Box 02468
     \square 8 2
2. 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)
3. 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
     \Box 01234
     \square 1 2 3 4
4. 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();
```

```
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. 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 1 2 3 4
     \square 4 3 2 1
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 :
    \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");
8. 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);

```
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, x\} == b\{n, x\}
     □ a == b
     \Box (a.n == b.n) && (a.x == b.x)
10. 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
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
      □ (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
```

□ (a<n) || (n>b)

т		-1
	acence	

Prénom:	Nom:	
N° etu :		

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 1 0 1 0 1 0 1
     \Box 1 2 1 2 3
     \Box 0 0 0 1 1 1
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. 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);
4. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \Box 1
     \square 1.5
     \Box 0
     \square 0.5
```

```
5. 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>
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:
    □ A
     ПЪ
    ПВ
    \Box C
7. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
    \square ses cases
    \square ses blocs
    \square ses chants
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<=b
    \square (a<=n) && (n<=b)
     \square (n<=a) && (n<=b)
     \square (a<n) || (n>b)
```

```
9. 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 manguante
10. 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);
11. 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
     ☐ une directive préprocesseur #include manquante
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
12. 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
```

```
13. Un fichier source est:
      \square 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
      □ un document de référence du système
      \square un fichier texte qui sera traduit en instructions
         processeur
14. Le code suivant :
     int i;
     for (i = 0; i < 5; 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 0 1 2 3 4
15. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      \square int[] new tableau(5);
      \Box int tab[] = 5;
      □ char tableau[5];
      \square int toto[5];
      ☐ int toto[taille=5];
```

```
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 = 5
      \Box i = %d
     \Box j = 4
      \Box i = 0
17. Dans la commande gcc, l'option -Wall signifie :
      □ qu'il faut indenter le fichier source
     □ qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut lancer un déboggueur
      ☐ que l'on veut voir tous les avertissements
18. Une variable booléenne est un variable :
      □ qui est vraie ou fausse
     □ NaN (not a number, qui n'est pas un nombre)
      □ à laquelle une valeur vient d'être affectée
     □ jamais nulle
```

☐ réelle positive

```
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 0 1 0 1
     \Box 0 0 1 1 2 2
     \Box 0 1 2 0 1 2
     \Box 1 2 3 1 2
20. 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
     \square 8 2
     \Box 02468
```

 \Box 8 6 4 2

т		-1
	acence	

Prénom:	Nom:
N° etu :	

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

. 1	réponse fausse. Durée : 20 minutes.			
l.	Un programme en langage C doit comporter une et une seule définition de la fonction :			
	\square include			
	□ main			
	□ init			
	□ begin			
2.	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);			
	\square n = pgcd(n, 3);			
	\square n = pgcd(int p, int q);			
3.	Pour l'extrait de programme suivant :			
	<pre>int somme = 0; for (i = 0; i < 5; i = i + 1) { somme = somme + i; i = i + 1; /* attention ! */ }</pre>			
	<pre>printf("somme = %d",somme);</pre>			
	La valeur de somme affichée est :			
	\Box 6			
	\Box 0			
	\Box 10			
	\square 15			
1.	Un enregistrement permet de grouper plusieurs valeurs dans : $% \frac{1}{2} \left(\frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left(\frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left(\frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{2} \left(\frac{1}{2} - \frac{1}{2$			
	\square ses champs			
	□ ses cases			
	□ ses blocs			
	□ ses chants			

```
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(2/3);
     \square x = racine(racine(x)*racine(x));
     \square x = racine(x * x) - racine(x);
 6. Le langage C est un langage
      □ composé
     □ lu, écrit, parlé
     □ interprété
     □ compilé
 7. Sur unix (ou linux), la commande mkdir permet de :
      □ créer un fichier texte
     □ changer de répertoire courant
      □ créer un répertoire
      □ ouvrir un fichier texte
 8. 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)
 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(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);
10. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     \square analyse harmonique
     \square analyse syntaxique
      □ analyse sémantique
      \square analyse lexicale
```

```
11. 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
      \square il risque d'afficher bonjour à la place de coucou
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=%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");
13. Le code suivant :
     int age = 15;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
   affichera:
      □ Majeur
     □ Mineur
     □ rien
      ☐ Mineur
        Majeur
14. Le type des réels en C est :
     □ double
      □ char
      □ int
      \square real
```

15.	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 ma- chine et qui ne sont pas programmables en pro- grammation structurée
	\Box certains programmes sont de vrais plats de spaghetti
	□ en programmation structurée on peut programmer tous les calculs programmables en langage machine
16.	Pour l'extrait de programme suivant :
	<pre>int i; int j; for(i=4;i>0;i=i-1) { for(j=i;j<6;j=j+1) { printf("*"); }</pre>

```
printf(" ");
   qu'est ce qui sera affiché?
      ***** *** ***
17. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise :
     ☐ int afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu(int char);
     □ void afficher_menu();
     ☐ char afficher_menu(printf("menu"));
18. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \Box 7
     \square 3
     \square 8
```

```
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 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
     \Box 8 6 4 2 0
```

 \square 8 2

 ,,,,	ıce	

Prénom :	Nom:
N° etu :	

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 : \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) 2. 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 double exposant(double x, int n); \square exposant(double x, int n, int r); \square void exposant(double x^n); 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 \square une fonction déclarée mais non définie ☐ une fonction appelée avant sa déclaration □ une directive préprocesseur #include manquante 4. Après exécution du programme : lecture 8 r0 valeur 3 r1 mult r1 r0 valeur 1 r2 add r2 r0 ecriture r0 8 7 stop □ la case mémoire 8 contiendra 16 □ la case mémoire 8 contiendra 0 \square le bus explose

□ le terminal affiche 8

```
5. 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 void pgcd(int x, int y);
    \Box int pgcd(int x, int x);
     \square int pgcd(int x, y);
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:
    \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");
7. 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);
8. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
     ☐ yppasswd
     ☐ kwrite TP4
     □ mkdir TP4
     □ new TP4
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, x\} == b\{n, x\}
    \square a == b
     \Box (a.n == b.n) && (a.x == b.x)
```

 \square a = b

```
10. Sous unix (ou linux), la commande 1s permet de :
      □ voir des clips musicaux
     □ afficher la liste de fichiers contenus dans un
        répertoire
      \square afficher le contenu d'un fichier texte
      □ compiler un programme
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=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
12. Un fichier source est:
      ☐ 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
      □ un fichier texte qui sera traduit en instructions
        processeur
      □ un document qui doit être protégé
13. 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
      □ vous n'avez pas déclaré de fonction
      ☐ l'incrément de la variable de boucle n'est pas 1
```

<pre>□ la variable x vaut -½ □ le programme affiche **** □ la variable x vaut 16 □ le programme affiche x 17. 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?

18.	Un enregistrement permet de grouper plusieurs valeurs
	dans:
	\square ses blocs
	\square ses chants
	□ ses cases
	\square ses champs
19.	Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :
	\Box retourner un bloc
	$\hfill \square$ répéter un bloc tant qu'une condition est vérifée
	$\hfill \square$ sélectionner entre deux blocs à l'aide d'une condition
	$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres
20.	Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
	☐ #include <studlib.h></studlib.h>
	☐ #include <studio.h></studio.h>
	☐ #appart <stdlib.h></stdlib.h>
	☐ #include <stdio.h></stdio.h>

т		-1
	100000	- 1
	acence	

 \Box 1

 \Box 0

 \square 0.5

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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
     \Box 0
     \square 3
2. 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)
3. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     \square ses blocs
     \square ses chants
     \square ses champs
4. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
```

```
5. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
    \square init
    □ include
    \square main
    □ begin
6. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
    ☐ que l'on veut voir tous les avertissements
    □ qu'il faut lancer un déboggueur
    □ qu'on veut changer alétoirement de fond d'écran
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 n = carre():
    \Box int carre(2);
    \square n = carre(n);
    \square n = carre(int n);
8. Le code suivant :
    for (i = 4; i > 0; i = i - 1)
        printf("%d ", i);
   printf("\n");
   affichera:
    \Box 4 3 2 1 0
    \square 4 3 2 1
    \Box 0123
    \Box 01234
9. 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 -Wall prog.c -o prog.exe
    ☐ gcc prog.c -o -Wall prog.exe
```

```
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:
     \square toto_s n, x;
     \square int struct toto_s = {3, -1e10};
     □ struct toto_s toto;
     \Box toto_s struct z = {3, 0.5};
11. Sous unix (ou linux), pour créer un répertoire TP4
    dans le répertoire courant on peut utiliser la com-
   mande:
     ☐ mkdir TP4
     □ vppasswd
     □ new TP4
     ☐ kwrite TP4
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. 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. Le bus système sert à :
     ☐ Arriver à l'heure en cours
     □ transporter les processus du tourniquet au pro-
        cesseur
     ☐ Écrire des données sur le dique dur
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
```

utiliser l'instruction \Box j = 5	
<pre>□ int toto[taille=5]; □ char tableau[5]; □ int toto[5]; □ int toto[5]; □ int tab[] = 5; □ int tab[] = 5; □ int[] new tableau(5); 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)</pre>	-

Éléments d'informatique – contrôle continue

 \Box 6

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

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 7
     \Box 0
     \Box 5
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
     □ la case mémoire 8 contiendra 16
     □ la case mémoire 8 contiendra 0
     □ le terminal affiche 8
3. 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
```

```
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;
    \square int toto.n = 3;
    □ struct toto s toto:
    \square toto_s struct z = {3, 0.5};
    \Box int struct toto_s = {3, -1e10};
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 4
    \square 8
    \Box 0
    \Box 16
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 3
    \square 16
    \square 20
```

```
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. Le type des réels en C est :
      □ char
      \square int
      □ real
      □ double
 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 :
      \square 10
      \Box 0
      \square 15
      \Box 6
10. Une variable booléenne est un variable :
      \square iamais nulle
      □ NaN (not a number, qui n'est pas un nombre)
      ☐ réelle positive
      □ qui est vraie ou fausse
      □ à laquelle une valeur vient d'être affectée
```

```
11. Les lignes
                                                           15. Le code suivant :
                                                                                                                            \square 3
   int i;
                                                                int age = 15;
                                                                                                                            \Box 4
   int x=0;
                                                                if (age < 18)
   for(i=0,i<5,i=i+1)
                                                                                                                            \Box 0
                                                                     printf("Mineur\n");
      x=x+1;
                                                                }
                                                                                                                      19. Soit le programme principal suivant :
                                                                else
                                                                ₹
     \square ne comportent aucune erreur
                                                                                                                          int main()
                                                                    printf("Majeur\n");
     □ comportent une erreur qui sera détectée au cours
                                                                }
        de l'édition de lien
                                                                                                                           int a = 3;
     □ comportent une erreur qui ne sera pas détectée
                                                                                                                           int b = 5;
                                                               affichera:
     □ comportent une erreur qui sera détectée au cours
                                                                                                                           printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                           return EXIT_SUCCESS;
        de l'analyse syntaxique
                                                                 □ Mineur
12. Sur un ordinateur avec un seul processeur, habituelle-
                                                                 ☐ Mineur
   ment les processus sont exécutés :
                                                                    Majeur
                                                                                                                          appelant la fonction f ainsi définie :
     □ chacun son tour, après que le processus précédent
                                                                 □ Majeur
        a terminé
                                                                 □ rien
                                                                                                                          int f(int a, int b)
     □ en parallèle, chacun dans un registre
                                                           16. Sur unix (ou linux), la commande mkdir permet de :
     \square tour à tour, un petit peu à chaque fois
                                                                                                                            a = a + b;
                                                                 □ changer de répertoire courant
     \square tous ensemble
                                                                                                                            return a;
                                                                 □ créer un fichier texte
13. Soit la fonction f définie par :
                                                                 □ créer un répertoire
   int f(int a)
                                                                                                                          L'affichage dans le main est le suivant :
                                                                 □ ouvrir un fichier texte
      printf("a = \n", %d);
                                                           17. Pour déclarer une procédure afficher_menu sans ar-
                                                                                                                            \Box f(3,5)=8, a=3, b=5
      if (a > 0)
                                                               gument et qui ne renvoie rien on utilise :
                                                                 ☐ char afficher_menu(printf("menu"));
                                                                                                                            \Box f(a,b)=8, a=8, b=5
        return f(a - 1) + 1;
                                                                 □ void afficher_menu();
                                                                                                                            \Box f(a,b)=13, a=8, b=5
      }
                                                                 ☐ int afficher_menu(int char);
      return 4;
                                                                                                                            \Box f(a,b)=8, a=3, b=5
                                                                 ☐ double afficher_menu();
   Alors l'expression f(1) prendra la valeur :
                                                                 ☐ int afficher_menu();
                                                                                                                      20. Un fichier source est:
     \Box 5
                                                           18. Soit la fonction f définie par :
     \Box 0
                                                                                                                            □ un fichier texte qui sera traduit en instructions
                                                               int f(int a)
     \Box 1
                                                                                                                               processeur
     \Box 4
                                                                 printf("a = \n", %d);
                                                                                                                            □ un document qui doit être protégé
14. Pour déclarer une fonction saisie_utilisateur qui
                                                                 if (a > 0)
   demande à l'utilisateur d'entrer un entier au clavier et
                                                                 ₹
                                                                                                                            □ un document de référence du système
   renvoie cet entier on écrit:
                                                                   return 3;
     □ void saisie_utilisateur(char c);
                                                                                                                            □ un document illisible pour les humains
     □ void saisie_utilisateur(int n);
                                                                 return 4:
                                                                                                                            □ un fichier que l'ont doit citer dans les documents
     ☐ saisie_utilisateur(scanf(%d));
                                                                                                                               produits sur l'ordinateur
     ☐ int saisie_utilisateur();
                                                               Alors l'expression f(0) prendra la valeur :
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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
     \square 7
2. 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:
     \Box ccccc
     ☐ ABCDEF
     □ A
     \sqcap i
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
     \square (a.n == b.n) && (a.x == b.x)
    \Box a{n, x} == b{n, x}
     \Box a = b
```

```
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 y, int x);
    \Box int pgcd(int x, int x);
     \square void pgcd(int x, int y);
    \Box int pgcd(int x, y);
5. Quel est l'opérateur de différence en C :
     \square \neq
     \Box !
     \sqcap \iff
6. 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
     □ un caractère interdit en C
     \square une faute de frappe dans un appel de fonction
     □ une variable non déclarée
7. Pour déclarer un tableau d'entiers de taille 5, on peut
   utiliser l'instruction
     \square int[] new tableau(5);
    \square int toto[5]:
     \square int tab[] = 5;
     □ char tableau[5];
     ☐ int toto[taille=5];
8. Un bit est:
     □ la longueur d'un mot mémoire
     □ un battement d'horloge processeur
     \square un chiffre binaire (0 ou 1)
     ☐ l'instruction qui met fin à un programme
9. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses chants
     \square ses cases
     \square ses blocs
```

```
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 1
      \square 5
      \Box 0
      \Box 4
11. Un programme en langage C doit comporter une et une
    seule définition de la fonction :
      \square include
      □ begin
      \square main
      \square init
12. 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
      □ sélectionner entre deux blocs à l'aide d'une condi-
         tion
      \square mettre les blocs en séquence les uns à la suite des
         autres
      □ retourner un bloc
```

```
13. 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 1 0 1 0 1 0 1
     \Box 0 0 0 1 1 1
     \Box 1 2 1 2 3
14. Vous utilisez une boucle while quand:
     \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
     □ vous n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
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 j = 5
     \Box j = 0
     \Box j = 4
16. 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
     ☐ jamais nulle
     ☐ qui est vraie ou fausse
17. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
     \square analyse harmonique
     \square analyse syntaxique
     □ analyse sémantique
     \square analyse lexicale
18. 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
      ☐ il y a des calculs programmables en programma-
        tion structurée qui ne sont pas programmables en
        langage machine
19. 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)
      □ (n<=a) && (n<=b)
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 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);

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
N° etu :	

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 :
 - \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)
 - ☐ for(i=1;i<=5;i=i+1)
- 2. Un programme en langage C doit comporter une et une seule définition de la fonction :
 - \square init
 - □ main
 - □ begin
 - \square include

10

- 3. Après exécution jusqu'à la ligne 15 du programme C :
- int main() {
 11 int x = 5;
 13
 14 x = 3 * x +
- x = 3 * x + 1;15
- 16 ... 17 }
 - $\Box\,$ le programme affiche ****
 - $\Box\,$ le programme affiche x
 - \Box la variable x vaut $-\frac{1}{2}$
 - \Box la variable x vaut 16
- 4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
 - $\Box\,$ analyse sémantique
 - \square analyse harmonique
 - $\Box\,$ analyse syntaxique
 - \square analyse lexicale

- 5. Un enregistrement permet de grouper plusieurs valeurs dans :
 - \square ses chants
 - \square ses blocs
 - \square ses cases
 - \square ses champs
- 6. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction
 - □ loop i;
 - \square int k;
 - $\hfill\Box$ int loop n;
 - \square int %d;
- 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 4
- □ 5
- \Box 0
- \Box 1
- 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 == TRUE) && (B == TRUE)
 - □ A && B
 - \square (!A || B)
 - $\square !(!A || B) == (A \&\& !B)$

- $9. \ \, \text{Une variable booléenne est un variable}:$
 - ☐ jamais nulle☐ réelle positive
 - \Box à la quelle une valeur vient d'être affectée
 - □ NaN (not a number, qui n'est pas un nombre)
 - \Box qui est vraie ou fausse
- 10. 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 3
 - \Box le programme affiche "Faux"
 - \square la variable x vaut 5 et la variable y vaut 3
 - \Box la variable y vaut 5
- 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);</pre>
```

La valeur affichée est :

- \Box 16 \Box 0
- □ 8
- \Box 4
- 12. 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 carre(2); □ n = carre(n);
```

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

 13. 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; 14. 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); 15. Le langage C est un langage □ interprété □ lu, écrit, parlé 	□ une directive préprocesseur #include manquante □ un désaccord entre la déclaration et la définition d'une fonction □ 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 : □ 6 □ 42 □ 0	□ un point-virgule en trop □ une accolade manquante 19. Si le code: struct toto_s { int n; double x; }; précède la fonction main(), alors on peut écrire en début de main(): □ toto_s struct z = {3, 0.5}; □ int struct toto_s = {3, -1e10}; □ struct toto_s toto; □ int toto.n = 3; □ toto_s n, x; 20. Vous utilisez une boucle while quand:
 □ composé □ compilé 16. 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 	 □ 1 18. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit- on chercher dans le programme? □ une accolade en trop □ un point-virgule manquant 	 □ 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 n'avez pas déclaré de fonction □ l'incrément de la variable de boucle n'est pas 1

Éléments d'informatique – contrôle continue

□ rien

Prénom :	Nom:	
N° etu :		

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

1. 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 □ en temps d'accès □ certaines données de la mémoire de travail 2. 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 3. 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 1 2 3 1 2 \Box 0 1 0 1 0 1 \Box 0 1 2 0 1 2 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=%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");

```
5. Pour déclarer une procédure afficher_menu sans ar-
  gument et qui ne renvoie rien on utilise :
    □ void afficher_menu();
    ☐ int afficher_menu(int char);
    ☐ char afficher_menu(printf("menu"));
    ☐ int afficher_menu();
    ☐ double afficher_menu();
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)=13, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=3, b=5
7. Le code suivant :
   int age = 20;
   if (age < 18)
   {
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Mineur
    □ Mineur
       Majeur
    □ Majeur
```

```
8. Le code suivant :
     int i;
     for (i = 0; i < 5; 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
 9. Le code suivant :
     int age = 18;
     if (age < 18)
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
    affichera:
     □ Mineur
        Majeur
      □ Majeur
     □ Mineur
     □ rien
10. 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);
```

11. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	☐ un désaccord entre la déclaration et la définition d'une fonction	18. Soit la fonction g définie par :
☐ char "c";	\square une fonction appelée avant sa déclaration	int g(int a)
☐ char c;	\square une fonction déclarée mais non définie	{ printf("a = \n", %d);
☐ char 'c';	15. Si racine est une fonction prenant en entrée un réel	if (1 > 0)
☐ int char;	et renvoyant la racine carrée de cet réel, et que x est	{
12. Après exécution jusqu'à la ligne 15 du programme ${\bf C}$:	une variable réelle définie et initialisée, il est incorrect d'écrire :	return 5; }
10	\Box x = racine(racine(x)*racine(x));	return 7;
11 int main() {	\square x = racine(2/3);	}
12 int x = 5; 13	\square x - 1 = racine(x);	Alors l'expression g(0) prendra la valeur :
x = 3 * x + 1;	$\square x = racine(x * x) - racine(x);$	\square 0
15	16. Si le code :	□ 7
16 17 }	struct toto_s	
	int n;	
☐ la variable x vaut 16	double x;	19. Pour déclarer une fonction pgcd qui calcule et renvoie
☐ le programme affiche ****	};	le plus grand diviseur commun de deux entiers positifs
\Box le programme affiche x	précède la fonction main(), alors on peut écrire en	passés en arguments on écrit :
\Box la variable x vaut $-\frac{1}{2}$	début de main() :	☐ void pgcd(int x, int y);
13. Vous utilisez une boucle while quand:	\Box toto_s struct z = {3, 0.5};	☐ int pgcd(int x, int x);
□ vous n'avez pas déclaré de fonction	☐ struct toto_s toto;	☐ int pgcd(int x, y);
☐ l'incrément de la variable de boucle n'est pas 1	□ toto_s n, x;	·
□ vous ne connaissez pas le nombre d'itérations de	\Box int struct toto_s = {3, -1e10};	\Box int pgcd(int y, int x);
la boucle à l'avance	☐ int toto.n = 3;	20. Dans la commande gcc, l'option -Wall signifie :
□ vous avez déjà fait un for dans le même programme principal	17. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :	☐ qu'on veut changer alétoirement de fond d'écran
	☐ int factorielle();	☐ que l'on veut voir tous les avertissements
14. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max	, ☐ struct int factorielle(int n);	☐ qu'il faut indenter le fichier source
, que doit-on chercher dans le programme?	\square int factorielle(double n);	•
☐ une directive préprocesseur #include manquante	☐ int factorielle(int x);	□ qu'il faut lancer un déboggueur

Éléments d'informatique – contrôle continue

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

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 == TRUE) && (B == TRUE)
☐ A && B
☐ (!A || B)
☐ !(!A || B) == (A && !B)

2. 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
☐ 1 2 3 4
```

3. Le type des réels en C est :

□ char
□ double

 $\Box 01234$

 \square real \square int

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));

☐ int saisie_utilisateur();
☐ void saisie_utilisateur(int n);

5. Un enregistrement permet de grouper plusieurs valeurs dans :

☐ ses cases☐ ses champs

☐ ses chants☐ ses blocs

6. Un fichier source est:

 \Box un fichier que l'ont doit citer dans les documents produits sur l'ordinateur

 \Box un document illisible pour les humains

□ un document qui doit être protégé

 \Box un fichier texte qui sera traduit en instructions processeur

□ un document de référence du système

7. 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 :

□ сссссс

□ A
□ ABCDEF

□ i

8. 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 comporte une boucle infinie

 $\Box\,$ il n'affiche rien

 \Box il ne compile pas

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);</pre>
```

qu'est ce qui sera affiché par ce printf?

```
\Box j = %d
\Box j = 5
```

 \Box j = 4

□ j = 0

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

```
\Box un débogueur
```

 \Box printf("Valeur de n ? %g\n", n);

□ scanf("%d", &n);

 \square printf("Valeur de n ? %d\n", n);

11. 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?

□ j = 4

\Box j = %d
□ j = 0
□ j = 5
12. Un bit est:
\Box un battement d'horloge processeur
\Box la longueur d'un mot mémoire
\Box l'instruction qui met fin à un programme
\Box un chiffre binaire (0 ou 1)
13. Avant de faire appel à une fonction il est nécessaire de :
☐ l'avoir déclarée
$\hfill \square$ avoir défini une constante symbolique de la taille de cette fonction
☐ l'avoir définie
\Box l'avoir déclarée et définie
14. 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 :
□ A
□ В
□ b
□ с

```
15. 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)
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 n = carre();
     \square n = carre(int n);
     \square int carre(2);
      \square n = carre(n);
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
     □ une directive préprocesseur #include manquante
      \square un désaccord entre la déclaration et la définition
         d'une fonction
18. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     }
     printf("Majeur\n");
```

```
affichera:
      □ Majeur
      ☐ Mineur
      □ Mineur
        Majeur
      \square rien
19. 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 1
      \Box 6
      \Box 0
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 :
     \Box 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);

Éléments d'informatique – contrôle continue

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

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

```
    Le code suivant :
        int i;
        for (i = 0; i < 5; i = i + 1)
        {
             printf("%d ", i);
        }
        printf("\n");
        affichera :
        □ 0 1 2 3
        □ 4 3 2 1 0
        □ 0 1 2 3 4
        □ 4 3 2 1</li>
    Vous avez déclaré préalablement un ensemble de fonctions utilisées par votre programme principal. L'ordre
```

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 al
phabétique \Box dans lequel vous avez déclaré ces fonction
```

- \Box dans lequel ces fonctions sont appelées dans le main
- $\Box\,$ un ordre quel conque

3. Un enregistrement permet de grouper plusieurs valeurs dans :

```
□ ses blocs□ ses cases□ ses champs
```

 \square ses chants

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

```
de :

□ l'avoir déclarée
```

Ш	avoir défini une constante symbolique de la taille
	de cette fonction
	l'avoir définie

```
□ l'avoir déclarée et définie
```

```
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)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
6. 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 i = 4

```
7. 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 sera détectée au cours
       de l'édition de lien
    □ ne comportent aucune erreur
    □ comportent une erreur qui ne sera pas détectée
8. 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 0 1 2
    \Box 0 1 2 0 1 2 3
    \Box 0 1 2 3 0 1 2
9. 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
     }
    \square le terminal affiche x = 2
    □ le terminal affiche "Faux"
    \square le terminal affiche 5
```

 \square le terminal affiche x = 5

10. Une variable booléenne est un variable :	14. Pour déclarer une procédure afficher_menu sans ar-	18. Soit la fonction f définie par :
\square qui est vraie ou fausse	gument et qui ne renvoie rien on utilise:	int f(int a)
□ NaN (not a number, qui n'est pas un nombre)	☐ void afficher_menu();	{
□ réelle positive	☐ double afficher_menu();	<pre>printf("a = \n", %d); if (a > 0)</pre>
□ jamais nulle	☐ char afficher_menu(printf("menu"));	{
☐ à laquelle une valeur vient d'être affectée	☐ int afficher_menu(int char);	return f(a - 1) + 1;
11. Dans la commande gcc, l'option -Wall signifie :	☐ int afficher_menu();	return 4;
□ qu'il faut lancer un déboggueur	15. Sous unix (ou linux), la commande 1s permet de :	}
□ qu'il faut indenter le fichier source	□ compiler un programme	Alors l'expression f(1) prendra la valeur :
☐ qu'on veut changer alétoirement de fond d'écran	□ voir des clips musicaux	\Box 4 \Box 0
☐ que l'on veut voir tous les avertissements	\square afficher la liste de fichiers contenus dans un	□ 1
12. L'ordonnancement par tourniquet permet :	répertoire	□ 5
12. L'ordonnancement par tourniquet permet :	$\hfill\Box$ afficher le contenu d'un fichier texte	19. Si cet avertissement apparaît à la compilation :
\Box de doubler la mémoire disponible	16. Laquelle de ces écritures correspond à la déclaration	warning: implicit declaration of function 'max'
\Box de ne pas perdre de temps avec la commutation	d'une variable de type caractère en langage C?	, que doit-on chercher dans le programme?
de contexte	☐ char "c";	☐ un désaccord entre la déclaration et la définition d'une fonction
☐ d'entretenir l'illusion que les processus tournent en parallèle	□ char c;	d une fonction ☐ une directive préprocesseur #include manquante
☐ d'afficher des ronds colorés à l'écran	☐ int char;	□ une fonction déclarée mais non définie
ancher des fonds colores à l'écran	☐ char 'c';	☐ une fonction appelée avant sa déclaration
13. Si pgcd est une fonction prenant en entrée deux entiers	17. Sur unix (ou linux), la commande mkdir permet de :	20. Si <i>n</i> est une variable entière pour demander sa valeur
et renvoyant un entier, il est correct d'écrire :		à l'utilisateur, on utilise plutôt :
\square int n = pgcd();	□ ouvrir un fichier texte	☐ printf("Valeur de n ? %d\n", n);
\square int pgcd(2);	□ créer un répertoire	□ un débogueur
\square n = pgcd(int p, int q);	\Box créer un fichier texte	☐ printf("Valeur de n ? %g\n", n);
\square n = pgcd(n, 3);	\Box changer de répertoire courant	<pre>□ scanf("%d", &n);</pre>

Éléments d'informatique – contrôle continue

Prénom:	Nom:
	NOIII .
N° etu :	
11 004.	

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 : erreur: conflicting types for 'max', que doiton chercher dans le programme? \square 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 2. 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 1 \Box 6 \Box 0 3. 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 prog.exe -Wall -o prog.c ☐ gcc -Wall prog.c -o prog.exe

ligne de commande :

□ gcc prog.c -o -Wall prog.exe
□ gcc -Wall prog.exe -o prog.c
□ gcc prog.exe -Wall -o prog.c
□ gcc -Wall prog.c -o prog.exe

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 faute de frappe dans un appel de fonction
□ une variable non déclarée

 \Box une directive préprocesseur #include manquante

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");
☐ 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);
6. L'écriture 101 en binaire correspond au nombre natu-
rel:
□ 101
7. Le code suivant :
<pre>int age = 20; if (age < 18)</pre>
{
<pre>printf("Mineur\n");</pre>
}
<pre>printf("Majeur\n");</pre>
affichera:
□ rien
☐ Mineur
☐ Mineur
Majeur
□ Majeur
8. Afin de représenter la taille d'un tableau, définir une
constante symbolique N valant 3.
☐ #define N = 3
☐ #define taille = N
☐ #define taille = 3
☐ #define N 3
9. Une variable booléenne est un variable :
□ jamais nulle
□ réelle positive
□ à laquelle une valeur vient d'être affectée
□ NaN (not a number, qui n'est pas un nombre)
□ qui est vraie ou fausse

10.	Si x est une variable réelle (de type double) alors $x = 3/2$ lui affecte la valeur :
	□ 1
	\square 0.5
	□ 1.5
	\square 0
11.	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 afficher_menu(int char);
12.	Le bus système sert à :
	\Box transporter les processus du tourniquet au processeur
	\Box Arriver à l'heure en cours
	\Box Écrire des données sur le dique dur
	\Box Transférer des données et intructions entre processeur et mémoire
13.	Quel est le problème d'un programme comportant les lignes suivantes?
	while (1) {
	<pre>printf("coucou\n"); }</pre>
	\square il comporte une boucle infinie
	☐ il risque d'afficher bonjour à la place de coucou
	☐ il ne compile pas
	□ il n'affiche rien
14.	Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande :
	\square new TP4
	☐ mkdir TP4
	\square yppasswd
	□ kwrite TP4

15. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :	18. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	□ rien
□ n = mccarthy(p, q);	☐ int char;	□ Mineur Majeur
<pre>□ n = mccarthy();</pre> □ x = mccarthy(n);	☐ char "c";	\square Majeur
☐ int mccarthy(int 2);	□ char 'c'; □ char c;	20. Pour l'extrait de programme suivant :
16. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :	19. Le code suivant :	<pre>int produit = 0; int serie[4] = {2, 2, 2, 2};</pre>
☐ int factorielle(int 2);	int age = 15;	for $(i = 0; i < 4; i = i + 1)$
\square n = factorielle(p, q);	if (age < 18)	{
☐ printf("%d", factorielle(n));	{	<pre>produit = produit * serie[i];</pre>
☐ n = factorielle();	<pre>printf("Mineur\n"); }</pre>	} printf("produit = %d", produit);
17. Vous utilisez une boucle while quand:	else	
\Box vous ne connaissez pas le nombre d'itérations de	{	La valeur affichée est :
la boucle à l'avance	<pre>printf("Majeur\n");</pre>	\Box 16
□ vous avez déjà fait un for dans le même programme principal	}	\Box 4
□ vous n'avez pas déclaré de fonction	affichera:	\Box 0
☐ l'incrément de la variable de boucle n'est pas 1	☐ Mineur	□ 8

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 : □ alphabétique \square un ordre quelconque □ dans lequel ces fonctions sont appelées dans le main \square dans lequel vous avez déclaré ces fonction 2. 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 exposant(double x, int n, int r); \square double exposant(double x, int n); 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 \Box a = b \square a{n, x} == b{n, x} \square (a.n == b.n) && (a.x == b.x) 4. Un fichier source est: □ un document qui doit être protégé \square un document illisible pour les humains
 - \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 de référence du système

```
5. 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
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
    \square ses blocs
    \square ses champs
    \square ses chants
    \square ses cases
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 -Wall prog.exe -o prog.c
    ☐ gcc prog.c -o -Wall prog.exe
8. 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);
```

 \Box int factorielle(int x):

```
9. 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
      \square une variable non déclarée
      \square un caractère interdit en C
      ☐ une directive préprocesseur #include manquante
10. Le code suivant :
     int i:
     for (i = 4; i >= 0; i = i - 1)
          printf("%d ", i);
     printf("\n");
    affichera:
     \square 1 2 3 4
     \Box 01234
     \Box \ 4\ 3\ 2\ 1\ 0
     \square 4 3 2 1
11. Après exécution jusqu'à la ligne 14 du programme C:
       int main() {
 10
 11
            int x = 5;
 12
            printf(" x = %d\n", 2);
 13
 14
 15
            . . .
 16
      }
      \square le terminal affiche x = 2
      □ le terminal affiche 5
      □ le terminal affiche "Faux"
      \square le terminal affiche x = 5
12. 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";

<pre>13. 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)</pre>	<pre>16. Après exécution jusqu'à la ligne 15 du programme C : 10</pre>	{ printf("*"); } printf(" "); }
☐ for(i=1;i<5;i=i+1)	15	qu'est ce qui sera affiché?
☐ for(i=0;i<5;i=i+1)	16 17 }	_ *** *** **** _ ** *** ****
14. Quel est l'opérateur de différence en C :	\Box la variable x vaut $-\frac{1}{2}$	
□!	☐ la variable x vaut 16	
□ !=	☐ le programme affiche ****	19. Après la déclaration : int mccarthy(int n);, il est
	\square le programme affiche x	correct d'écrire :
□≠	17. Le type des réels en C est : \Box double	$\Box n = mccarthy(p, q);$ $\Box x = mccarthy(n);$
15. 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	□ int□ real□ char	☐ int mccarthy(int 2); ☐ n = mccarthy(); 20. L'écriture 101 en binaire correspond au nombre natu-
☐ un désaccord entre la déclaration et la définition d'une fonction	18. Pour l'extrait de programme suivant : int i; int j;	rel : □ 3 □ 5
\Box une directive préprocesseur $\verb"#include"$ manquante	for(i=4;i>0;i=i-1)	□ 101
\Box une fonction appelée avant sa déclaration	{ for(j=i;j<6;j=j+1)	

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 1 2 1 2 3
- \Box 0 1 2 0 1 2
- \Box 0 0 0 1 1 1
- \Box 0 1 0 1 0 1 0 1

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 == TRUE) && (B == TRUE)
```

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

3. 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, y);
```

- \square int pgcd(int x, int x);
- 4. 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
    □ le terminal affiche 8
    □ la case mémoire 8 contiendra 16
     \square le bus explose
5. 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
6. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
  {
     printf("coucou\n");
     \square il ne compile pas
    \Box il comporte une boucle infinie
    □ il n'affiche rien
     ☐ il risque d'afficher bonjour à la place de coucou
```

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

7. Pour l'extrait de programme suivant :

ıa	vaieur	amenee	est
	$\supset 4$		

8
16
Ω

```
8. 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);
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
     \Box f(a,b)=8, a=8, b=5
10. Pour déclarer un tableau d'entiers de taille 5, on peut
   utiliser l'instruction
     \square int[] new tableau(5);
     ☐ char tableau[5];
```

☐ int toto[taille=5];

 \square int tab[] = 5;

 \square int toto[5];

```
11. Quelle étape de la compilation vient d'échouer lors-
                                                              15. Le code suivant :
                                                                                                                                  }
    qu'on a un message comme celui-ci :
                                                                                                                                  return 4;
                                                                   int age = 20;
    Undefined symbols : "_prinft" ou
                                                                   if (age < 18)
    référence indéfinie vers « prinft »
                                                                                                                                Alors l'expression f(1) prendra la valeur :
      □ l'édition de liens
                                                                        printf("Mineur\n");
                                                                                                                                  \Box 0
      □ l'analyse sémantique
      □ l'analyse des entrées clavier
                                                                   printf("Majeur\n");
                                                                                                                                  \Box 5
      □ l'analyse harmonique
                                                                  affichera:
                                                                                                                                  \Box 4
12. Le code suivant :
                                                                    ☐ Mineur
                                                                                                                                  \Box 1
                                                                       Majeur
     int i:
     for (i = 0; i < 5; i = i + 1)
                                                                                                                            19. Un bit est:
                                                                    \square rien
                                                                                                                                  \square la longueur d'un mot mémoire
                                                                    □ Mineur
         printf("%d ", i);
                                                                                                                                  \Box un battement d'horloge processeur
                                                                    □ Majeur
     printf("\n");
                                                              16. Pour afficher à l'aide de printf("%d\n",tab[i]);
                                                                                                                                  \square un chiffre binaire (0 ou 1)
    affichera:
                                                                  le contenu d'un tableau de 5 entiers initialisé au
                                                                                                                                  ☐ l'instruction qui met fin à un programme
                                                                  préalable, on utilise plutôt :
      \Box 0123
                                                                                                                            20. Le code suivant :
      \Box 01234
                                                                    \square for(i=1;i<5;i=i+1)
      \Box \ 4\ 3\ 2\ 1\ 0
                                                                   \square for(i=0;i<=5;i=i+1)
                                                                                                                                 int age = 20;
      \square 4 3 2 1
                                                                   \square for(i=1;i<=5;i=i+1)
                                                                                                                                 if (age < 18)
13. Si pgcd est une fonction prenant en entrée deux entiers
                                                                                                                                 {
                                                                    \Box for(i=0;i<5;i=i+1)
    et renvoyant un entier, il est correct d'écrire :
                                                                                                                                      printf("Mineur\n");
                                                             17. Un enregistrement permet de grouper plusieurs valeurs
                                                                                                                                 }
      \square int pgcd(2);
                                                                  dans:
                                                                                                                                 else
      \square n = pgcd(n, 3);
                                                                    □ ses cases
                                                                                                                                 {
      \square int n = pgcd();
                                                                                                                                      printf("Majeur\n");
                                                                    \square ses blocs
      \square n = pgcd(int p, int q);
                                                                    \square ses champs
14. Vous avez déclaré préalablement un ensemble de fonc-
    tions utilisées par votre programme principal. L'ordre
                                                                    \square ses chants
                                                                                                                                affichera:
    dans lequel vous devez maintenant définir ces fonctions
                                                              18. Soit la fonction f définie par :
    est l'ordre :
                                                                                                                                  ☐ Mineur
                                                                  int f(int a)
      \square dans lequel vous avez déclaré ces fonction
                                                                                                                                     Majeur
      □ alphabétique
                                                                                                                                  \square rien
                                                                    printf("a = \n", %d);
      \square dans lequel ces fonctions sont appelées dans le
                                                                    if (a > 0)
                                                                                                                                  □ Majeur
         main
                                                                    {
      \square un ordre quelconque
                                                                                                                                  ☐ Mineur
                                                                      return f(a - 1) + 1;
```

Éléments d'informatique – contrôle continue

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

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

1. 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

 \square le terminal affiche x = 2

□ le terminal affiche "Faux"

 \square le terminal affiche 5

2. 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 ne connaissez pas le nombre d'itérations de la boucle à l'avance

□ vous avez déjà fait un **for** dans le même programme principal

3. Quel est l'opérateur de différence en C :

```
□ <>
□ ≠
```

□ !=

□!

4. Sous unix (ou linux), la commande cd permet de :

```
\Box\,ouvir un bureau partagé (common desktop)
```

 $\Box\,$ détruire un fichier

□ changer de répertoire courant

 \Box jouer de la musique

 $\hfill \square$ récupérer un programme arrêté avec la commande ab

```
5. 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 :

	3

 \Box 4 \Box 0

6. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :

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

 \square x = mccarthy(n);

 \Box int mccarthy(int 2);

 \square n = mccarthy();

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

```
\hfill\Boxanalyse sémantique
```

 $\hfill\Box$ analyse harmonique

 \square analyse lexicale

 $\Box\,$ analyse syntaxique

8. 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 n = carre(n);

 \square int n = carre();

```
9. \  \, \text{Le code suivant}:
```

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

affichera:

 \square Majeur

 \square rien

 \square Mineur

□ Mineur Majeur

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 :

\Box dans	lequel	\cos	fonctions	sont	appelées	dans	le
main							

 \square dans lequel vous avez déclaré ces fonction

□ alphabétique

□ un ordre quelconque

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=1;i<5;i=i+1)
```

 \square for(i=0;i<=5;i=i+1)

☐ for(i=1;i<=5;i=i+1)

 \square for(i=0;i<5;i=i+1)

12. 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)

☐ (A == TRUE) && (B == TRUE)

13. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
\square int n = pgcd();
$\square \ n = pgcd(int \ p, \ int \ q);$
$\square \ n = pgcd(n, 3);$
☐ int pgcd(2);
14. Pour l'extrait de programme suivant :
<pre>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);</pre>
La valeur affichée est :
\Box 4
□ 16
□ 8
□ 0
15. Avant de faire appel à une fonction il est nécessaire de :
□ l'avoir déclarée
□ l'avoir définie
□ l'avoir déclarée et définie
\Box avoir défini une constante symbolique de la taille de cette fonction

```
16. Si cette erreur apparaît à la compilation :
                                                                 \Box j = 5
    error: expected ';' before '}' token que doit-
                                                                 \Box j = 4
    on chercher dans le programme?
                                                           19. L'écriture 101 en binaire correspond au nombre natu-
      \Box\, une accolade manquante
                                                               rel:
      \square une accolade en trop
                                                                 \Box 5
      □ un point-virgule en trop
                                                                 \square 101
      ☐ un point-virgule manquant
                                                                 \square 3
17. Sous unix (ou linux), pour créer un répertoire TP4
                                                                 \Box 4
    dans le répertoire courant on peut utiliser la com-
    mande:
                                                           20. Soit le programme principal suivant :
      ☐ kwrite TP4
                                                               int main()
      ☐ mkdir TP4
      □ new TP4
                                                                int a = 3;
                                                                int b = 5;
      □ yppasswd
                                                                printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
18. Soit un programme contenant les lignes suivantes :
                                                                return EXIT_SUCCESS;
     int i = 0;
     int j = 0;
                                                               appelant la fonction f ainsi définie :
     for (i = 0; i < 3; i = i + 1)
                                                               int f(int a, int b)
         for (j = 0; j < 5; j = j + 1)
                                                                 a = a + b;
                                                                 return a;
         }
                                                               L'affichage dans le main est le suivant :
     printf("j = %d\n", j);
                                                                 \Box f(a,b)=13, a=8, b=5
                                                                 \Box f(3,5)=8, a=3, b=5
    qu'est ce qui sera affiché par ce printf?
                                                                 \Box f(a,b)=8, a=8, b=5
      \Box j = 0
                                                                 \Box f(a,b)=8, a=3, b=5
      \Box j = %d
```

Éléments d'informatique - contrôle continue

Prénom:	Nom:
N° etu:	

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 ouvrir un fichier texte □ créer un répertoire □ changer de répertoire courant 2. 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> 3. Un enregistrement permet de grouper plusieurs valeurs dans: \square ses chants \square ses champs □ ses cases \square ses blocs 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 0 1 2 3 \Box 0 1 2 0 1 2 \Box 0 1 2 3 0 1 2

 \Box 0 0 1 1 2 2 3

```
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 »
     \square l'analyse des entrées clavier
    □ l'édition de liens
    ☐ l'analyse sémantique
     ☐ l'analyse harmonique
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?
    □ A && B
    \square !(!A | | B) == (A && !B)
    \square (A == TRUE) && (B == TRUE)
    \square (!A || B)
7. 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
      }
     ☐ le terminal affiche "Faux"
    □ le terminal affiche 5
    \square le terminal affiche x = 2
    \square le terminal affiche x = 5
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
    ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
```

☐ de doubler la mémoire disponible

```
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(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
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, x} == b{n, x}
     \square (a.n == b.n) && (a.x == b.x)
     \square a == b
     \Box a = b
11. 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 indenter le fichier source
      □ qu'il faut lancer un déboggueur
```

 12. 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 ☐ une fonction appelée avant sa déclaration 13. 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 	15. Avant de faire appel à une fonction il est nécessaire de : □ l'avoir déclarée et définie □ l'avoir défini une constante symbolique de la taille de cette fonction □ l'avoir déclarée 16. 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 □ jamais nulle □ qui est vraie ou fausse	<pre>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 : □ 3</pre>
□ B □ A □ C 14. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n"); } □ il ne compile pas □ il comporte une boucle infinie □ il n'affiche rien □ il risque d'afficher bonjour à la place de coucou	<pre>17. Le type des réels en C est :</pre>	 □ 4 □ 0 20. Un programme en langage C doit comporter une et une seule définition de la fonction : □ init □ main □ begin □ include

т		-1
	acence	

 \square 101

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 \square int □ double □ real 2. 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 □ un désaccord entre la déclaration et la définition d'une fonction ☐ une fonction appelée avant sa déclaration 3. 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 4. 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 15 \Box 10 5. L'écriture 101 en binaire correspond au nombre naturel: \Box 4 \square 3 \Box 5

```
6. 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
     \Box la variable x vaut 0
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 n = carre();
     \square n = carre(n);
     \square n = carre(int n);
     \square int carre(2);
8. 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 n = pgcd(int p, int q);
     \square int pgcd(2);
     \square int n = pgcd();
9. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses chants
     \square ses champs
     □ ses cases
     \square ses blocs
```

```
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 :
      \square (a<n) || (n>b)
      □ a<=n<=b
      □ (a<=n) && (n<=b)
      \square (n<=a) && (n<=b)
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};
      \square int struct toto_s = {3, -1e10};
      □ struct toto_s toto;
      \square int toto.n = 3;
12. 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'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

```
13. Pour l'extrait de programme suivant :
                                                            16. Au début de la fonction main() on place le code :
                                                                                                                            L'affichage dans le main est le suivant :
                                                                 char i;
                                                                                                                              \Box f(a,b)=8, a=8, b=5
      int produit = 1;
                                                                 for (i = 'A'; i \le 'F'; i = i + 1)
      int serie[4] = \{2, 2, 2, 2\};
                                                                                                                               \Box f(a,b)=8, a=3, b=5
      for (i = 0; i < 4; i = i + 1)
                                                                                                                              \Box f(3,5)=8, a=3, b=5
                                                                   printf("%c", i);
        produit = produit * serie[i];
                                                                                                                              \Box f(a,b)=13, a=8, b=5
                                                                 printf("\n");
                                                                                                                         19. Le langage C est un langage
                                                                Alors l'affichage sera :
      printf("produit = %d", produit);
                                                                  \square ccccc
                                                                                                                               □ composé
   La valeur affichée est :
                                                                  \square i
                                                                                                                              □ compilé
      \Box 16
                                                                  \square A
                                                                                                                               □ interprété
                                                                  □ ABCDEF
      \square 8
                                                                                                                               □ lu, écrit, parlé
                                                            17. Un programme en langage C doit comporter une et une
      \Box 4
                                                                seule définition de la fonction :
                                                                                                                         20. Le code suivant :
      \Box 0
                                                                  \square begin
                                                                                                                              int age = 20;
                                                                  \square include
14. Une variable booléenne est un variable :
                                                                                                                              if (age < 18)
                                                                  \square init
                                                                                                                              {
      ☐ jamais nulle
                                                                  \square main
                                                                                                                                   printf("Mineur\n");
                                                            18. Soit le programme principal suivant :
     □ NaN (not a number, qui n'est pas un nombre)
                                                                                                                              }
                                                                                                                              else
                                                                int main()
      ☐ réelle positive
                                                                                                                              {
                                                                {
      \square qui est vraie ou fausse
                                                                                                                                   printf("Majeur\n");
                                                                 int a = 3;
                                                                                                                              }
                                                                 int b = 5;
     □ à laquelle une valeur vient d'être affectée
                                                                 printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
15. Laquelle de ces écritures correspond à la déclaration
                                                                 return EXIT_SUCCESS;
                                                                                                                             affichera:
   d'une variable de type caractère en langage C?
                                                                                                                               \square rien
                                                                appelant la fonction f ainsi définie :
      \Box char c;
                                                                                                                               □ Majeur
                                                                int f(int a, int b)
     ☐ char "c";
                                                                                                                               ☐ Mineur
     □ char 'c';
                                                                  a = a + b;
                                                                                                                               ☐ Mineur
                                                                  return a;
      \square int char;
                                                                                                                                 Majeur
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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?
 - $\Box\,$ une fonction appelée avant sa déclaration
 - ☐ un désaccord entre la déclaration et la définition d'une fonction
 - $\hfill \square$ une directive préprocesseur $\#\mbox{include}$ man quante
 - \Box une fonction déclarée mais non définie
- 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");</pre>
```

Alors l'affichage sera :

- \square i
- ☐ ABCDEF
- \square A
- \square ccccc
- 3. Après exécution jusqu'à la ligne 15 du programme ${\bf C}$:

```
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 $-\frac{1}{2}$
- \Box le programme affiche x
- \Box le programme affiche ****

```
4. Le code suivant :
   int i;
   for (i = 4; i > 0; i = i - 1)
        printf("%d ", i);
   printf("\n");
  affichera:
    \Box 43210
    \Box 01234
    \square 0 1 2 3
    \square 4 3 2 1
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
    \square l'incrément de la variable de boucle n'est pas 1
    □ vous avez déjà fait un for dans le même pro-
       gramme principal
6. 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
    □ un désaccord entre la déclaration et la définition
       d'une fonction
    ☐ une fonction appelée avant sa déclaration
7. 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 3 0 1 2
      \Box 0 1 2 0 1 2 3
     \Box 0 0 1 1 2 2 3
 8. Sous unix (ou linux), pour créer un répertoire TP4
    dans le répertoire courant on peut utiliser la com-
   mande:
     □ new TP4
     □ mkdir TP4
     ☐ yppasswd
     ☐ kwrite TP4
 9. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
    printf("Majeur\n");
    affichera:
     □ Majeur
     □ Mineur
      ☐ Mineur
        Majeur
     \square rien
10. 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
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(x * x) - racine(x);
     \square x = racine(racine(x)*racine(x));
      \square x - 1 = racine(x);
```

12.	Un enregistrement permet de grouper plusieurs valeurs	
	dans:	
	□ ses blocs	
	\square ses chants	
	\square ses champs	
	□ ses cases	16.
13.	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	17.
14.	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 qui doit être protégé	
	\Box un document de référence du système	
	\Box un fichier texte qui sera traduit en instructions processeur	
l5.	Le code suivant :	
	int i; for (i = 8; i > 0; i = i - 2) {	
	<pre>printf("%d ", i); }</pre>	
	<pre>printf("\n");</pre>	

```
affichera:
 \square 8 2
 \Box 8 6 4 2 0
 \Box 02468
 \square 8 6 4 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;
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?
   ***** *** ***
```

```
18. 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
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 7
      \Box 0
      \Box 5
20. L'écriture 111 en binaire correspond au nombre natu-
    rel:
      \square 3
      □ 111
      \Box 7
```

 \square 8

 \Box 4

Éléments d'informatique – contrôle continue

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

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

```
1. 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
2. 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
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 :
     \square 3
     \Box 0
```

```
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 int pgcd(int y, int x);
     \square void pgcd(int x, int y);
    \Box int pgcd(int x, y);
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 :
     \Box 4
     \square 5
     \square 0
     \Box 1
6. Une variable booléenne est un variable :
     □ à laquelle une valeur vient d'être affectée
     □ qui est vraie ou fausse
     □ réelle positive
     □ NaN (not a number, qui n'est pas un nombre)
     \square jamais nulle
7. 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 4 3 2 1
     \Box 01234
```

 \square 0 1 2 3

```
8. 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
      \square une fonction déclarée mais non définie
      ☐ une directive préprocesseur #include manquante
9. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 5
     \Box 4
     \square 3
      \square 101
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
      □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
11. Le code suivant :
     int i:
     for (i = 0; i < 7; i = i + 2)
          printf("%d ", i);
    printf("\n");
   affichera:
     \Box 02468
     \Box 0246
     \Box 0 1 2 3 4 5 6 7
     \Box 0 1 2 3 4 5 6
```

```
12. Quel est le problème d'un programme comportant les
                                                          16. Le code suivant :
                                                                                                                           ☐ le programme affiche "Faux"
   lignes suivantes?
                                                               int age = 18;
                                                                                                                           \square la variable y vaut 5
   while (1)
                                                               if (age < 18)
   {
                                                                                                                           \square la variable x vaut 5 et la variable y vaut 3
      printf("coucou\n");
                                                                    printf("Mineur\n");
                                                               }
                                                                                                                     19. Soit un programme contenant les lignes suivantes :
      □ il n'affiche rien
                                                               else
                                                               {
     □ il risque d'afficher bonjour à la place de coucou
                                                                    printf("Majeur\n");
                                                                                                                          int i = 0;
     □ il comporte une boucle infinie
                                                               }
                                                                                                                          int j = 0;
     \square il ne compile pas
                                                                                                                          for (i = 0; i < 2; i = i + 1)
13. Les lignes
                                                              affichera:
                                                                                                                               for (j = 0; j < 3; j = j + 1)
   int i;
                                                                □ Majeur
   int x=0;
                                                                ☐ Mineur
                                                                                                                                   printf("%d ", i);
   for(i=0,i<5,i=i+1)
                                                                   Majeur
                                                                                                                              }
   {
                                                                                                                          }
      x=x+1;
                                                                \square rien
                                                                □ Mineur
      \square ne comportent aucune erreur
                                                          17. Le bus système sert à :
                                                                                                                         qu'est ce qui sera affiché?
      □ comportent une erreur qui sera détectée au cours
                                                                ☐ Arriver à l'heure en cours
        de l'analyse syntaxique
                                                                                                                          □ transporter les processus du tourniquet au pro-
     □ comportent une erreur qui ne sera pas détectée
                                                                   cesseur
     □ comportent une erreur qui sera détectée au cours
                                                                                                                           de l'édition de lien
                                                                ☐ Écrire des données sur le dique dur
                                                                                                                           \Box 0 1 2 0 1 2
14. Si n est une variable entière pour demander sa valeur
                                                                ☐ Transférer des données et intructions entre pro-
   à l'utilisateur, on utilise plutôt :
                                                                   cesseur et mémoire
                                                                                                                          □ 1 2 1 2 3
      \square un débogueur
                                                          18. Après exécution jusqu'à la ligne 15 du programme C :
     □ printf("Valeur de n ? %g\n", n);
                                                                                                                     20. Un programme en langage C doit comporter une et une
                                                                 int main() {
                                                            10
     □ printf("Valeur de n ? %d\n", n);
                                                                                                                         seule définition de la fonction :
                                                            11
                                                                      int x = 5;
      □ scanf("%d", &n);
                                                            12
                                                                      int y = 3;
15. Pour compiler un programme prog.c, on utilise la
                                                                                                                           \square init
                                                            13
   ligne de commande :
                                                            14
                                                                      x = y;
                                                                                                                           \square main
     ☐ gcc -Wall prog.c -o prog.exe
                                                            15
     ☐ gcc -Wall prog.exe -o prog.c
                                                            16
                                                                                                                           □ begin
                                                            17
                                                                 }
     ☐ gcc prog.exe -Wall -o prog.c
                                                                                                                           \square include
     ☐ gcc prog.c -o -Wall prog.exe
                                                                \square la variable x vaut 3
```

 \Box C

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
1. 004.	

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

```
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 :
     □ alphabétique
     □ dans lequel ces fonctions sont appelées dans le
        main
     \square dans lequel vous avez déclaré ces fonction
     \square un ordre quelconque
7. 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 y vaut 5
     □ le programme affiche "Faux"
     \square la variable x vaut 3
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:
     ☐ ABCDEF
     □ cccccc
     \Box i
     □ A
9. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     \square ses blocs
     \square ses champs
```

 \square ses chants

```
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 :
      \Box 16
      \Box 4
      \Box 0
      \square 8
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 \mid | B) == (A \&\& !B)
     □ A && B
      \square (A == TRUE) && (B == TRUE)
      \square (!A || B)
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 6
      \Box 1
      \square 42
```

 \Box 0

```
18. Le bus système sert à :
13. Soit la fonction g définie par :
                                                                affichera:
    int g(int a)
                                                                  \Box 0123456
                                                                                                                              ☐ Arriver à l'heure en cours
                                                                  \Box 02468
                                                                                                                              □ transporter les processus du tourniquet au pro-
      printf("a = \n", %d);
                                                                  \Box 0 1 2 3 4 5 6 7
                                                                                                                                 cesseur
      if (1 > 0)
                                                                  \Box 0246
                                                                                                                              \Box Transférer des données et intructions entre pro-
                                                            16. Pour l'extrait de programme suivant :
                                                                                                                                 cesseur et mémoire
        return 5;
                                                                 int i;
                                                                                                                              ☐ Écrire des données sur le dique dur
      return 7;
                                                                 int j;
                                                                                                                        19. L'écriture 111 en binaire correspond au nombre natu-
                                                                 for(i=4;i>0;i=i-1)
                                                                                                                            rel:
    Alors l'expression g(0) prendra la valeur :
                                                                   for(j=i;j<6;j=j+1)
      \Box 5
                                                                                                                              \square 3
      \Box 7
                                                                                                                              □ 111
                                                                     printf("*");
      \Box 0
                                                                                                                              \square 7
14. Quel est le problème d'un programme comportant les
                                                                   printf(" ");
                                                                                                                              \square 8
    lignes suivantes?
    while (1)
                                                                                                                        20. Le code suivant :
                                                                qu'est ce qui sera affiché?
                                                                                                                             int age = 20;
      printf("coucou\n");
                                                                   ***** *** ***
                                                                                                                             if (age < 18)
                                                                   ** *** **** *****
      \square il n'affiche rien
                                                                                                                                  printf("Mineur\n");
      \square il comporte une boucle infinie
                                                                      **** **** ****
      \square il ne compile pas
                                                                                                                             printf("Majeur\n");
                                                            17. Si cette erreur apparaît à la compilation :
      \square il risque d'afficher bonjour à la place de coucou
                                                                                                                            affichera:
                                                                Undefined symbols : "_prinft" ou
15. Le code suivant :
                                                                référence indéfinie vers « prinft » que doit-
                                                                                                                              ☐ Mineur
                                                                on chercher dans le programme?
     int i;
                                                                                                                                 Majeur
     for (i = 0; i < 7; i = i + 2)
                                                                  \square une faute de frappe dans un appel de fonction
                                                                                                                              \square rien
                                                                  □ une directive préprocesseur #include manquante
         printf("%d ", i);
                                                                                                                              ☐ Mineur
                                                                  \square une variable non déclarée
                                                                                                                              □ Majeur
                                                                  \Box un caractère interdit en C
     printf("\n");
```

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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

- 1. Une de ces manière de composer les blocs de programmes ne fait pas partie des opérations de la programmation structurée :
 - \Box répéter un bloc tant qu'une condition est vérifée
 - □ retourner un bloc
 - $\square\,$ sélectionner entre deux blocs à l'aide d'une condition
 - \square mettre les blocs en séquence les uns à la suite des autres
- 2. 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)
- 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
- \Box la variable x vaut 3
- \Box la variable y vaut 5
- $\Box\:$ le programme affiche "Faux"
- 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 :

□ 4
□ 0
```

- \square 0
- □ 1
- 5. Vous utilisez une boucle while quand :
 - \Box l'incrément de la variable de boucle n'est pas 1
 - $\hfill \square$ vous n'avez pas déclaré de fonction
 - □ vous avez déjà fait un for dans le même programme principal
 - \Box vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
- 6. Pour compiler un programme prog.c, on utilise la ligne de commande :
 - $\hfill\Box$ gcc prog.c -o -Wall prog.exe
 - \square gcc -Wall prog.c -o prog.exe
 - $\hfill\Box$ gcc prog.exe -Wall -o prog.c
 - ☐ gcc -Wall prog.exe -o prog.c
- 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 :

- □ a == b
- \Box a = b
- \Box (a.n == b.n) && (a.x == b.x)
- \square a{n, x} == b{n, x}
- 8. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
 - $\Box\,$ analyse sémantique
 - \Box analyse lexicale
 - \Box analyse harmonique
 - $\hfill\Box$ analyse syntaxique

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. 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);
    }
}</pre>
```

qu'est ce qui sera affiché?

- \Box 1 2 1 2 3
- $\square \ 0 \ 0 \ 0 \ 1 \ 1 \ 1$
- \square 0 1 2 0 1 2
- \square 0 1 0 1 0 1 0 1

```
14. Pour déclarer une fonction factorielle qui prend en
11. Si le code:
                                                                                                                        17. Pour déclarer une procédure afficher_date qui prend
                                                                argument un entier et renvoie sa factorielle on écrit :
                                                                                                                            en argument un struct date_s et affiche le contenu
   struct toto_s
                                                                                                                            du struct, on écrit :
                                                                  \square int factorielle(int x);
                                                                                                                              □ void afficher_date(date_s d);
      int n;
                                                                  ☐ int factorielle(double n);
      double x;
                                                                                                                              □ void afficher_date(struct date_s d);
                                                                  □ struct int factorielle(int n);
   };
                                                                                                                              ☐ struct date_s afficher_date(struct date_s d);
                                                                  ☐ int factorielle();
   précède la fonction main(), alors on peut écrire en
                                                                                                                              ☐ int afficher_date(date_s d);
   début de main() :
                                                            15. On considère deux variables booléennes A et B initia-
                                                                                                                        18. Un registre du processeur est :
      \Box int struct toto_s = {3, -1e10};
                                                                lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                                                                              \square un composant qui contient la liste des fichiers du
                                                                pressions booléennes suivantes, laquelle a pour valeur
      \Box toto_s struct z = {3, 0.5};
                                                                                                                                 système
                                                                TRUE?
      □ struct toto_s toto;
                                                                                                                              □ une unité de calcul spécialisée de l'ordinateur
                                                                  □ A && B
      \square toto_s n, x;
                                                                                                                              \square une gamme de fréquence de fonctionnement du
      \square int toto.n = 3;
                                                                  \square (A == TRUE) && (B == TRUE)
                                                                                                                                 processeur
                                                                  \square (!A || B)
12. Au début de la fonction main() on place le code :
                                                                                                                              □ une case mémoire interne au processeur qui sera
                                                                                                                                 manipulée directement lors des calculs
                                                                 \square !(!A \mid | B) == (A \&\& !B)
     char b = 'A';
     b = b + 2;
                                                                                                                        19. Un enregistrement permet de grouper plusieurs valeurs
                                                            16. Soit la fonction f définie par :
     printf("%c\n", b);
                                                                                                                            dans:
                                                                int f(int a)
   Alors l'affichage sera :
                                                                                                                              □ ses chants
      □ b
                                                                                                                              \square ses champs
                                                                  printf("a = \n", %d);
      □В
                                                                  if (a > 0)
                                                                                                                              \square ses blocs
                                                                  {
     □ A
                                                                                                                              □ ses cases
                                                                    return 3;
      \Box C
                                                                                                                        20. Si cette erreur apparaît à la compilation :
13. Pour déclarer une fonction saisie_utilisateur qui
                                                                                                                            erreur: conflicting types for 'max', que doit-
                                                                  return 4;
   demande à l'utilisateur d'entrer un entier au clavier et
                                                                                                                            on chercher dans le programme?
   renvoie cet entier on écrit :
                                                                                                                              \square une fonction appelée avant sa déclaration
                                                                Alors l'expression f(0) prendra la valeur :
      □ saisie_utilisateur(scanf(%d));
                                                                                                                              \square une fonction déclarée mais non définie
                                                                  \Box 0
     □ void saisie_utilisateur(char c);
                                                                                                                             □ une directive préprocesseur #include manquante
                                                                  \square 3
     ☐ int saisie_utilisateur();
                                                                                                                              \square un désaccord entre la déclaration et la définition
     □ void saisie_utilisateur(int n);
                                                                  \Box 4
                                                                                                                                 d'une fonction
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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 == b
     \square a{n, x} == b{n, x}
     \square (a.n == b.n) & (a.x == b.x)
     \square a = b
2. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
     \square int factorielle(int x);
     □ struct int factorielle(int n);
     ☐ int factorielle();
     ☐ int factorielle(double n);
3. Sous unix (ou linux), la commande 1s permet de :
     \square afficher le contenu d'un fichier texte
     □ afficher la liste de fichiers contenus dans un
        répertoire
     □ voir des clips musicaux
     \square compiler un programme
4. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
```

Undefined symbols : "_prinft" ou

☐ l'édition de liens

 \square l'analyse harmonique

☐ l'analyse sémantique

□ l'analyse des entrées clavier

référence indéfinie vers « prinft »

```
5. 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
6. Le code suivant :
   int i;
   for (i = 8; i > 0; i = i - 2)
        printf("%d ", i);
   printf("\n");
  affichera:
    \square 8 2
    \Box 02468
    \Box 8 6 4 2 0
    \Box 8 6 4 2
7. Le code suivant :
   int age = 15;
   if (age < 18)
   {
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
  affichera:
    □ Majeur
    ☐ Mineur
       Majeur
    ☐ Mineur
     □ rien
```

```
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
      \Box 0
      \square 3
 9. Quels calculs peut-on programmer en programmation
    structurée?
      ☐ il y a des calculs programmables en langage ma-
         chine et qui ne sont pas programmables en pro-
         grammation structurée
      □ en programmation structurée on peut program-
         mer tous les calculs programmables en langage
         machine
      \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
10. 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";
11. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses champs
      \square ses blocs
      \square ses chants
      □ ses cases
```

```
15. Pour déclarer une fonction pgcd qui calcule et renvoie
12. Si le code :
                                                                                                                              }
                                                                le plus grand diviseur commun de deux entiers positifs
   struct toto_s
                                                                passés en arguments on écrit :
                                                                                                                               else
                                                                                                                              {
      int n;
                                                                  \square int pgcd(int x, y);
      double x;
                                                                  \Box int pgcd(int x, int x);
                                                                                                                              }
   };
                                                                  \square void pgcd(int x, int y);
   précède la fonction main(), alors on peut écrire en
                                                                  \square int pgcd(int y, int x);
   début de main():
                                                                                                                              affichera:
     \square int struct toto_s = {3, -1e10};
                                                            16. Pour déclarer une fonction exposant qui prend en ar-
                                                                                                                               ☐ Mineur
     □ struct toto_s toto;
                                                                 gument un réel x et un entier positif n et renvoie la
                                                                                                                               □ Majeur
                                                                valeur de x^n on écrit :
     \square toto_s struct z = {3, 0.5};
                                                                                                                               ☐ Mineur
                                                                  \square void exposant(double x^n);
     \square toto_s n, x;
                                                                                                                                  Majeur
      \square int toto.n = 3;
                                                                  \square exposant(double x, int n, int r);
                                                                                                                               \square rien
13. Pour l'extrait de programme suivant :
                                                                  \square double exposant(double x, int n);
     int i = 0;
                                                                  \square int exposant(double n, int x);
                                                                                                                         19. Un bit est:
     int j = 0;
                                                            17. Soit la fonction g définie par :
     for (i = 0; i < 3; i = i + 1)
                                                                int g(int a)
         for (j = 0; j < 2; j = j + 1)
                                                                   printf("a = \n", %d);
              printf("%d ", i);
                                                                   if (1 > 0)
         }
                                                                   {
                                                                                                                         20. Le code suivant :
                                                                     return 5;
     printf("\n");
                                                                                                                               int i;
   qu'est ce qui sera affiché?
                                                                   return 7;
     \Box 1 2 3 1 2
     \Box 0 0 1 1 2 2
                                                                Alors l'expression g(0) prendra la valeur :
      \Box 0 1 2 0 1 2
                                                                  \Box 7
      \Box 0 1 0 1 0 1
                                                                  \Box 0
                                                                                                                              affichera:
14. Si n est une variable entière pour demander sa valeur
                                                                  \Box 5
   à l'utilisateur, on utilise plutôt :
                                                                                                                               \Box \ 4\ 3\ 2\ 1\ 0
      \square un débogueur
                                                            18. Le code suivant :
                                                                                                                               \Box 01234
     □ printf("Valeur de n ? %d\n", n);
                                                                 int age = 20;
                                                                                                                               \square 1 2 3 4
      □ scanf("%d", &n);
                                                                 if (age < 18)
     \square printf("Valeur de n ? %g\n", n);
                                                                                                                               \square 4 3 2 1
                                                                 {
```

```
printf("Mineur\n");
    printf("Majeur\n");
\square un chiffre binaire (0 ou 1)
☐ l'instruction qui met fin à un programme
□ la longueur d'un mot mémoire
□ un battement d'horloge processeur
for (i = 4; i \ge 0; i = i - 1)
    printf("%d ", i);
printf("\n");
```

т.			-4
1.1	cer	000	- 1

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 :
 - □ 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
- 2. 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 1 0 1 0 1 0 1
- \Box 0 1 2 0 1 2
- \Box 0 0 0 1 1 1
- 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 16
```

 \square 0 \square 8

 \Box 4

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 :

```
\Box 0
```

 \square 7

 \square 5

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=8, b=5
```

☐ f(3,5)=8, a=	=3,	b=5
----------------	-----	-----

\square f(a,b)=13, a	=8, b=5
------------------------	---------

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

6. 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();
- ☐ char afficher_menu(printf("menu")); □ void afficher_menu();
- 7. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?

```
□ char 'c';
```

- ☐ char "c";
- ☐ int char;
- \Box char c;
- 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=%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. 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 la variable x vaut 16
- \square la variable x vaut $-\frac{1}{2}$ \square le programme affiche x
- 10. Une variable booléenne est un variable :
 - \Box jamais nulle
 - ☐ réelle positive
 - □ NaN (not a number, qui n'est pas un nombre)
 - □ à laquelle une valeur vient d'être affectée
 - \square qui est vraie ou fausse

```
11. 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
12. 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, y);
      \square int pgcd(int x, int x);
13. 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
```

```
14. Laquelle des analyses suivantes ne fait pas partie des
                                                               18. On considère deux variables booléennes A et B initia-
    étapes de la compilation :
                                                                   lisées à TRUE et FALSE respectivement. Parmi les ex-
                                                                   pressions booléennes suivantes, laquelle a pour valeur
      \square analyse harmonique
                                                                   TRUE?
      □ analyse sémantique
      \square analyse lexicale
                                                                     \square !(!A || B) == (A && !B)
      \square analyse syntaxique
                                                                     □ A && B
15. Le type des réels en C est :
                                                                     \square (A == TRUE) && (B == TRUE)
      ☐ double
                                                                     \square (!A || B)
      \square int
      \square real
                                                               19. Le code suivant :
      □ char
                                                                    int i;
16. Soit la fonction f définie par :
                                                                    for (i = 4; i > 0; i = i - 1)
    int f(int a)
                                                                         printf("%d ", i);
      printf("a = \n", %d);
      if (a > 0)
                                                                    printf("\n");
      {
                                                                   affichera:
         return f(a - 1) + 1;
                                                                     \Box 01234
      return 4;
                                                                     \square 0 1 2 3
                                                                     \square 4 3 2 1
    Alors l'expression f(1) prendra la valeur :
      \Box 0
                                                                     \Box \ 4\ 3\ 2\ 1\ 0
      \Box 5
                                                               20. Un fichier source est:
      \Box 1
                                                                     □ un document qui doit être protégé
      \Box 4
17. Pour déclarer une fonction exposant qui prend en ar-
                                                                     \square un fichier que l'ont doit citer dans les documents
    gument un réel x et un entier positif n et renvoie la
                                                                        produits sur l'ordinateur
    valeur de x^n on écrit :
                                                                     □ un document de référence du système
      \square void exposant(double x^n);
                                                                     \square un document illisible pour les humains
      \square double exposant(double x, int n);
      \square exposant(double x, int n, int r);
                                                                     □ un fichier texte qui sera traduit en instructions
                                                                        processeur
      \square int exposant(double n, int x);
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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 ! □ <> □!= $\square \neq$ 2. 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) 3. 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 $\Box 0246$ \Box 0 1 2 3 4 5 6 7 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 ces fonctions sont appelées dans le main □ alphabétique \square dans lequel vous avez déclaré ces fonction

 \square un ordre quelconque

```
5. 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 1 2 0 1 2
     \Box 0 1 0 1 0 1 0 1
     \Box 0 0 0 1 1 1
6. 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();
7. 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
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 y, int x);
    \square void pgcd(int x, int y);
    \Box int pgcd(int x, int x);
     \square int pgcd(int x, y);
```

```
9. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     printf("Majeur\n");
    affichera:
     □ Mineur
        Majeur
     □ Mineur
     □ Majeur
     □ rien
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 :
     \square 3
     \Box 0
     \Box 4
11. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée et définie
      □ l'avoir déclarée
     □ avoir défini une constante symbolique de la taille
        de cette fonction
      ☐ l'avoir définie
```

```
12. Le code suivant :
     int i;
     for (i = 8; i > 0; i = i - 2)
         printf("%d ", i);
     }
     printf("\n");
    affichera:
      \square 8 2
      \Box 02468
      \Box 8 6 4 2 0
      \square 8 6 4 2
13. Le code suivant :
     int i;
     for (i = 4; i \ge 0; 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
14. 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 n = pgcd();
      \square int pgcd(2);
```

```
18. Laquelle de ces écritures correspond à la déclaration
15. Soit un programme contenant les lignes suivantes :
                                                                d'une variable de type caractère en langage C?
     int i = 0;
     int j = 0;
                                                                  \Box char c;
     for (i = 0; i < 3; i = i + 1)
                                                                  ☐ int char;
         for (j = 0; j < 5; j = j + 1)
                                                                  ☐ char "c";
                                                                 □ char 'c';
         }
                                                            19. Pour déclarer une fonction exposant qui prend en ar-
                                                                gument un réel x et un entier positif n et renvoie la
     printf("j = %d\n", j);
                                                                valeur de x^n on écrit :
   qu'est ce qui sera affiché par ce printf?
                                                                  \square int exposant(double n, int x);
     \Box j = %d
                                                                  \square exposant(double x, int n, int r);
     \Box j = 4
                                                                  \square void exposant(double x^n);
     \Box j = 0
     \Box j = 5
                                                                  \square double exposant(double x, int n);
16. Si le code:
                                                            20. Pour l'extrait de programme suivant :
    struct toto_s
                                                                 int i = 0;
      int n;
                                                                 int j = 0;
      double x;
                                                                 for (i = 0; i < 3; i = i + 1)
    précède la fonction main(), alors on peut écrire en
                                                                      for (j = 0; j < 2; j = j + 1)
    début de main() :
     \square toto_s n, x;
                                                                          printf("%d ", i);
                                                                      }
     \Box toto_s struct z = {3, 0.5};
      \square int struct toto_s = {3, -1e10};
                                                                 printf("\n");
      \square int toto.n = 3;
      ☐ struct toto_s toto;
                                                                qu'est ce qui sera affiché?
17. Un enregistrement permet de grouper plusieurs valeurs
                                                                  \Box 0 1 0 1 0 1
    dans:
      \square ses cases
                                                                  \Box 1 2 3 1 2
     \square ses blocs
                                                                  \Box 0 1 2 0 1 2
     \square ses champs
                                                                  \Box 0 0 1 1 2 2
      \square ses chants
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 qui doit être protégé □ 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 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(char c); □ void saisie_utilisateur(int n); □ saisie_utilisateur(scanf(%d)); ☐ int saisie_utilisateur(); 3. Avant de faire appel à une fonction il est nécessaire de: □ avoir défini une constante symbolique de la taille de cette fonction □ l'avoir définie □ l'avoir déclarée □ l'avoir déclarée et définie 4. Le langage C est un langage □ compilé □ interprété □ composé □ lu, écrit, parlé 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 = 4
    \Box j = 5
    \Box j = 0
    \Box j = %d
6. Après la déclaration : int mccarthy(int n);, il est
  correct d'écrire:
    \square n = mccarthy(p, q);
    \square n = mccarthy();
    \square int mccarthy(int 2);
    \square x = mccarthy(n);
7. 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:
    \square i
    ☐ ABCDEF
    \square A
    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. 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
10. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \Box 7
     \square 111
     \square 8
     \square 3
11. 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?
     *** **** ****
```

```
12. Si factorielle est une fonction prenant en entrée un
                                                                 précède la fonction main(), alors on peut écrire en
                                                                                                                                \Box 0
   entier et renvoyant un entier, il est correct d'écrire :
                                                                 début de main() :
                                                                                                                                \Box 6
      \square n = factorielle();
                                                                   \square toto_s n, x;
                                                                                                                                \square 42
      \square n = factorielle(p, q);
                                                                   \square int struct toto_s = {3, -1e10};
      ☐ printf("%d", factorielle(n));
                                                                   ☐ struct toto_s toto;
                                                                                                                          19. Pour déclarer une variable qui sera utilisée comme va-
      ☐ int factorielle(int 2);
                                                                   \square int toto.n = 3;
                                                                                                                              riable de boucle on peut utiliser l'instruction
13. Dans la commande gcc, l'option -Wall signifie :
                                                                   \Box toto_s struct z = {3, 0.5};
      \square qu'on veut changer alétoirement de fond d'écran
                                                                                                                                \square int k;
                                                             17. Une segmentation fault est une erreur qui survient
      □ qu'il faut lancer un déboggueur
                                                                 lorsque:
                                                                                                                                \square loop i;
      ☐ que l'on veut voir tous les avertissements
                                                                   ☐ le programme source a été enregistré sur le disque
                                                                                                                                \square int %d;
                                                                      dur en plusieurs morceaux et l'un d'entre eux ne
      □ qu'il faut indenter le fichier source
                                                                      peut pas être chargé par le compilateur
14. Afin de représenter la taille d'un tableau, définir une
                                                                                                                                \square int loop n;
                                                                   □ le programme tente d'accèder à une partie de la
   constante symbolique N valant 3.
                                                                      mémoire qui ne lui est pas réservée
                                                                                                                          20. Soit la fonction g définie par :
      \square #define taille = N
                                                                   □ la division du programme en zones homogènes
      \square #define taille = 3
                                                                      échoue
                                                                                                                              int g(int a)
      \square #define N 3
                                                                   □ le programme tente d'afficher des caractères sur
      \square #define N = 3
                                                                                                                                printf("a = \n", %d);
                                                                      une ligne qui va au delà de la largeur de la fenêtre
                                                                                                                                if (1 > 0)
15. Lorsqu'un programme utilise printf ou scanf il faut
                                                                      du terminal
                                                                                                                                {
   qu'il contienne l'instruction préprocesseur :
                                                             18. Le code suivant :
                                                                                                                                   return 5;
      ☐ #appart <stdlib.h>
                                                                  int somme = 0;
      ☐ #include <studlib.h>
                                                                  int i;
                                                                                                                                return 7;
      ☐ #include <stdio.h>
                                                                  for (i = 1; i < 4; i = i + 1)
      ☐ #include <studio.h>
                                                                                                                              Alors l'expression g(0) prendra la valeur :
                                                                    somme = somme + i;
16. Si le code:
   struct toto_s
                                                                                                                                \Box 7
                                                                  printf("%d", somme);
   {
                                                                                                                                \Box 0
      int n;
                                                                 affichera:
      double x;
                                                                                                                                \Box 5
   };
                                                                   \Box 1
```

Éléments d'informatique – contrôle continue

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

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:
     \square 42
     \square 0
    \Box 6
     \Box 1
2. 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
    □ la variable y vaut 5
     ☐ le programme affiche "Faux"
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
    \square 3
    \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)=8, a=3, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=13, a=8, b=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 :
    □ dans lequel ces fonctions sont appelées dans le
    □ dans lequel vous avez déclaré ces fonction
    □ alphabétique
     \square un ordre quelconque
6. 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
     \Box 8 6 4 2 0
     \square 8 2
7. 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
     \square une faute de frappe dans un appel de fonction
     ☐ une directive préprocesseur #include manquante
     □ un caractère interdit en C
8. 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
 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. 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 afficher_menu(int char);
```

11. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande : □ mkdir TP4 □ new TP4 □ kwrite TP4	15. Après exécution jusqu'à la ligne 15 du programme C: 10 11 int main() { 12 int x = 5; 13 14 x = 3 * x + 1;	
☐ yppasswd 12. Le code suivant : int i; for (i = 0; i < 5; i = i + 1) { printf("%d ", i); } printf("\n"); affichera : ☐ 4 3 2 1 ☐ 0 1 2 3 ☐ 0 1 2 3 4	15 16 17 } □ le programme affiche **** □ la variable x vaut 16 □ la variable x vaut −½ □ le programme affiche x 16. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit- on chercher dans le programme?	
☐ 4 3 2 1 0 13. 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'édition de liens ☐ l'analyse des entrées clavier 14. Le type des réels en C est : ☐ int ☐ real ☐ double ☐ char	□ 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 17. 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 == TRUE) && (B == TRUE) □ A && B	

```
18. 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 0246
     \Box 0 1 2 3 4 5 6
19. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     \square créer un fichier texte
     □ changer de répertoire courant
     □ ouvrir un fichier texte
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 :
```

 $\begin{array}{c} \square \ 0 \\ \square \ 7 \\ \square \ 5 \end{array}$

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 0 1 1 2 2
    \Box 1 2 3 1 2
     \Box 0 1 2 0 1 2
     \Box 0 1 0 1 0 1
2. 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
    □ l'analyse des entrées clavier
     □ l'édition de liens
3. 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
4. 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
     \Box 8 6 4 2
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=1;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
6. 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 1 2 3 4
     \square 4 3 2 1
7. 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);
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:
      □ A
      \Box i
      ☐ ABCDEF
      \Box ccccc
 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 0
      \square 7
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 x = 2
      \square le terminal affiche 5
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, y);
      \square int pgcd(int y, int x);
```

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

12. Le type des réels en C est :	$\hfill\Box$ l'incrément de la variable de boucle n'est pas 1	18. Sous unix (ou linux), pour créer un répertoire TP
□ char	□ vous avez déjà fait un for dans le même programme principal	dans le répertoire courant on peut utiliser la com mande :
☐ double	16. Soit un programme contenant les lignes suivantes :	□ kwrite TP4
\square real	1	
\square int	int i = 0; int j = 0;	□ new TP4
13. Vous avez déclaré préalablement un ensemble de fonc-	for (i = 0; i < 2; i = i + 1)	\square mkdir TP4
tions utilisées par votre programme principal. L'ordre	{	\square yppasswd
dans lequel vous devez maintenant définir ces fonctions est l'ordre :	for (j = 0; j < 3; j = j + 1) {	19. Au début de la fonction main() on place le code :
\Box dans lequel vous avez déclaré ces fonction	<pre>printf("%d ", i);</pre>	char $b = 'A';$
\Box dans le quel ces fonctions sont appelées dans le main	}	b = b + 2; printf("%c\n", b);
\square un ordre quelconque		Alors l'affichage sera :
□ alphabétique	qu'est ce qui sera affiché?	□ b
14. Pour déclarer une procédure afficher_date qui prend		□ С
en argument un struct date_s et affiche le contenu		□В
du struct, on écrit :	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	□ A
☐ void afficher_date(date_s d);		
\square struct date_s afficher_date(struct date_s	d) 17. Si racine est une fonction prenant en entrée un réel et renvoyant la racine carrée de cet réel, et que x est	20. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit
\square void afficher_date(struct date_s d);	une variable réelle définie et initialisée, il est incorrect	on chercher dans le programme?
☐ int afficher_date(date_s d);	d'écrire: $\square x = racine(x * x) - racine(x);$	\Box une accolade manquante
15. Vous utilisez une boucle while quand:		une accolade en trop
$\hfill \square$ vous ne connaissez pas le nombre d'itérations de	$\square x = racine(racine(x)*racine(x));$	<u>-</u>
la boucle à l'avance	$\square x = racine(2/3);$	un point-virgule manquant
$\hfill \square$ vous n'avez pas déclaré de fonction	$\square x - 1 = racine(x);$	\square un point-virgule en trop

Éléments d'informatique – contrôle continue

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

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=0;i<=5;i=i+1)</li>
```

```
☐ 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)
```

2. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit :

```
□ void pgcd(int x, int y);
□ int pgcd(int x, int x);
□ int pgcd(int x, y);
```

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

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 :

```
\begin{array}{c}
\square \ 0 \\
\square \ 4 \\
\square \ 3
\end{array}
```

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)
    {
        ...</pre>
```

```
}
   }
   printf("j = %d\n", j);
   }
  qu'est ce qui sera affiché?
    \Box j = %d
    \Box j = 5
    \Box i = 0
    \Box j = 4
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)=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
6. 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-
```

☐ l'incrément de la variable de boucle n'est pas 1

gramme principal

```
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 -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
 8. 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 directive préprocesseur \#include manquante
     \square une fonction déclarée mais non définie
     ☐ une fonction appelée avant sa déclaration
9. 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 directive préprocesseur #include manquante
     \square une faute de frappe dans un appel de fonction
10. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <studlib.h>
     ☐ #appart <stdlib.h>
     ☐ #include <stdio.h>
     ☐ #include <studio.h>
11. Après exécution jusqu'à la ligne 14 du programme C:
       int main() {
 10
 11
           int x = 5;
 12
           printf(" x = %d\n", 2);
 13
 14
 15
      }
 16
     \square le terminal affiche x = 2
     □ le terminal affiche "Faux"
     □ le terminal affiche 5
     \square le terminal affiche x = 5
```

```
12. Le code suivant :
                                                                  □ retourner un bloc
                                                                                                                            Quelle est la condition cond:
                                                                  □ répéter un bloc tant qu'une condition est vérifée
     int age = 15;
                                                            15. Si factorielle est une fonction prenant en entrée un
     if (age < 18)
                                                                entier et renvoyant un entier, il est correct d'écrire :
         printf("Mineur\n");
                                                                  ☐ printf("%d", factorielle(n));
     }
                                                                  \square n = factorielle(p, q);
     else
                                                                  ☐ int factorielle(int 2);
                                                                                                                        1
     {
                                                                  \square n = factorielle();
         printf("Majeur\n");
                                                            16. Le code suivant :
     }
                                                                 int i;
                                                                 for (i = 1; i < 5; i = i + 1)
    affichera:
      □ Mineur
                                                                      printf("%d ", i);
         Majeur
      □ Majeur
                                                                                                                        1
                                                                 printf("\n");
      \square rien
                                                                affichera:
      □ Mineur
                                                                  \Box \ 4\ 3\ 2\ 1\ 0
13. Si n est une variable entière pour demander sa valeur
                                                                  \square 1 2 3 4
    à l'utilisateur, on utilise plutôt :
                                                                  \Box 01234
      □ printf("Valeur de n ? %d\n", n);
                                                                  \square 4 3 2 1
     □ printf("Valeur de n ? %g\n", n);
                                                            17. On souhaite faire une boucle de contrôle de saisie : tant
                                                                que l'entier n n'appartient pas à l'intervalle [a..b], on
      □ scanf("%d", &n);
                                                                recommence la saisie de n. Soit le programme suivant :
      \Box un débogueur
                                                                 int a = 0;
14. Une de ces manière de composer les blocs de pro-
                                                                 int b = 20;
    grammes ne fait pas partie des opérations de la pro-
                                                                 int n;
    grammation structurée :
                                                                 scanf("%d", &n);
      \Box\, sélectionner entre deux blocs à l'aide d'une condi-
                                                                                                                              □ jouer de la musique
                                                                 while(cond)
         tion
                                                                 {
                                                                                                                              □ changer de répertoire courant
     \square mettre les blocs en séquence les uns à la suite des
                                                                   scanf("%d", &n);
                                                                                                                              □ ouvir un bureau partagé (common desktop)
         autres
```

	·
	□ a<=n<=b
	□ (a <n) (n="" ="">b)</n)>
	□ (n<=a) && (n<=b)
	☐ (a<=n) && (n<=b)
8.	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;
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);
	\square struct date_s afficher_date(struct date_s d)
0.	Sous unix (ou linux), la commande cd permet de :
	\Box détruire un fichier
	$\hfill \square$ récupérer un programme arrêté avec la commande $\verb"ab"$

Éléments d'informatique – contrôle continue

4. Un fichier source est:

Prénom:	Nom:	
N° etu :		

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 n = pgcd(n, 3);
     \square int n = pgcd();
     \square int pgcd(2);
     \square n = pgcd(int p, int q);
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. Vous utilisez une boucle while quand:
     □ 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
     □ vous ne connaissez pas le nombre d'itérations de
        la boucle à l'avance
```

```
☐ 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
    □ un document de référence du système
     \square un fichier texte qui sera traduit en instructions
       processeur
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
      }
     \square la variable x vaut 5 et la variable y vaut 0
    \Box la variable x vaut 0
    ☐ le programme affiche "Faux"
    \square la variable y vaut 5
6. 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
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:
    □ dans lequel vous avez déclaré ces fonction
    □ alphabétique
    □ dans lequel ces fonctions sont appelées dans le
       main
     \square un ordre quelconque
```

etu	<u> </u>
8.	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
	\Box certains programmes sont de vrais plats de spaghetti
	□ il y 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
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(char c);
	☐ int saisie_utilisateur();
	☐ void saisie_utilisateur(int n);
10.	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	\Box un point-virgule en trop
	$\hfill\Box$ un point-virgule man quant
	\Box une accolade en trop
	\Box une accolade man quante
11.	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);
12.	Après exécution du programme :
-	l lecture 8 r0
	valeur 3 r1
	3 mult r1 r0

valeur 1 r2

```
add r2 r0
       ecriture r0 8
       stop
  8
      5
     □ la case mémoire 8 contiendra 16
     \square la case mémoire 8 contiendra 0
     \square le terminal affiche 8
     \square le bus explose
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 16
     \Box 4
     \Box 0
14. 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
15. 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
     \square printf("Valeur de n ? %g\n", n);
```

```
16. 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?
17. 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 des entrées clavier
     ☐ l'analyse sémantique
     □ l'édition de liens
18. 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;
     \Box toto_s struct z = {3, 0.5};
      \square toto_s n, x;
     \square int toto.n = 3;
19. 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
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 0
      \Box 7
      \Box 5
```

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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. 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 définie
```

□ l'avoir déclarée et définie

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)=13, 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)=8, a=8, b=5
```

4. 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
   \Boxla variable x vaut 5 et la variable y vaut 0
    \square la variable y vaut 5
    ☐ le programme affiche "Faux"
```

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
```

- □ 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. 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 \mid | B) == (A \&\& !B)
```

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

```
\square 3
□ 111
\square 8
\square 7
```

8. Un enregistrement permet de grouper plusieurs valeurs dans:

```
\square ses chants
     □ ses cases
     \square ses blocs
     \square ses champs
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

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

 $\Box 0123$

10. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés :

chacun son	tour,	après	que le	processus	précéden
a terminé					

 \square tous ensemble

□ en parallèle, chacun dans un registre

□ tour à tour, un petit peu à chaque fois

11. 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?	14. Pour déclarer une fonction saisie_utilisateur qui	Alors l'affichage sera :
□ j = 5	demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :	□ сссссс
□ j = %d	□ void saisie_utilisateur(char c);	□ A
\Box j = 0	☐ int saisie_utilisateur();	☐ ABCDEF
□ j = 4	☐ saisie_utilisateur(scanf(%d));	□ i
12. Si x est une variable réelle (de type double) alors	\square void saisie_utilisateur(int n);	18. Pour déclarer une variable qui sera utilisée comme va-
x = 3/2 lui affecte la valeur :	15. Soient deux variables entières x et y initialisées à 4 et	riable de boucle on peut utiliser l'instruction
□ 1	5 respectivement. L'affichage x=4 et y=5 est obtenu	☐ int k;
$\square \ 0.5$	avec la commande :	\square int loop n;
\square 1.5	<pre>□ printf("x=%d et y=%d\n",x,y);</pre> <pre>□ printf("x=%d et y=%d\n",x y);</pre>	☐ int %d;
\Box 0	☐ printf(x=%x et y=%y\n");	☐ loop i;
13. Soit la fonction g définie par :	☐ printf("x=%d et y=%d\n,x,y");	19. Un registre du processeur est :
int g(int a)	16. Vous utilisez une boucle while quand:	\Box une case mémoire interne au processeur qui sera
{	☐ l'incrément de la variable de boucle n'est pas 1	manipulée directement lors des calculs
<pre>printf("a = \n", %d); if (1 > 0)</pre>	□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance	☐ un composant qui contient la liste des fichiers du système
{	□ vous n'avez pas déclaré de fonction	une unité de calcul spécialisée de l'ordinateur
<pre>return 5; }</pre>	□ vous avez déjà fait un for dans le même pro- gramme principal	☐ une gamme de fréquence de fonctionnement du processeur
return 7;	17. Au début de la fonction main() on place le code :	•
}	char i;	20. Le langage C est un langage
Alors l'expression g(0) prendra la valeur :	for (i = 'A'; i <= 'F'; i = i + 1)	\square lu, écrit, parlé
\Box 5	{	□ compilé
□ 7	<pre>printf("%c", i);</pre>	\square interprété
\Box 0	<pre>printf("\n");</pre>	\square composé

{

x=x+1;

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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

1. Les lignes
int i;
int x=0;

 $\hfill\Box$ ne comportent aucune erreur

for(i=0,i<5,i=i+1)

□ comportent une erreur qui ne sera pas détectée
 □ comportent une erreur qui sera détectée au cours

de l'édition de lien

 $\hfill \Box$ comportent une erreur qui sera détectée au cours de l'analyse syntaxique

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 :

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

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 :

□ 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");
□ printf("x=%d et y=%d\n,x,y");

4. Si cet avertissement apparaı̂t 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 appelée avant sa déclaration

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

☐ une directive préprocesseur #include manquante

5. Si cette erreur apparaît à la compilation :
error: expected ';' before '}' token que doiton chercher dans le programme?
□ un point-virgule manquant
□ un point-virgule en trop

6. Quels calculs peut-on programmer en programmation structurée?

☐ une accolade manquante

 \square une accolade en trop

□ 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

 \Box il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine

 \Box certains programmes sont de vrais plats de spaghetti

7. Le code suivant :

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

affichera:

☐ Mineur
Majeur
☐ Mineur

□ Majeur
□ rien

8. Dans la commande gcc, l'option -Wall signifie :

☐ que l'on veut voir tous les avertissements
☐ qu'il faut indenter le fichier source

□ qu'il faut lancer un déboggueur

9. 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

□ qu'on veut changer alétoirement de fond d'écran

 $\Box\,$ un ordre quel conque

 \Box alphabétique

 $\Box\,$ dans lequel vous avez déclaré ces fonction

10. Le code suivant :

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

affichera : \Box rien

□ Majeur

□ Mineur Majeur

 \square Mineur

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

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

```
12. Le code suivant :
     int i:
     for (i = 0; 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 0123
13. 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 struct z = {3, 0.5};
     \square toto_s n, x;
     \Box int struct toto_s = {3, -1e10};
      \square int toto.n = 3;
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, x} == b{n, x}
      □ a == b
      \square (a.n == b.n) && (a.x == b.x)
      \Box a = b
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 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);
16. Vous utilisez une boucle while quand:
     □ vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      \square 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
17. 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
```

```
18. 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(int n);
     □ saisie_utilisateur(scanf(%d));
     □ void saisie_utilisateur(char c);
19. Le type des réels en C est :
     □ double
     \square real
     □ char
     \square int
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)=8, a=3, b=5
     \Box f(a,b)=8, a=8, b=5
```

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

Éléments d'informatique – contrôle continue

Prénom: Nom:	
N° etu :	

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(); \square n = mccarthy(p, q); 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)=8, a=8, b=5 \Box f(3,5)=8, a=3, b=5 \Box f(a,b)=13, a=8, b=5 3. 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 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=%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");

```
5. 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:
    □ A
    \Box i
    □ cccccc
    ☐ ABCDEF
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 :
    □ alphabétique
    □ dans lequel ces fonctions sont appelées dans le
        main
    \square un ordre quelconque
    □ dans lequel vous avez déclaré ces fonction
7. 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);
8. 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'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
```

```
9. 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 n'avez pas déclaré de fonction
     □ vous avez déjà fait un for dans le même pro-
        gramme principal
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. 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 3
      \square 20
      \Box 16
12. Si cette erreur apparaît à la compilation :
    error: expected ';' before '}' token que doit-
   on chercher dans le programme?
      \Box une accolade en trop
     □ un point-virgule en trop
      ☐ un point-virgule manquant
      □ une accolade manquante
```

```
13. Soit la fonction f définie par :
                                                             15. Un fichier source est:
                                                                                                                                \Box C
                                                                  \square un document illisible pour les humains
   int f(int a)
                                                                                                                                □В
   {
                                                                  \square un fichier que l'ont doit citer dans les documents
                                                                                                                                □ b
      printf("a = \n", %d);
                                                                      produits sur l'ordinateur
                                                                                                                                □ A
      if (a > 0)
                                                                   □ un fichier texte qui sera traduit en instructions
                                                                                                                          19. Le code suivant :
        return f(a - 1) + 1;
                                                                                                                               int i;
                                                                  □ un document qui doit être protégé
                                                                                                                               for (i = 4; i \ge 0; i = i - 1)
      return 4;
                                                                   □ un document de référence du système
                                                                                                                                    printf("%d ", i);
                                                             16. Un registre du processeur est :
   Alors l'expression f(1) prendra la valeur :
                                                                  \square une unité de calcul spécialisée de l'ordinateur
      \Box 4
                                                                                                                               printf("\n");
                                                                  \square un composant qui contient la liste des fichiers du
     \Box 0
                                                                                                                              affichera:
                                                                      système
     \Box 5
                                                                                                                               \Box 01234
                                                                   □ une gamme de fréquence de fonctionnement du
     \Box 1
                                                                      processeur
                                                                                                                               \Box 1 2 3 4
14. Pour l'extrait de programme suivant :
                                                                   □ une case mémoire interne au processeur qui sera
                                                                                                                               \square 4 3 2 1
     int i = 0;
                                                                     manipulée directement lors des calculs
                                                                                                                                \Box \ 4\ 3\ 2\ 1\ 0
     int j = 0;
                                                            17. Sous unix (ou linux), pour créer un répertoire TP4
                                                                                                                          20. Le code suivant :
     for (i = 0; i < 3; i = i + 1)
                                                                 dans le répertoire courant on peut utiliser la com-
                                                                                                                               int i;
                                                                mande:
         for (j = 0; j < 2; j = j + 1)
                                                                                                                               for (i = 4; i > 0; i = i - 1)
                                                                   ☐ mkdir TP4
              printf("%d ", i);
                                                                  ☐ yppasswd
                                                                                                                                   printf("%d ", i);
         }
                                                                   □ new TP4
                                                                                                                               printf("\n");
                                                                   ☐ kwrite TP4
    printf("\n");
                                                                                                                              affichera:
                                                            18. Au début de la fonction main() on place le code :
   qu'est ce qui sera affiché?
                                                                                                                               \Box \ 0\ 1\ 2\ 3\ 4
     \Box 0 0 1 1 2 2
                                                                  char b = 'A';
                                                                                                                               \Box \ 4\ 3\ 2\ 1\ 0
     \Box 0 1 0 1 0 1
                                                                  b = b + 2;
                                                                 printf("%c\n", b);
                                                                                                                                \square 0 1 2 3
     \Box 1 2 3 1 2
     \Box 0 1 2 0 1 2
                                                                                                                                \square 4 3 2 1
                                                                 Alors l'affichage sera :
```

Éléments d'informatique – contrôle continue

 \square A

Prénom: Nom: N° etu :

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 1
    \Box 0
    \Box 5
2. Sur unix (ou linux), la commande mkdir permet de :
     \square créer un fichier texte
     □ changer de répertoire courant
    □ créer un répertoire
     □ ouvrir un fichier texte
3. Sous unix (ou linux), la commande cd permet de :
     □ ouvir un bureau partagé (common desktop)
     \square jouer de la musique
     □ détruire un fichier
    \square récupérer un programme arrêté avec la commande
     □ changer de répertoire courant
4. Le code suivant :
    int age = 20;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
```

printf("Majeur\n");

```
affichera:
     □ Mineur
        Majeur
     □ rien
     □ Mineur
     □ Majeur
5. 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
     \Box une unité de calcul spécialisée de l'ordinateur
6. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
    \square 1 2 3 4
    \square 4 3 2 1
    \Box \ 4\ 3\ 2\ 1\ 0
     \Box 01234
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 :
     \Box C
     □В
     □b
```

```
8. 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
     □ une fonction déclarée mais non définie
     ☐ une directive préprocesseur #include manquante
9. Le code suivant :
     int somme = 0;
     int i;
    for (i = 1; i < 4; i = i + 1)
       somme = somme + i;
    printf("%d", somme);
    affichera:
     \square 0
     \Box 6
     \Box 1
     \Box 42
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. 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;
```

```
12. Pour déclarer une fonction exposant qui prend en ar-
                                                                   □ a<=n<=b
   gument un réel x et un entier positif n et renvoie la
                                                                                                                                   return 3;
                                                                   \square (n<=a) && (n<=b)
   valeur de x^n on écrit :
                                                                   □ (a<=n) && (n<=b)
                                                                                                                                return 4;
      \square exposant(double x, int n, int r);
                                                             15. Si racine est une fonction prenant en entrée un réel
      \square void exposant(double x^n);
                                                                 et renvoyant la racine carrée de cet réel, et que x est
                                                                                                                              Alors l'expression f(0) prendra la valeur :
     \square double exposant(double x, int n);
                                                                 une variable réelle définie et initialisée, il est incorrect
                                                                 d'écrire :
                                                                                                                                \square 3
      \square int exposant(double n, int x);
                                                                   \square x = racine(racine(x)*racine(x));
                                                                                                                                \Box 4
13. Soit le programme principal suivant :
                                                                   \square x = racine(2/3);
                                                                                                                                \Box 0
   int main()
                                                                   \square x = racine(x * x) - racine(x);
   {
                                                                                                                          19. Après exécution jusqu'à la ligne 15 du programme C:
     int a = 3;
                                                                   \square x - 1 = racine(x);
     int b = 5:
                                                                                                                            10
                                                             16. Dans la commande gcc, l'option -Wall signifie :
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b)
                                                                                                                            11
                                                                                                                                 int main() {
                                                                   □ qu'on veut changer alétoirement de fond d'écran
     return EXIT_SUCCESS;
                                                                                                                            12
                                                                                                                                      int x = 5;
                                                                   □ qu'il faut indenter le fichier source
                                                                                                                            13
                                                                                                                            14
                                                                                                                                      x = 3 * x + 1;
                                                                   □ qu'il faut lancer un déboggueur
   appelant la fonction f ainsi définie :
                                                                                                                            15
                                                                   \square que l'on veut voir tous les avertissements
   int f(int a, int b)
                                                                                                                            16
                                                            17. Pour l'extrait de programme suivant :
                                                                                                                                 }
                                                                                                                            17
      a = a + b;
                                                                  int i;
      return a;
                                                                                                                                □ la variable x vaut 16
                                                                  int j;
                                                                                                                                \square la variable x vaut -\frac{1}{2}
                                                                  for(i=4:i>0:i=i-1)
   L'affichage dans le main est le suivant :
                                                                                                                                \square le programme affiche x
      \Box f(a,b)=8, a=8, b=5
                                                                    for(j=i;j<6;j=j+1)
                                                                                                                                \square le programme affiche ****
     \Box f(a,b)=13, a=8, b=5
                                                                       printf("*");
     \Box f(3,5)=8, a=3, b=5
                                                                                                                          20. Soit la fonction g définie par :
      \Box f(a,b)=8, a=3, b=5
                                                                    printf(" ");
                                                                                                                              int g(int a)
14. On souhaite faire une boucle de contrôle de saisie : tant
   que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                                                                                printf("a = \n", %d);
   recommence la saisie de n. Soit le programme suivant :
                                                                                                                                if (1 > 0)
                                                                 qu'est ce qui sera affiché?
     int a = 0;
                                                                                                                                  return 5;
     int b = 20;
     int n;
                                                                                                                                return 7;
     scanf("%d", &n);
                                                                    ***** *** ***
     while(cond)
                                                                                                                              Alors l'expression g(0) prendra la valeur :
                                                             18. Soit la fonction f définie par :
       scanf("%d", &n);
                                                                                                                                \Box 7
                                                                 int f(int a)
                                                                                                                                \Box 0
   Quelle est la condition cond :
                                                                   printf("a = \n", %d);
                                                                                                                                \Box 5
      \square (a<n) || (n>b)
                                                                   if (a > 0)
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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

r 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 :
☐ saisie_utilisateur(scanf(%d));
\square void saisie_utilisateur(char c);
☐ int saisie_utilisateur();
\square void saisie_utilisateur(int n);
2. Le code suivant :
<pre>int i; for (i = 0; i < 5; i = i + 1) {</pre>
<pre>printf("%d ", i);</pre>
<pre>} printf("\n");</pre>
affichera:
$\Box \ 0 \ 1 \ 2 \ 3$
$\Box \ 4\ 3\ 2\ 1\ 0$
$\square \ 4\ 3\ 2\ 1$
$\Box \ 0 \ 1 \ 2 \ 3 \ 4$
3. Le langage C est un langage
\Box lu, écrit, parlé
\square compilé
\square composé
\Box interprété
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 »
$\hfill\Box$ l'analyse harmonique
$\hfill\Box$ l'analyse sémantique
$\hfill\Box$ l'analyse des entrées clavier
☐ l'édition de liens

```
5. Un fichier source est:
     ☐ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
     \square un document illisible pour les humains
     □ un document de référence du système
     □ un fichier texte qui sera traduit en instructions
        processeur
     □ un document qui doit être protégé
6. 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 x
     □ la variable x vaut 16
     \square la variable x vaut -\frac{1}{2}
     ☐ le programme affiche ****
7. Sur unix (ou linux), la commande mkdir permet de :
     □ changer de répertoire courant
     □ créer un répertoire
     \(\sigma\) ouvrir un fichier texte
     □ créer un fichier texte
8. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses cases
     \square ses chants
     \square ses champs
     \square ses blocs
9. Le type des réels en C est :
     ☐ double
     \square int
     □ char
     \square real
```

```
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 :
     \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);
11. Vous utilisez une boucle while quand :
      □ vous ne connaissez pas le nombre d'itérations de
         la boucle à l'avance
      \Box 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
12. 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)
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
      \square il comporte une boucle infinie
      \square il ne compile pas
      \square il n'affiche rien
14. Pour déclarer une variable qui sera utilisée comme va-
    riable de boucle on peut utiliser l'instruction
      \square int k;
      \square int %d;
      □ loop i;
      \square int loop n;
```

15. Pour déclarer une procédure afficher_date qui prend	\Box int struct toto_s = {3, -1e10};	\square x = racine(2/3);
en argument un struct date_s et affiche le contenu	☐ struct toto_s toto;	\square x = racine(x * x) - racine(x);
du struct, on écrit :	17. Le code suivant :	\square x = racine(racine(x)*racine(x));
☐ void afficher_date(date_s d);	int i;	19. Si n est une variable entière pour demander sa valeur
☐ void afficher_date(struct date_s d);	for $(i = 8; i > 0; i = i - 2)$	à l'utilisateur, on utilise plutôt :
☐ int afficher_date(date_s d);	{	☐ printf("Valeur de n ? %d\n", n);
\square struct date_s afficher_date(struct date_s	d); printf("%d ", i);	
16. Si le code :	} nrin+f("\n").	\square printf("Valeur de n ? %g\n", n);
10. Si ic code .	<pre>printf("\n");</pre>	\square scanf("%d", &n);
struct toto_s	affichera:	□ un débogueur
{	\Box 8 6 4 2 0	20 C:tt
int n;	\square 8 2	20. Si cet avertissement apparaît à la compilation :
<pre>double x; };</pre>	$\Box \ 0\ 2\ 4\ 6\ 8$	warning: implicit declaration of function 'max', que doit-on chercher dans le programme?
	\square 8 6 4 2	une directive préprocesseur #include manquante
précède la fonction main(), alors on peut écrire en	18. Si racine est une fonction prenant en entrée un réel	
début de main():	et renvoyant la racine carrée de cet réel, et que x est	\square une fonction appelée avant sa déclaration
□ toto_s n, x;	une variable réelle définie et initialisée, il est incorrect	$\hfill \square$ une fonction déclarée mais non définie
\Box toto_s struct z = {3, 0.5};	d'écrire :	\square un désaccord entre la déclaration et la définition
\square int toto.n = 3;	\square x - 1 = racine(x);	d'une fonction

лсепсе	

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

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

☐ #include <studio.h>

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

Alors l'affichage sera :

- □i
- \square A
- ☐ ABCDEF
- 3. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
 - \square #define taille = N
 - ☐ #define taille = 3
 - \square #define N 3
 - \square #define N = 3
- 4. Sur unix (ou linux), la commande mkdir permet de :
 - \Box créer un fichier texte
 - $\Box\,$ changer de répertoire courant
 - \square ouvrir un fichier texte
 - \square créer un répertoire

```
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 :
     \square 8
    \Box 16
    \Box 0
    \Box 4
6. 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 une fonction déclarée mais non définie
    \square un désaccord entre la déclaration et la définition
       d'une fonction
    ☐ 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
    ☐ il risque d'afficher bonjour à la place de coucou
     \Box il comporte une boucle infinie
```

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

8. Les lignes

- $\hfill \square$ 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
- □ comportent une erreur qui ne sera pas détectée
- 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 1
- \Box 0
- \Box 5
- \Box 4
- 10. 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
 - ☐ que l'on veut voir tous les avertissements
 - □ qu'il faut lancer un déboggueur
- 11. Après exécution jusqu'à la ligne 14 du programme ${\bf C}$:

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

- \square le terminal affiche x = 5
- \square le terminal affiche 5
- □ le terminal affiche "Faux"
- \square le terminal affiche x = 2

12. 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 exposant(double n, int x); ☐ double exposant(double x, int n); ☐ exposant(double x, int n, int r);	19. 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 :
<pre>int a = 0; int b = 20; int n; scanf("%d", &n); while(cond) { scanf("%d", &n);</pre>	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	<pre> n = carre(n); n = carre(int n); int carre(2); </pre>
} Quelle est la condition cond:	16. Si x est une variable réelle (de type double) alors $x = 3/2$ lui affecte la valeur : \Box 1.5	☐ int n = carre();
☐ (n<=a) && (n<=b) ☐ (a<=n) && (n<=b) ☐ a<=n<=b	□ 0 □ 1 □ 0.5	20. Une $segmentation\ fault$ est une erreur qui survient lorsque :
☐ (a <n) (n="" ="">b) 13. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :</n)>	17. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C? □ char "c";	☐ 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
□ analyse lexicale □ analyse harmonique	<pre>□ int char; □ char c; □ char 'c';</pre>	☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée
\square analyse sémantique \square analyse syntaxique	18. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :	\Box la division du programme en zones homogènes échoue
 14. Pour déclarer une fonction exposant qui prend en argument un réel x et un entier positif n et renvoie la valeur de xⁿ on écrit : □ void exposant(double x^n); 	<pre>□ int pgcd(2); □ int n = pgcd(); □ n = pgcd(int p, int q); □ n = pgcd(n, 3);</pre>	□ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal
<u>-</u>	·	

 \Box 0

Éléments d'informatique – contrôle continue

 \square 111

Prénom: Nom:	
N° etu :	

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:
     □В
     □ b
     \Box C
     \square A
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. Sous unix (ou linux), la commande 1s permet de :
     □ compiler un programme
     \square afficher la liste de fichiers contenus dans un
        répertoire
     □ voir des clips musicaux
     □ afficher le contenu d'un fichier texte
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 4
```

```
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 20
     \Box 6
     \square 3
     \square 16
6. 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 toto[5]:
     \square int tab[] = 5;
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 :
     \Box 6
     \Box 0
     \square 15
     \Box 10
8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 8
     \square 7
     \square 3
```

```
9. 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
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 :
     \square 7
     \Box 0
     \square 5
11. Un enregistrement permet de grouper plusieurs valeurs
    dans:
     □ ses cases
     \square ses champs
     \square ses chants
```

 \square ses blocs

```
12. Soit le programme principal suivant :
                                                                 affichera:
                                                                                                                                □ une fonction déclarée mais non définie
                                                                   \square 8 2
   int main()
                                                                                                                                ☐ une fonction appelée avant sa déclaration
                                                                   \Box 8 6 4 2 0
                                                                                                                          18. Dans la commande gcc, l'option -Wall signifie :
     int a = 3;
                                                                   \square 8 6 4 2
     int b = 5:
                                                                                                                                ☐ que l'on veut voir tous les avertissements
                                                                   \Box 02468
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
                                                                                                                                □ qu'il faut indenter le fichier source
     return EXIT_SUCCESS;
                                                             15. Le code suivant :
                                                                                                                                □ qu'on veut changer alétoirement de fond d'écran
                                                                  int i;
                                                                                                                                □ qu'il faut lancer un déboggueur
   appelant la fonction f ainsi définie :
                                                                  for (i = 4; i > 0; i = i - 1)
   int f(int a, int b)
                                                                                                                          19. Pour déclarer une fonction exposant qui prend en ar-
                                                                       printf("%d ", i);
   {
                                                                                                                               gument un réel x et un entier positif n et renvoie la
                                                                  }
      a = a + b;
                                                                                                                              valeur de x^n on écrit :
                                                                  printf("\n");
      return a;
                                                                                                                                \square exposant(double x, int n, int r);
                                                                 affichera:
                                                                                                                                \square void exposant(double x^n);
   L'affichage dans le main est le suivant :
                                                                   \Box 01234
                                                                                                                                \square double exposant(double x, int n);
      \Box f(a,b)=8, a=8, b=5
                                                                   \square 4 3 2 1
     \Box f(3,5)=8, a=3, b=5
                                                                                                                                \square int exposant(double n, int x);
                                                                   \square 0 1 2 3
     \Box f(a,b)=13, a=8, b=5
                                                                   \Box 4 3 2 1 0
                                                                                                                          20. Après exécution jusqu'à la ligne 15 du programme C:
     \Box f(a,b)=8, a=3, b=5
                                                             16. Pour afficher à l'aide de printf("%d\n",tab[i]);
                                                                                                                                  int main() {
                                                                                                                            10
13. L'écriture 101 en binaire correspond au nombre natu-
                                                                 le contenu d'un tableau de 5 entiers initialisé au
                                                                                                                            11
                                                                                                                                       int x = 5;
   rel:
                                                                 préalable, on utilise plutôt :
                                                                                                                            12
                                                                                                                                       int y;
      \square 101
                                                                   \Box for(i=0;i<5;i=i+1)
                                                                                                                            13
      \Box 5
                                                                                                                            14
                                                                                                                                       y = x;
                                                                   \square for(i=1;i<=5;i=i+1)
      \Box 4
                                                                                                                            15
                                                                   \square for(i=1;i<5;i=i+1)
      \square 3
                                                                                                                            16
                                                                   \square for(i=0;i<=5;i=i+1)
                                                                                                                            17
                                                                                                                                 }
14. Le code suivant :
                                                             17. Si cette erreur apparaît à la compilation :
     int i;
                                                                                                                                \Box la variable x vaut 0
                                                                 erreur: conflicting types for 'max', que doit-
     for (i = 8; i > 0; i = i - 2)
                                                                 on chercher dans le programme?
                                                                                                                                \square la variable y vaut 5
                                                                   □ une directive préprocesseur #include manquante
                                                                                                                                □ le programme affiche "Faux"
         printf("%d ", i);
                                                                   \square un désaccord entre la déclaration et la définition
                                                                                                                                \square la variable x vaut 5 et la variable y vaut 0
    printf("\n");
                                                                      d'une fonction
```

т		-1
	acence	

Éléments d'informatique – contrôle continue

Prénom : Nom : No etu :		
N° etu :		Nom:
	N° etu :	

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? □ certains programmes sont de vrais plats de spa- \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 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 programme principal ☐ l'incrément de la variable de boucle n'est pas 1 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(); □ struct int factorielle(int n); ☐ int factorielle(double n): 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 4
     \square 0
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 :
     \Box 0
     \square 7
     \Box 5
6. Le code suivant :
    int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     □ Majeur
     □ Mineur
        Majeur
     □ Mineur
     \square rien
7. 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;

```
8. Le code suivant :
     int age = 15;
     if (age < 18)
         printf("Mineur\n");
     else
         printf("Majeur\n");
   affichera:
     □ Mineur
     □ Majeur
     ☐ Mineur
        Majeur
     \square rien
9. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
     ☐ #define taille = 3
     \square #define N = 3
     □ #define N 3
     ☐ #define taille = N
10. 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);
11. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     \square changer de répertoire courant
     \square ouvrir un fichier texte
     \square créer un fichier texte
```

```
12. Au début de la fonction main() on place le code :
                                                             15. Soit la fonction f définie par :
                                                                                                                          19. Pour l'extrait de programme suivant :
                                                                 int f(int a)
     char b = 'A';
                                                                                                                                int i = 0;
                                                                 {
     b = b + 2;
                                                                                                                               int j = 0;
                                                                   printf("a = \n", %d);
     printf("%c\n", b);
                                                                                                                               for (i = 0; i < 3; i = i + 1)
                                                                   if (a > 0)
   Alors l'affichage sera :
                                                                   {
                                                                                                                                    for (j = 0; j < 2; j = j + 1)
                                                                     return f(a - 1) + 1;
      □ b
      □ A
                                                                                                                                         printf("%d ", i);
                                                                   return 4;
                                                                                                                                    }
     \Box C
                                                                 Alors l'expression f(1) prendra la valeur :
     □В
                                                                                                                               printf("\n");
                                                                   \Box 0
13. Pour l'extrait de programme suivant :
                                                                   \Box 4
                                                                                                                              qu'est ce qui sera affiché?
     int i;
                                                                   \Box 1
     int j;
                                                                                                                                \Box 0 1 0 1 0 1
                                                                   \Box 5
     for(i=4;i>0;i=i-1)
                                                             16. Si racine est une fonction prenant en entrée un réel
                                                                                                                                □ 1 2 3 1 2
                                                                 et renvoyant la racine carrée de cet réel, et que x est
       for(j=i;j<6;j=j+1)
                                                                 une variable réelle définie et initialisée, il est incorrect
                                                                                                                                \Box 0 0 1 1 2 2
                                                                 d'écrire :
         printf("*");
                                                                                                                                \Box 0 1 2 0 1 2
                                                                   \square x = racine(racine(x)*racine(x));
                                                                   \Box x = racine(x * x) - racine(x);
       printf(" ");
                                                                                                                          20. Après exécution jusqu'à la ligne 15 du programme C:
                                                                   \square x = racine(2/3);
                                                                   \square x - 1 = racine(x);
                                                                                                                                  int main() {
                                                                                                                            10
                                                             17. Un fichier source est:
                                                                                                                            11
                                                                                                                                       int x = 5;
   qu'est ce qui sera affiché?
                                                                   \square un fichier que l'ont doit citer dans les documents
                                                                                                                            12
                                                                                                                                       int y;
       ***** *** ***
                                                                      produits sur l'ordinateur
                                                                                                                            13
                                                                   □ un document qui doit être protégé
                                                                                                                            14
                                                                                                                                       y = x;
                                                                                                                            15
                                                                   □ un document de référence du système
                                                                                                                            16
                                                                                                                                       . . .
                                                                   \square un fichier texte qui sera traduit en instructions
              **** **** ****
                                                                                                                                 }
                                                                                                                            17
                                                                      processeur
14. Laquelle des analyses suivantes ne fait pas partie des
                                                                   \square un document illisible pour les humains
   étapes de la compilation :
                                                                                                                                \square la variable y vaut 5
                                                             18. L'écriture 111 en binaire correspond au nombre natu-
     \square analyse lexicale
                                                                 rel:
                                                                                                                                □ le programme affiche "Faux"
                                                                   \square 3
     \square analyse harmonique
                                                                                                                                \square la variable x vaut 5 et la variable y vaut 0
                                                                   \Box 7
     □ analyse syntaxique
                                                                   □ 111
                                                                                                                                \Box la variable x vaut 0
     □ analyse sémantique
                                                                   \square 8
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	
1. 004.	

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 reelle definie et initialisee, il est incorrect d'écrire :
\square x = racine(x * x) - racine(x);
\square x = racine(2/3);
\Box x = racine(racine(x)*racine(x));
\square x - 1 = racine(x);
2. Le code suivant :
<pre>int age = 20;</pre>
if (age < 18)
<pre>{ printf("Mineur\n");</pre>
}
<pre>printf("Majeur\n");</pre>
affichera:
☐ Mineur
Majeur
☐ Majeur
☐ Mineur
□ rien
3. 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
\Box dans le quel ces fonctions sont appelées dans le main
4. Sous unix (ou linux), la commande cd permet de :
\Box ouvir un bureau partagé (common desktop)
\Box jouer de la musique
\Box détruire un fichier
☐ récupérer un programme arrêté avec la commande ab
□ changer de répertoire courant
in changer de repersone courant

```
5. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     □ analyse sémantique
     \square analyse lexicale
    \square analyse syntaxique
     \square analyse harmonique
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);
    \Box int pgcd(int y, int x);
    \square void pgcd(int x, int y);
                                                           10
    \Box int pgcd(int x, int x);
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)
                                                           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
8. Quelle étape de la compilation vient d'échouer lors-
   qu'on a un message comme celui-ci :
                                                           12
   Undefined symbols : "_prinft" ou
   référence indéfinie vers « prinft »
     □ l'édition de liens
     □ l'analyse des entrées clavier
    ☐ l'analyse sémantique
    ☐ l'analyse harmonique
```

u	<u> </u>
9.	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 programma- tion structurée qui ne sont pas programmables en langage machine
	□ il y a des calculs programmables en langage ma- chine et qui ne sont pas programmables en pro- grammation structurée
).	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 ☐ en temps d'accès ☐ les fichiers du disque ☐ certaines données de la mémoire de travail
1.	Soit la fonction g définie par :
	<pre>int g(int a) {</pre>
	<pre>printf("a = \n", %d); if (1 > 0) { return 5; }</pre>
	return 7; }
	Alors l'expression g(0) prendra la valeur :
2.	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

☐ un point-virgule manquant

13. Quel est le problème d'un programme comportant les lignes suivantes?	☐ un désaccord entre la déclaration et la définition d'une fonction	18. Le bus système sert à : □ Écrire des données sur le dique dur			
while (1)	$\hfill\Box$ une fonction déclarée mais non définie	☐ Arriver à l'heure en cours			
<pre>{ printf("coucou\n");</pre>	\Box une directive préprocesseur $\verb"#include"$ manquante	\Box Transférer des données et intructions entre pro-			
}	\Box une fonction appelée avant sa déclaration	cesseur et mémoire			
\Box il ne compile pas	16. L'écriture <u>101</u> en binaire correspond au nombre naturel :	\Box transporter les processus du tourniquet au processeur			
\square il n'affiche rien		19. Si carre est une fonction prenant en entrée un en-			
\Box il risque d'afficher bonjour à la place de coucou	□ 101 □ 2	tier et renvoyant le carré de cet entier, et que n est			
\Box il comporte une boucle infinie		une variable entière définie et initialisée, il est correct d'écrire :			
14. Pour déclarer un tableau d'entiers de taille 5, on peut	\Box 5	\square n = carre(n);			
utiliser l'instruction	\Box 4	\square n = carre(int n);			
☐ int[] new tableau(5);	17. On considère deux variables booléennes A et B initia-	☐ int n = carre();			
☐ char tableau[5];	lisées à TRUE et FALSE respectivement. Parmi les ex- pressions booléennes suivantes, laquelle a pour valeur	☐ int carre(2);			
<pre>□ int tab[] = 5;</pre>	TRUE?	20. Si x est une variable réelle (de type double) alors			
☐ int toto[taille=5];	□ !(!A B) == (A && !B)	x = 3/2 lui affecte la valeur :			
☐ int toto[5];	☐ (A == TRUE) && (B == TRUE)	\Box 1			
15. Si cet avertissement apparaît à la compilation :					
warning: implicit declaration of function 'max	□ A && B	□ 1.5			
, que doit-on chercher dans le programme?	□ (!A B)	\square 0.5			

т		-1
	100000	- 1
	acence	

affichera:

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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); □ printf("%d", factorielle(n)); \square n = factorielle(); \square n = factorielle(p, q); 2. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur : \square 0.5 \Box 0 \Box 1 \square 1.5 3. 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 \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 □ le programme tente d'afficher des caractères sur une ligne qui va au delà de la largeur de la fenêtre du terminal 4. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n"); else printf("Majeur\n"); }

```
☐ Mineur
    □ Majeur
    □ Mineur
       Majeur
    \square rien
5. 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 0 1 2 3
    \square 4 3 2 1
6. 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
    □ en parallèle, chacun dans un registre
    □ tour à tour, un petit peu à chaque fois
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(a,b)=8, a=3, b=5
     \Box f(3,5)=8, a=3, b=5
 8. 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?
          **** **** ****
 9. 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
     \square mettre les blocs en séquence les uns à la suite des
      □ répéter un bloc tant qu'une condition est vérifée
      □ retourner un bloc
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=0;i<=5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
```

11. Un enregistrement permet de grouper plusieurs valeurs	15. L'écriture <u>111</u> en binaire correspond au nombre naturel :	qu'est ce qui sera affiché par ce printf?
dans : \Box ses chants	·	\Box j = %d
□ ses champs	□ 7 □ 111	□ j = 0
ses cases	□ 111 □ a	□ j = 4
□ ses blocs	□ 3	□ j = 5
	□ 8	
12. Vous utilisez une boucle while quand : □ vous ne connaissez pas le nombre d'itérations de	16. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	19. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu
la boucle à l'avance	☐ char "c";	du struct, on écrit :
□ vous avez déjà fait un for dans le même programme principal	☐ char 'c';	☐ void afficher_date(date_s d);
□ vous n'avez pas déclaré de fonction	☐ int char;	\square struct date_s afficher_date(struct date_s
☐ l'incrément de la variable de boucle n'est pas 1	□ char c;	☐ void afficher_date(struct date_s d);
13. Au début de la fonction main() on place le code :	17. Si n est une variable entière pour demander sa valeur	☐ int afficher_date(date_s d);
char i;	à l'utilisateur, on utilise plutôt :	20. Soit la fonction f définie par :
for (i = 'A'; i <= 'F'; i = i + 1)	□ un débogueur	-
{	□ scanf("%d", &n);	<pre>int f(int a) f</pre>
<pre>printf("%c", i); }</pre>	\square printf("Valeur de n ? %d\n", n);	printf("a = \n", %d);
<pre>printf("\n");</pre>	\square printf("Valeur de n ? %g\n", n);	if (a > 0)
Alors l'affichage sera :	18. Soit un programme contenant les lignes suivantes :	{
☐ ABCDEF	int i = 0;	return f(a - 1) + 1;
□ A	int $j = 0$;	return 4;
□ сссссс	for (i = 0; i < 3; i = i + 1)	}
□i	{ fam (i = 0, i < 5, i = i + 1)	Al 1/2
14. Dans la commande gcc, l'option -Wall signifie :	for (j = 0; j < 5; j = j + 1)	Alors l'expression f(1) prendra la valeur :
☐ qu'on veut changer alétoirement de fond d'écran		\Box 4
□ qu'il faut lancer un déboggueur	}	□ 1
☐ que l'on veut voir tous les avertissements	}	\square 0
□ qu'il faut indenter le fichier source	printf("j = %d\n", j);	□ 5
-	'	

d);

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 < 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 j = 0
     \Box j = 5
    □ j = %d
2. 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
     □ vous n'avez pas déclaré de fonction
     ☐ l'incrément de la variable de boucle n'est pas 1
3. Un bit est:
    \square un chiffre binaire (0 ou 1)
    \square un battement d'horloge processeur
     □ la longueur d'un mot mémoire
     ☐ l'instruction qui met fin à un programme
4. 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);
```

```
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 :
     □ certaines données de la mémoire de travail
     □ les fichiers du disque
     \square des processus
     \square en temps d'accès
6. Le bus système sert à :
     □ transporter les processus du tourniquet au pro-
        cesseur
     ☐ Écrire des données sur le dique dur
     ☐ Transférer des données et intructions entre pro-
        cesseur et mémoire
     ☐ Arriver à l'heure en cours
7. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     □ analyse sémantique
     \square analyse syntaxique
     □ analyse harmonique
     \square analyse lexicale
8. 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 3
     □ la variable y vaut 5
     \Boxla variable x vaut 5 et la variable y vaut 3
```

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

    a <= n <= b
</p>
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. Laquelle de ces écritures correspond à la déclaration
   d'une variable de type caractère en langage C?
      ☐ char "c";
     \square int char:
     \Box char c;
      □ char 'c';
12. Le code suivant :
     int i;
     for (i = 0; i < 5; i = i + 1)
          printf("%d ", i);
```

printf("\n");

affichera:

$\Box \ 0 \ 1 \ 2 \ 3$	affichera:	18. L'écriture $\underline{101}$ en binaire correspond au nombre natu-		
$\Box \ 4\ 3\ 2\ 1\ 0$	□ rien	rel:		
$\Box \ 0\ 1\ 2\ 3\ 4$	□ Majeur	\Box 4		
$\square \ 4\ 3\ 2\ 1$	□ Mineur	□ 101		
13. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?	□ Mineur Majeur	□ 3 □ 5		
□ un point-virgule en trop □ un point-virgule manquant □ une accolade manquante □ une accolade en trop 14. Un enregistrement permet de grouper plusieurs valeurs dans: □ ses blocs □ ses chants □ ses cases	16. 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 □ 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	<pre>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 :</pre>		
□ ses champs 15. Le code suivant :	☐ en programmation structurée on peut programmer tous les calculs programmables en langage machine	□ 3 □ 0		
<pre>int age = 15; if (age < 18) { printf("Mineur\n"); }</pre>	17. 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	\Box 4 20. 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 :		
else	d'une fonction	\square void exposant(double x^n);		
{	\square une directive préprocesseur #include manquante	\square double exposant(double x, int n);		
<pre>printf("Majeur\n"); }</pre>	$\hfill\Box$ une fonction déclarée mais non définie	\square int exposant(double n, int x);		
,	\Box une fonction appelée avant sa déclaration	\square exposant(double x, int n, int r);		

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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 document qui doit être protégé \square 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 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 □ 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 3. 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 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. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
    ☐ #include <studlib.h>
    ☐ #appart <stdlib.h>
    ☐ #include <stdio.h>
    ☐ #include <studio.h>
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(a,b)=13, a=8, b=5
    \Box f(3,5)=8, a=3, b=5
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(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);
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 :
    □ certaines données de la mémoire de travail
    \square des processus
```

□ les fichiers du disque

□ en temps d'accès

```
9. 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
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
     \square ne comportent aucune erreur
10. Le code suivant :
     int age = 20;
     if (age < 18)
         printf("Mineur\n");
     else
         printf("Majeur\n");
   affichera:
     □ rien
     □ Mineur
     □ Mineur
        Majeur
     □ Majeur
11. 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
```

12. 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 :	 □ 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 	<pre>18. Soit la fonction f définie par : int f(int a) { printf("a = \n", %d);</pre>		
□ n = carre(n);	une ligne qui va au delà de la largeur de la fenêtre du terminal	$ \begin{array}{ll} \text{if } (\mathbf{a} > 0) \\ \text{f} \end{array} $		
□ n = carre(int n);		return 3;		
<pre>□ int n = carre();</pre> □ int carre(2);	16. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	<pre>} return 4;</pre>		
13. Soient deux variables entières x et y initialisées à 4 et	☐ char "c";	}		
5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :	□ char 'c';	Alors l'expression $f(0)$ prendra la valeur : $\Box 0$		
\Box printf("x=%x et y=%y\n");	□ char c;	\Box 3		
\Box printf("x=%d et y=%d\n",x,y);	☐ int char;	□ 4		
\square printf("x=%d et y=%d\n,x,y");	17. Si le code :	19. Si x est une variable réelle (de type double) alors		
☐ printf("x=%d et y=%d\n",x y);	struct toto_s	x = 3/2 lui affecte la valeur :		
14. Un enregistrement permet de grouper plusieurs valeurs	{	\square 0.5		
dans:	int n;	\square 1.5		
□ ses blocs	double x;	□ 1		
\square ses chants	};	\Box 0		
□ ses cases	précède la fonction main(), alors on peut écrire en	20. Si cette erreur apparaît à la compilation :		
\square ses champs	début de main() :	erreur: conflicting types for 'max', que doit-		
15. Une segmentation fault est une erreur qui survient	☐ int toto.n = 3;	on chercher dans le programme?		
lorsque :	☐ struct toto_s toto;	un désaccord entre la déclaration et la définition		
☐ la division du programme en zones homogènes	☐ int struct toto_s = {3, -1e10};	d'une fonction		
échoue	□ toto_s n, x;	 □ une fonction appelée avant sa déclaration □ une directive préprocesseur #include manquante □ une fonction déclarée mais non définie 		
\Box le programme tente d'accèder à une partie de la	□ toto_s struct z = {3, 0.5};			
mémoire qui ne lui est pas réservée	□ toto_s struct z = {5, 0.5},			

Éléments d'informatique – contrôle continue

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

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);
    }
}</pre>
```

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

2. Le code suivant :

 \square 4 3 2 1

```
int i;
for (i = 4; i >= 0; i = i - 1)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 0 1 2 3 4
    □ 4 3 2 1 0
    □ 1 2 3 4
```

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

```
□ int factorielle(int x);
□ struct int factorielle(int n);
□ int factorielle(double n);
□ int factorielle();
```

```
4. Soit la fonction {\tt 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
```

□ 7

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
```

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

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

□ a == b

6. Vous utilisez une boucle while quand :

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

- \Box vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
- \Box l'incrément de la variable de boucle n'est pas 1
- $\Box\,$ vous n'avez pas déclaré de fonction
- 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 :

```
☐ for(i=0;i<=5;i=i+1)
```

$$\square$$
 for(i=0;i<5;i=i+1)

```
8. Après exécution du programme :
```

```
2 valeur 3 r1
3 mult r1 r0
4 valeur 1 r2
5 add r2 r0
6 ecriture r0 8
```

lecture 8 r0

7 stop

```
8 5
```

 \Box la case mémoire 8 contiendra 16

```
\Box\,le bus explose
```

 \Box le terminal affiche 8

```
\square la case mémoire 8 contiendra 0
```

9. L'ordonnancement par tourniquet permet :

```
☐ de ne pas perdre de temps avec la commutation de contexte
```

☐ d'afficher des ronds colorés à l'écran

```
\Box\, de doubler la mémoire disponible
```

☐ d'entretenir l'illusion que les processus tournent en parallèle

10. Après la déclaration : int mccarthy(int n);, il est correct d'écrire :

```
□ x = mccarthy(n);
□ n = mccarthy();
```

 \square n = mccarthy(p, q);

☐ int mccarthy(int 2);

11. Le code suivant :

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

 \Box 0 1 2 3 4

```
\square 4 3 2 1
```

 \square 0 1 2 3

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

<pre>12. Au début de la fonction main() on place le code :</pre>	15. Quel est le problème d'un programme comportant les lignes suivantes? while (1) { printf("coucou\n"); } □ il n'affiche rien □ il comporte une boucle infinie □ il ne compile pas □ il risque d'afficher bonjour à la place de coucou 16. 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 : □ 3 □ 6 □ 20 □ 16	 □ char tableau[5]; □ int[] new tableau(5); □ int toto[taille=5]; □ int toto[5]; 18. Sous unix (ou linux), pour créer un répertoire TP4 dans le répertoire courant on peut utiliser la commande: □ kwrite TP4 □ yppasswd □ new TP4 □ mkdir TP4 19. Le type des réels en C est: □ int □ char □ real □ double 20. Pour déclarer une fonction pgcd qui calcule et renvoie le plus grand diviseur commun de deux entiers positifs passés en arguments on écrit: □ void pgcd(int x, int y); □ int pgcd(int x, int x);
□ compilé	\Box 20	□ void pgcd(int x, int y);

т	•	-1
- 1	acence	- 1

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 3 \Box 4 \square 101 \Box 5 2. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction \square int[] new tableau(5); \square int tab[] = 5; ☐ int toto[taille=5]: □ char tableau[5]: \square int toto[5]; 3. Quels calculs peut-on programmer en programmation structurée? □ 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 □ 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 4. Vous utilisez une boucle while quand: □ 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 la boucle à l'avance ☐ l'incrément de la variable de boucle n'est pas 1

```
5. Le type des réels en C est :
    □ double
    \square real
    □ char
    □ int.
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:
    \square A
    \Box C
    □ b
    \square B
7. 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
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 :
     \Box 16
    \Box 0
    \Box 4
     \square 8
```

```
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);
      \square n = factorielle();
     ☐ int factorielle(int 2);
      ☐ printf("%d", factorielle(n));
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 0 1 1 2 2 3
     \Box 0 1 2 0 1 2 3
     \Box 0 1 2 0 1 2
     \Box 0 1 2 3 0 1 2
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. 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
```

```
13. Le code suivant :
     int i;
     for (i = 1; i < 5; i = i + 1)
         printf("%d ", i);
     }
     printf("\n");
    affichera:
      \Box 0 1 2 3 4
      \Box \ 4\ 3\ 2\ 1\ 0
      \square 1 2 3 4
     \square 4 3 2 1
14. 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);
15. 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
      \square 4 3 2 1
      \Box 01234
      \square 1 2 3 4
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=0;i<5;i=i+1)
      \square for(i=1;i<5;i=i+1)
      \square for(i=1;i<=5;i=i+1)
17. 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;
                                                               20. Soient deux variables entières x et y initialisées à 4 et
     scanf("%d", &n);
                                                                   5 respectivement. L'affichage x=4 et y=5 est obtenu
     while(cond)
        scanf("%d", &n);
    Quelle est la condition cond :
      □ a<=n<=b
```

□ (a <n) (n="" ="">b)</n)>
□ (a<=n) && (n<=b)
\square (n<=a) && (n<=b)
18. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doiton chercher dans le programme?
\Box un point-virgule man quant
$\hfill\Box$ une accolade man quante
\Box un point-virgule en trop
\Box une accolade en trop
19. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :
☐ int factorielle(int x);
☐ int factorielle();
☐ int factorielle(double n);
☐ struct int factorielle(int 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);

 \Box printf("x=%x et y=%y\n");

avec la commande :

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

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. Le code suivant :
    int age = 20;
   if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    □ Mineur
    ☐ Mineur
       Majeur
    □ rien
    □ Majeur
3. Le langage C est un langage
    □ composé
    □ lu, écrit, parlé
    □ interprété
    □ compilé
```

```
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:
     \Box (a<n) || (n>b)
     □ (a<=n) && (n<=b)
     ☐ a<=n<=b
     \square (n<=a) && (n<=b)
5. 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
     \square une fonction appelée avant sa déclaration
     \square une directive préprocesseur \#include manquante
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 ? %g\n", n);
     ☐ printf("Valeur de n ? %d\n", n);
7. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise :
     ☐ double afficher_menu();
     □ void afficher_menu();
    ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
```

☐ char afficher_menu(printf("menu"));

CCG	•
8.	Quel est le problème d'un programme comportant les lignes suivantes?
	while (1)
	<pre>{ printf("coucou\n"); }</pre>
	 □ il risque d'afficher bonjour à la place de coucou □ il n'affiche rien □ il comporte une boucle infinie □ il ne compile pas
9.	Vous utilisez une boucle while quand :
	\square vous n'avez pas déclaré de fonction
	□ vous avez déjà fait un for dans le même pro gramme principal
	\Box 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
10.	Un enregistrement permet de grouper plusieurs valeurs dans :
	\square ses champs
	\square ses chants
	\square ses blocs
	□ ses cases
11.	Lorsqu'un programme utilise printf ou scanf il fau qu'il contienne l'instruction préprocesseur :
	\square #include <stdio.h></stdio.h>
	☐ #appart <stdlib.h></stdlib.h>
	\square #include <studlib.h></studlib.h>
	\square #include <studio.h></studio.h>
12.	Une variable booléenne est un variable :
	☐ réelle positive
	\Box qui est vraie ou fausse
	\Box à la quelle une valeur vient d'être affectée
	\square jamais nulle
	□ NaN (not a number, qui n'est pas un nombre)

6 7 } □ la variable y vaut 5	demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit : U void saisie_utilisateur(int n);
☐ la variable y vaut 5	
· ·	☐ void saisie_utilisateur(int n);
□ 1	,
□ le programme affiche "Faux"	☐ void saisie_utilisateur(char c);
\square la variable x vaut 5 et la variable y vaut 3	☐ saisie_utilisateur(scanf(%d));
\square la variable x vaut 3	☐ int saisie_utilisateur();
tions utilisées par votre programme principal. L'ordre	19. Pour déclarer une variable qui sera utilisée comme variable de boucle on peut utiliser l'instruction
est l'ordre :	\square loop i;
□ alphabétique	☐ int %d;
\Box dans lequel ces fonctions sont appelées dans le	☐ int loop n;
main	☐ int k;
	20. Un fichier source est:
-	□ un document de référence du système
-	☐ un document illisible pour les humains
•	☐ un fichier texte qui sera traduit en instructions
	processeur
a terminé	□ un document qui doit être protégé
\square tour à tour, un petit peu à chaque fois	☐ un fichier que l'ont doit citer dans les documents
\Box en parallèle, chacun dans un registre	produits sur l'ordinateur
	□ la variable x vaut 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 □ dans lequel ces fonctions sont appelées dans le main □ un ordre quelconque □ dans lequel vous avez déclaré ces fonction Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés : □ tous ensemble □ chacun son tour, après que le processus précédent a terminé □ tour à tour, un petit peu à chaque fois

Éléments d'informatique – contrôle continue

Barème : 1 points par réponse juste (unique) ; -0,5 points {

- 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
 - □ alphabétique
 - \square un ordre quelconque

par réponse fausse. Durée : 20 minutes.

- \Box dans lequel ces fonctions sont appelées dans le main
- 2. 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
- \Box une fonction déclarée mais non définie
- \Box une directive préprocesseur $\verb"#include"$ man quante
- 3. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?
 - ☐ char 'c';
 - □ char "c";
 - \square int char;
 - \square char c;
- 4. 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);
 - $\hfill\Box$ double exposant(double x, int n);
 - \square void exposant(double x^n);
 - \square exposant(double x, int n, int r);
- 5. Soit la fonction **f** définie par :

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

```
<u>ie – controle continue</u>
```

```
return f(a - 1) + 1;
}
return 4;
}
```

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

- \square 0 \square 1
- \Box 5
- \Box 4
- 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
- \Box 7
- 7. 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 n = pgcd(int p, int q);
- \square int n = pgcd();
- ☐ int pgcd(2);
- 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:
```

Prénom:

 N° etu :

- \square 42
- \Box 1 \Box 6
- \Box 0
- 9. Quelle étape de la compilation vient d'échouer lorsqu'on a un message comme celui-ci :

Nom:

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

- $\hfill\Box$ l'analyse harmonique
- \square l'analyse sémantique
- \Box l'analyse des entrées clavier
- □ l'édition de liens
- 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(a,b)=8, a=8, b=5
- \Box f(3,5)=8, a=3, b=5
- 11. 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. Laquelle des analyses suivantes ne fait pas partie des
                                                                                                                              Alors l'affichage sera:
 15
  16
                                                                 étapes de la compilation :
                                                                                                                                \Box i
 17
      }
                                                                   □ analyse sémantique
                                                                                                                                □ cccccc
                                                                  \square analyse harmonique
      \square la variable y vaut 5
                                                                                                                                □ A
                                                                   \square analyse lexicale
     \square la variable x vaut 5 et la variable y vaut 0
                                                                                                                                ☐ ABCDEF
                                                                   \square analyse syntaxique
      \square la variable x vaut 0
                                                                                                                          19. Pour l'extrait de programme suivant :
                                                             16. Un fichier source est:
      □ le programme affiche "Faux"
                                                                                                                               int i = 0;
                                                                   □ un fichier texte qui sera traduit en instructions
12. Le code suivant :
                                                                                                                               int j = 0;
                                                                                                                               for (i = 0; i < 3; i = i + 1)
     int i;
                                                                   \square un document illisible pour les humains
     for (i = 8; i > 0; i = i - 2)
                                                                  ☐ un fichier que l'ont doit citer dans les documents
                                                                                                                                    for (j = 0; j < 2; j = j + 1)
                                                                      produits sur l'ordinateur
         printf("%d ", i);
                                                                                                                                        printf("%d ", i);
                                                                   □ un document de référence du système
                                                                                                                                    }
    printf("\n");
                                                                   □ un document qui doit être protégé
                                                                                                                               }
   affichera:
                                                             17. Après exécution jusqu'à la ligne 15 du programme C:
                                                                                                                               printf("\n");
     \square 8 2
                                                                    int main() {
                                                                                                                              qu'est ce qui sera affiché?
                                                               11
                                                                         int x = 5:
     \Box 8 6 4 2 0
                                                                                                                                \Box 1 2 3 1 2
                                                               12
                                                                         int y = 3;
     \Box 02468
                                                                                                                                \Box 0 0 1 1 2 2
                                                               13
     \square 8 6 4 2
                                                               14
                                                                         x = y;
                                                                                                                                \Box 0 1 0 1 0 1
13. Si n est une variable entière pour demander sa valeur
                                                               15
                                                                                                                                \Box 0 1 2 0 1 2
                                                               16
   à l'utilisateur, on utilise plutôt :
                                                                         . . .
                                                               17
                                                                    }
                                                                                                                          20. Le code suivant :
     \square printf("Valeur de n ? %g\n", n);
                                                                   \square la variable y vaut 5
                                                                                                                               int i:
      \square un débogueur
                                                                                                                               for (i = 0; i < 5; i = i + 1)
                                                                   \square la variable x vaut 5 et la variable y vaut 3
      □ scanf("%d", &n);
                                                                   □ le programme affiche "Faux"
     ☐ printf("Valeur de n ? %d\n", n);
                                                                                                                                    printf("%d ", i);
                                                                   \square la variable x vaut 3
14. Pour déclarer une procédure afficher_menu sans ar-
                                                                                                                               printf("\n");
   gument et qui ne renvoie rien on utilise:
                                                             18. Au début de la fonction main() on place le code :
     ☐ double afficher_menu();
                                                                                                                              affichera:
                                                                  char i:
     ☐ char afficher_menu(printf("menu"));
                                                                                                                                \square 4 3 2 1
                                                                  for (i = 'A'; i \le 'F'; i = i + 1)
     ☐ int afficher_menu(int char);
                                                                                                                                \Box \ 4\ 3\ 2\ 1\ 0
                                                                    printf("%c", i);
     □ void afficher_menu();
                                                                                                                                \Box 01234
     ☐ int afficher_menu();
                                                                                                                                \square 0 1 2 3
                                                                  printf("\n");
```

affichera:

 \square 8 2

 \Box 8 6 4 2

 \square 8 6 4 2 0

 $\Box 02468$

Éléments d'informatique – contrôle continue

Prénom:	Nom:
	110111 .
N° etu :	
11 004.	

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

1. Un registre du processeur est : \square une unité de calcul spécialisée de l'ordinateur \Box une gamme de fréquence de fonctionnement du processeur \square un composant qui contient la liste des fichiers du système □ une case mémoire interne au processeur qui sera manipulée directement lors des calculs 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 : \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"); 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) \Box 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) 4. Le code suivant : int i; for (i = 8; i > 0; i = i - 2)printf("%d ", i); printf("\n");

```
5. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     □ ses chants
     \square ses champs
     □ ses cases
     \square ses blocs
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, y);
     \square int pgcd(int y, int x);
     \square void pgcd(int x, int y);
     \square int pgcd(int x, int x);
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 :
     \square en temps d'accès
     □ les fichiers du disque
     \square des processus
     □ certaines données de la mémoire de travail
8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \Box 7
     □ 111
     \square 8
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:
      □ (a<n) || (n>b)
      □ a<=n<=b
      □ (n<=a) && (n<=b)
      □ (a<=n) && (n<=b)
10. Quel est l'opérateur de différence en C :
      \square \neq
      \Box !
      □ <>
      □ !=
11. 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
12. Sur unix (ou linux), la commande mkdir permet de :
      □ créer un répertoire
      □ créer un fichier texte
      □ changer de répertoire courant
      □ ouvrir 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 \mid | B) == (A \&\& !B)
      \square (!A | | B)
      \square (A == TRUE) && (B == TRUE)
```

□ A && B

14.	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));
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 :
	\Box un ordre quel conque
	\Box dans lequel vous avez déclaré ces fonction
	\square alphabétique
	\Box dans lequel ces fonctions sont appelées dans le main
16.	Pour l'extrait de programme suivant :
	<pre>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 6
      \Box 16
      \square 3
      \square 20
17. 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 5
      □ le terminal affiche "Faux"
      \square le terminal affiche x = 2
      \square le terminal affiche x = 5
18. 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
19. 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 - 1 = racine(x);
      \square x = racine(2/3);
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 16

 \square 20

Éléments d'informatique – contrôle continue

TRUE?

 \square (!A | | B)

□ A && B

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

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

Prénom:	Nom:
N° etu :	

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? ☐ 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 2. Sous unix (ou linux), la commande cd permet de : \square jouer de la musique □ détruire un fichier □ changer de répertoire courant □ ouvir un bureau partagé (common desktop) ☐ récupérer un programme arrêté avec la commande ab 3. Sur un ordinateur avec un seul processeur, habituellement 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 4. 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 6

```
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 01234
    \square 1 2 3 4
    \Box \ 4\ 3\ 2\ 1\ 0
6. 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
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 0
    \square 3
     \Box 4
8. 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
```

```
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);
     \square int exposant(double n, int x);
     \square void exposant(double x^n);
10. 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:
     \Box i
     ☐ ABCDEF
     □ cccccc
     □ A
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 :
     \Box 0
     \Box 16
     \square 8
     \Box 4
12. 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;

13. Pour l'extrait de programme suivant :	15. Un registre du processeur est :	18. Pour déclarer une procédure afficher_menu sans ar-
int i;	☐ une gamme de fréquence de fonctionnement du	gument et qui ne renvoie rien on utilise:
<pre>int j;</pre>	processeur	\square double afficher_menu();
for(i=4;i>0;i=i-1)	☐ une case mémoire interne au processeur qui sera manipulée directement lors des calculs	☐ void afficher_menu();
for(j=i;j<6;j=j+1)	un composant qui contient la liste des fichiers du	☐ int afficher_menu();
{	système	☐ char afficher_menu(printf("menu"));
<pre>printf("*");</pre>	$\hfill \square$ une unité de calcul spécialisée de l'ordinateur	☐ int afficher_menu(int char);
<pre>printf(" ");</pre>	16. Quel est le problème d'un programme comportant les lignes suivantes?	19. Un fichier source est :
}	while (1)	\square un document qui doit être protégé
qu'est ce qui sera affiché?	{ printf("coucou\n");	☐ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
<pre> ***** **** ***</pre>	}	\square un fichier texte qui sera traduit en instructions
_ ** ** ** ** **	\square il ne compile pas	processeur
_ **** **** ****	\square il comporte une boucle infinie	\Box un document illisible pour les humains
_ ** *** ****	□ il n'affiche rien	\Box un document de référence du système
14. Sous unix (ou linux), la commande 1s permet de :	□ il risque d'afficher bonjour à la place de coucou	20. Lorsqu'un programme utilise printf ou scanf il faut
□ voir des clips musicaux	17. Un enregistrement permet de grouper plusieurs valeurs	qu'il contienne l'instruction préprocesseur :
\Box afficher le contenu d'un fichier texte	$dans:$ \square ses champs	☐ #include <studlib.h></studlib.h>
\square afficher la liste de fichiers contenus dans un	ses chants	☐ #appart <stdlib.h></stdlib.h>
répertoire	ses cases	☐ #include <stdio.h></stdio.h>
\Box compiler un programme	□ ses blocs	☐ #include <studio.h></studio.h>

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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
     \square 42
     \Box 1
     \Box 6
2. Un enregistrement permet de grouper plusieurs valeurs
  dans:
     \square ses chants
     \square ses champs
     \square ses cases
     \square ses blocs
3. 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
     \square une accolade en trop
     □ un point-virgule manquant
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:
     \Box toto_s struct z = {3, 0.5};
    \Box int struct toto_s = {3, -1e10};
     \square int toto.n = 3:
5. Quel est l'opérateur de différence en C :
     \Box !
     \sqcap \iff
     □ !=
     \square \neq
6. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 0.5
     \Box 1
     \Box 0
     \square 1.5
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(int n);
     \square int n = carre();
     \square n = carre(n):
     \square int carre(2);
8. 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
9. Vous utilisez une boucle while quand:
    \squarevous 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
     \square l'incrément de la variable de boucle n'est pas 1
```

```
10. 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
     \square 1 2 3 4
     \Box 01234
11. Le code suivant :
     int age = 15;
    if (age < 18)
         printf("Mineur\n");
    }
     else
     {
         printf("Majeur\n");
    }
   affichera:
     ☐ Mineur
        Majeur
     □ rien
     □ Mineur
     □ Majeur
12. 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
      \Box 0
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. 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 4
      \Box 5
      \Box 0
```

```
15. Laquelle de ces écritures correspond à la déclaration
                                                            18. Pour déclarer une variable qui sera utilisée comme va-
                                                                riable de boucle on peut utiliser l'instruction
    d'une variable de type caractère en langage C?
      \square int char;
                                                                  \square int %d;
      □ char c;
                                                                  \square int k;
      □ char "c";
      □ char 'c';
                                                                  \square int loop n;
16. Au début de la fonction main() on place le code :
                                                                  \square loop i;
     char i;
                                                            19. Si cet avertissement apparaît à la compilation :
     for (i = 'A'; i \le 'F'; i = i + 1)
                                                                warning: implicit declaration of function 'max'
                                                                , que doit-on chercher dans le programme?
       printf("%c", i);
                                                                  \square un désaccord entre la déclaration et la définition
     printf("\n");
                                                                     d'une fonction
    Alors l'affichage sera:
                                                                  ☐ une directive préprocesseur #include manquante
      ☐ ABCDEF
      \square A
                                                                  \square une fonction déclarée mais non définie
      □ ccccc
                                                                  ☐ une fonction appelée avant sa déclaration
      \Box i
17. Soit un programme contenant les lignes suivantes :
                                                            20. Après exécution du programme :
     int i = 0;
                                                                    lecture 8 r0
     int j = 0;
                                                                    valeur 3 r1
     for (i = 0; i < 2; i = i + 1)
                                                                    mult r1 r0
                                                                    valeur 1 r2
         for (j = 0; j < 3; j = j + 1)
                                                                    add r2 r0
                                                                    ecriture r0 8
              printf("%d ", i);
                                                                    stop
         }
     }
                                                                  □ la case mémoire 8 contiendra 16
    qu'est ce qui sera affiché?
                                                                  □ le terminal affiche 8
      \Box 1 2 1 2 3
                                                                  □ la case mémoire 8 contiendra 0
      \Box 0 1 2 0 1 2
      \Box 0 1 0 1 0 1 0 1
                                                                  \square le bus explose
      \Box 0 0 0 1 1 1
```

т		-1
	acence	

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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(date_s d); ☐ int afficher_date(date_s d); □ void afficher_date(struct date_s d); 2. 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); 3. 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, y); \square int pgcd(int x, int x); \square int pgcd(int y, int x); 4. Vous utilisez une boucle while quand: \square 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 5. Le code suivant : int age = 20; if (age < 18) { printf("Mineur\n"); } else printf("Majeur\n");

```
affichera:
                                                           □ rien
                                                           ☐ Mineur
                                                           □ Majeur
                                                           ☐ Mineur
                                                              Majeur
□ struct date_s afficher_date(struct date_s |d);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 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);
                                                           \Box int carre(2);
                                                      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. 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 = 0
     \Box j = 5
     \Box j = %d
     \Box j = 4
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 10
     \square 15
     \Box 6
     \Box 0
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=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)
12. 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();
      ☐ char afficher_menu(printf("menu"));
      □ void afficher_menu();
13. 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);

```
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
     \Box 4
     \square 8
     \Box 16
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 1 2 3 0 1 2
     \Box 0 0 1 1 2 2 3
```

```
16. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse harmonique
      \square analyse lexicale
      \square analyse syntaxique
      \square analyse sémantique
17. 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
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)
                                                                    \Box il n'affiche rien
      {
                                                                    \Box\,il comporte une boucle infinie
         somme = somme + serie[i];
```

<pre>} printf("somme = %d",somme);</pre>
La valeur de somme affichée est :
\Box 6
$\square 20$
\Box 16
\Box 3
19. L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
□ 7
□ 8
\square 3
□ 111
20. Quel est le problème d'un programme comportant le lignes suivantes?
while (1)
<pre>{ printf("coucou\n"); }</pre>
\square il ne compile pas
☐ il risque d'afficher bonjour à la place de coucou

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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 en trop □ un point-virgule manquant □ un point-virgule en trop \square une accolade manguante 2. 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 n = pgcd(); \square int pgcd(2); 3. 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 1 2 3 4 $\Box 01234$ 4. 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 \square la variable x vaut 16 \square le programme affiche **** \square la variable x vaut $-\frac{1}{2}$

 \square le programme affiche x

```
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 :
     \Box 0
     \Box 7
     \Box 5
6. 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>
7. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 5
     \square 101
     \square 3
     \Box 4
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;
     \square toto_s n, x;
     □ struct toto_s toto;
     \Box int struct toto_s = {3, -1e10};
     \Box toto_s struct z = {3, 0.5};
```

```
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?
      \square !(!A \mid | B) == (A \&\& !B)
      \square (!A || B)
      \square (A == TRUE) && (B == TRUE)
      □ A && B
10. 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
11. 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
      \Box la variable x vaut 0
12. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square n = mccarthy();
      \square int mccarthy(int 2);
      \square n = mccarthy(p, q);
```

 \square x = mccarthy(n);

13. Sur unix (ou linux), la commande mkdir permet de :	□ certaines données de la mémoire de travail	affichera:
\square ouvrir un fichier texte	\Box les fichiers du disque	☐ Majeur
□ créer un répertoire	\square des processus	☐ Mineur
\square changer de répertoire courant	17. Après exécution jusqu'à la ligne 15 du programme C :	\square rien
\square créer un fichier texte	10 int main() {	☐ Mineur
14. Soit la fonction f définie par :		Majeur
int f(int a)	12 int y = 3; 13	20. On souhaite faire une boucle de contrôle de saisie : tant que l'entier \mathbf{n} n'appartient pas à l'intervalle $[ab]$, on
<pre>printf("a = \n", %d);</pre>	14 x = y; 15	recommence la saisie de ${\tt n}.$ Soit le programme suivant :
if (a > 0) {	16 17 }	int a = 0; int b = 20;
return 3; } return 4; }	 □ le programme affiche "Faux" □ la variable y vaut 5 □ la variable x vaut 5 et la variable y vaut 3 	<pre>int n; scanf("%d", &n); while(cond) {</pre>
Alors l'expression f(0) prendra la valeur :	\Box la variable x vaut 3	scanf("%d", &n);
\square 3	18. Un enregistrement permet de grouper plusieurs valeurs	}
	dans : \square ses chants	Quelle est la condition cond :
\Box 4	□ ses cases	☐ (a <n) (n="" ="">b)</n)>
15. Si cet avertissement apparaît à la compilation :	□ ses blocs	□ (a<=n) && (n<=b)
warning: implicit declaration of function 'max , que doit-on chercher dans le programme?	□ ses champs	□ (n<=a) && (n<=b) □ a<=n<=b
☐ un désaccord entre la déclaration et la définition d'une fonction	19. Le code suivant : int age = 15;	
☐ une fonction déclarée mais non définie	if (age < 18) {	
\Box une directive préprocesseur $\verb"#include"$ man quante	<pre>printf("Mineur\n");</pre>	
\Box une fonction appelée avant sa déclaration	}	
16. La virtualisation de la mémoire permet notamment de stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :	<pre>else { printf("Majeur\n"); }</pre>	
\Box en temps d'accès		

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 □ 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 2. Le code suivant : int i; for (i = 0; i < 7; i = i + 2){ printf("%d ", i); printf("\n"); affichera: $\Box 0123456$ $\Box 02468$ \Box 0 1 2 3 4 5 6 7 $\Box 0246$ 3. L'écriture 101 en binaire correspond au nombre naturel: \Box 5 \square 3 \square 101 \Box 4 4. 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 \square 4 3 2 1

 $\Box 01234$

```
5. Le langage C est un langage
     □ compilé
     □ composé
     □ interprété
     □ lu, écrit, parlé
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(2/3);
    \square x = racine(x * x) - racine(x):
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
7. 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);
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:
     ☐ ABCDEF
     □ A
     \Box i
     □ ccccc
9. 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
     □ un caractère interdit en C
     □ une variable non déclarée
     \square une faute de frappe dans un appel de fonction
```

```
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 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
11. 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 fonction appelée avant sa déclaration
     ☐ une directive préprocesseur #include manquante
     □ un désaccord entre la déclaration et la définition
        d'une fonction
12. 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 0 1 2
     \Box 0 1 2 3 0 1 2
```

 \Box 0 1 2 0 1 2 3

13. Si x est une variable réelle (de type double) alors	14
x = 3/2 lui affecte la valeur :	15
□ 1.5	16
$\square 0.5$	17 }
\Box 0	□ la
□ 1	□ la
14. Un enregistrement permet de grouper plusieurs valeurs dans :	□ le
□ ses cases	17. Soit le
\square ses blocs	int m
\square ses chants	{
\square ses champs	int
15. Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?	int prin prin retu
$\hfill\Box$ un point-virgule man quant	appela
\Box un point-virgule en trop	int f
\Box une accolade en trop	{
$\hfill\Box$ une accolade manquante	a =
16. Après exécution jusqu'à la ligne 15 du programme C :	ret [.] }
10 int main() { 11 int x = 5;	L'affic
$ \begin{array}{ll} \text{11} & \text{int } x = 5, \\ 12 & \text{int } y = 3; \end{array} $	□ f
13 2m3 y 3,	□ f

```
x = y;
a variable x vaut 3
a variable x vaut 5 et la variable y vaut 3
e programme affiche "Faux"
a variable y vaut 5
 programme principal suivant:
ain()
a = 3;
b = 5;
tf("f(a,b)=\%d, a=\%d, b=\%d\n",f(a,b),a,b);
rn EXIT_SUCCESS;
ant la fonction f ainsi définie :
(int a, int b)
 a + b;
urn a;
chage dans le main est le suivant :
(a,b)=8, a=8, b=5
E(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
18. Vous utilisez une boucle while quand :
     \Box 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
     \square vous n'avez pas déclaré de fonction
19. 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
20. Pour déclarer une procédure afficher_menu sans ar-
   gument et qui ne renvoie rien on utilise:
     ☐ int afficher_menu(int char);
     ☐ char afficher_menu(printf("menu"));
      ☐ int afficher_menu();
     ☐ double afficher_menu();
      □ void afficher_menu();
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
renom.	INOIII .	
NTO		
N° etu :		

pa

	me : 1 points par reponse 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 alphabétique
	\Box dans lequel ces fonctions sont appelées dans le main
	\Box un ordre quel conque
2.	Le type des réels en C est :
	\square double
	\square int
	□ char
	\square real
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
	\Box les fichiers du disque
	\Box certaines données de la mémoire de travail
	\Box en temps d'accès
4.	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 exposant(double x, int n, int r);
	\square double exposant(double x, int n);
	\square void exposant(double x^n);
5.	Le code suivant :
	<pre>int i; for (i = 4; i > 0; i = i - 1) {</pre>

printf("%d ", i);

}

printf("\n");

```
affichera:
     \square 4 3 2 1
     \Box \ 4\ 3\ 2\ 1\ 0
     \square 0 1 2 3
     \Box 01234
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
     □ a == b
     \square (a.n == b.n) && (a.x == b.x)
     \square a\{n, x\} == b\{n, x\}
7. Après exécution jusqu'à la ligne 15 du programme C :
      int main() {
            int x = 5;
 11
 12
           int y;
 13
 14
           y = x;
 15
 16
 17
      }
     \Boxle programme affiche "Faux"
     \square la variable x vaut 5 et la variable y vaut 0
     \Box la variable x vaut 0
     □ la variable y vaut 5
8. 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
     \square il risque d'afficher bonjour à la place de coucou
```

```
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. 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);
11. 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;

```
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 = 4
     \Box j = 0
     \Box j = %d
     \Box j = 5
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 :
     \square 3
     \Box 0
     \Box 4
```

```
17. Pour afficher à l'aide de printf("%d\n",tab[i]);
14. Le code suivant :
                                                                le contenu d'un tableau de 5 entiers initialisé au
     int i;
                                                                préalable, on utilise plutôt :
     for (i = 1; i < 5; i = i + 1)
                                                                  \square for(i=1;i<=5;i=i+1)
                                                                  \square for(i=1;i<5;i=i+1)
         printf("%d ", i);
                                                                  \square for(i=0;i<=5;i=i+1)
     printf("\n");
                                                                  \square for(i=0;i<5;i=i+1)
    affichera:
                                                            18. Si cette erreur apparaît à la compilation :
                                                                error: expected ';' before '}' token que doit-
      \square 4 3 2 1
                                                                on chercher dans le programme?
     \Box \ 4\ 3\ 2\ 1\ 0
                                                                  \square un point-virgule manquant
     \Box 1 2 3 4
                                                                  \square une accolade en trop
                                                                  □ un point-virgule en trop
      \Box 01234
                                                                  \square une accolade manquante
15. Pour déclarer une procédure afficher_date qui prend
                                                            19. Laquelle des analyses suivantes ne fait pas partie des
    en argument un struct date_s et affiche le contenu
                                                                étapes de la compilation :
    du struct, on écrit :
                                                                  □ analyse syntaxique
     ☐ struct date_s afficher_date(struct date_s d);
                                                                  □ analyse sémantique
     □ void afficher_date(date_s d);
                                                                  \square analyse lexicale
      □ void afficher_date(struct date_s d);
                                                                  \square analyse harmonique
      ☐ int afficher_date(date_s d);
                                                            20. Au début de la fonction main() on place le code :
                                                                 char b = 'A';
16. Soient deux variables entières x et y initialisées à 4 et
    5 respectivement. L'affichage x=4 et y=5 est obtenu
                                                                 b = b + 2;
                                                                 printf("%c\n", b);
    avec la commande :
                                                                Alors l'affichage sera :
     \Box printf("x=%x et y=%y\n");
                                                                  \square A
     \square printf("x=%d et y=%d\n",x y);
                                                                  □ b
     \square printf("x=%d et y=%d\n,x,y");
                                                                  □В
      \square printf("x=%d et y=%d\n",x,y);
                                                                  \Box C
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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?

 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. Avant de faire appel à une fonction il est nécessaire de :
 - □ l'avoir définie
 - □ l'avoir déclarée et définie
 - ☐ l'avoir déclarée
 - $\hfill \square$ avoir défini une constante symbolique de la taille de cette fonction
- 3. Soit le programme principal suivant :

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

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

```
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(a,b)=8, a=3, b=5
```

```
4. 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 ? %g\n", n);
- \square printf("Valeur de n ? %d\n", n);
- 5. Le langage C est un langage
 - \Box interprété
 - \square composé
 - \Box compilé
 - \Box lu, écrit, parlé
- 6. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
 - \square #define taille = 3
 - □ #define N 3
 - \square #define taille = N
 - \square #define N = 3
- 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 :
 - \Box x = racine(racine(x)*racine(x));
 - \square x 1 = racine(x);
 - \square x = racine(2/3);
 - \square x = racine(x * x) racine(x);
- 8. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
 - ☐ #include <studio.h>
 - \square #include <studlib.h>
 - \square #appart <stdlib.h>
 - ☐ #include <stdio.h>
- 9. Un bit est:
 - \Box l'instruction qui met fin à un programme
 - \square un battement d'horloge processeur
 - \square un chiffre binaire (0 ou 1)
 - \Box la longueur d'un mot mémoire

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 :

- $\square a\{n, x\} == b\{n, x\}$
- □ a == b
- \square a = b
- \square (a.n == b.n) && (a.x == b.x)
- 11. 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)
- \square (n<=a) && (n<=b)
- □ a<=n<=b
- □ (a<=n) && (n<=b)
- 12. Un fichier source est :
 - \Box un fichier que l'ont doit citer dans les documents produits sur l'ordinateur
 - $\hfill \square$ un document qui doit être protégé
 - ☐ un fichier texte qui sera traduit en instructions processeur
 - \square un document illisible pour les humains
 - \Box un document de référence du système

```
13. Après exécution jusqu'à la ligne 15 du programme C:
                                                            16. Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                                □ l'analyse harmonique
                                                                 tions utilisées par votre programme principal. L'ordre
       int main() {
                                                                                                                                ☐ l'analyse sémantique
                                                                 dans lequel vous devez maintenant définir ces fonctions
  11
            int x = 5;
                                                                                                                                \square l'analyse des entrées clavier
                                                                est l'ordre :
  12
            int y;
                                                                                                                                □ l'édition de liens
                                                                   \square un ordre quelconque
  13
  14
            y = x;
                                                                  \square dans lequel vous avez déclaré ces fonction
                                                                                                                         19. L'écriture 101 en binaire correspond au nombre natu-
  15
                                                                                                                              rel:
                                                                  □ dans lequel ces fonctions sont appelées dans le
  16
                                                                      main
                                                                                                                               \square 3
  17
       }
                                                                  □ alphabétique
                                                                                                                               \Box 5
      \square le programme affiche "Faux"
                                                             17. Si le code :
      \square la variable x vaut 0
                                                                                                                               \square 101
      \Boxla variable x vaut 5 et la variable y vaut 0
                                                                 struct toto_s
                                                                                                                                \Box 4
      \square la variable y vaut 5
                                                                                                                         20. Soit la fonction g définie par :
                                                                   int n;
14. Pour déclarer une procédure afficher_date qui prend
                                                                   double x;
                                                                                                                             int g(int a)
    en argument un struct date_s et affiche le contenu
                                                                };
                                                                                                                             {
    du struct, on écrit :
                                                                précède la fonction main(), alors on peut écrire en
                                                                                                                                printf("a = \n", %d);
      □ struct date_s afficher_date(struct date_s d);
                                                                                                                                if (1 > 0)
                                                                 début de main() :
     ☐ int afficher_date(date_s d);
                                                                                                                                {
                                                                  □ struct toto_s toto;
      □ void afficher_date(date_s d);
                                                                                                                                  return 5;
                                                                  \square int toto.n = 3;
      □ void afficher_date(struct date_s d);
                                                                   \Box toto_s struct z = {3, 0.5};
15. Soient deux variables entières x et y initialisées à 4 et
                                                                                                                                return 7;
    5 respectivement. L'affichage x=4 et y=5 est obtenu
                                                                  \Box int struct toto_s = {3, -1e10};
    avec la commande :
                                                                  \square toto_s n, x;
                                                                                                                              Alors l'expression g(0) prendra la valeur :
      \square printf("x=%x et y=%y\n");
                                                            18. Quelle étape de la compilation vient d'échouer lors-
                                                                                                                               \Box 5
      \square printf("x=%d et y=%d\n",x y);
                                                                qu'on a un message comme celui-ci :
                                                                                                                               \Box 7
      \square printf("x=%d et y=%d\n",x,y);
                                                                Undefined symbols : "_prinft" ou
      \square printf("x=%d et y=%d\n,x,y");
                                                                                                                               \Box 0
                                                                référence indéfinie vers « prinft »
```

□ A

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	
v cu.	

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 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); 2. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <studio.h> ☐ #appart <stdlib.h> ☐ #include <studlib.h> ☐ #include <stdio.h> 3. 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 4. Le type des réels en C est : □ int. ☐ double □ real □ char 5. 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 ☐ ABCDEF \Box ccccc

```
6. 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?
     **** **** ****
     ** ** ** ** **
7. 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
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 :
    □ alphabétique
    □ dans lequel vous avez déclaré ces fonction
    □ dans lequel ces fonctions sont appelées dans le
```

main

□ un ordre quelconque

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

```
11. Le code suivant :
     int age = 15;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
   affichera:
     \square rien
     ☐ Mineur
     □ Majeur
      ☐ Mineur
        Majeur
12. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
      \Box 1
     \square 1.5
     \Box 0
     \square 0.5
13. 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);
```

```
14. Le code suivant :
     int age = 20;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
    affichera:
      □ Mineur
      ☐ Mineur
        Majeur
      \square rien
      □ Majeur
15. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      □ ses cases
      \square ses champs
      \square ses chants
      \square ses blocs
16. Le langage C est un langage
      □ composé
      □ interprété
      □ compilé
      □ lu, écrit, parlé
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 :
      \square 8
      \Box 4
      \Box 0
      \Box 16
18. 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
      \square 8
19. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 3
      \Box 5
      \square 101
      \Box 4
20. 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 || B) == (A && !B)
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

Barème : 1 points par réponse juste (unique) ; -0.5 point par réponse fausse. Durée : 20 minutes.
1. 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 d
la boucle à l'avance
2. Soit la fonction f définie par :
<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
\Box 3
3. Pour afficher à l'aide de printf("%d\n",tab[i]) le contenu d'un tableau de 5 entiers initialisé a 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)
4. 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 :
□ C
□ A
□В
□ b

```
5. 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 fichier texte qui sera traduit en instructions
        processeur
     □ un document qui doit être protégé
     \square un document illisible pour les humains
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses cases
     □ ses chants
     \square ses champs
     \square ses blocs
7. Quel est l'opérateur de différence en C:
     □!=
     \Box !
     \sqcap \iff
     \square \neq
8. Le type des réels en C est :
     ☐ double
     \square real
     □ char
     □ int.
9. Avant de faire appel à une fonction il est nécessaire
   de:
     □ l'avoir déclarée
     □ avoir défini une constante symbolique de la taille
        de cette fonction
     □ l'avoir définie
```

□ l'avoir déclarée et définie

```
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
      \square il n'affiche rien
      \square il ne compile pas
11. 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 gamme de fréquence de fonctionnement du
      □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
12. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise :
      ☐ char afficher_menu(printf("menu"));
      □ void afficher_menu();
      ☐ int afficher_menu();
      ☐ int afficher_menu(int char);
      □ double afficher_menu();
13. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 3
      \square 101
      \Box 5
      \Box 4
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)=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
15. 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 3
     \Box 6
      \square 20
16. 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?
       ***** *** ***
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. 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
```

$\hfill \square$ une fonction déclarée mais non définie
\Box un désaccord entre la déclaration et la définition d'une fonction
\Box une directive préprocesseur $\#\mbox{include}$ man quante
19. 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 certaines données de la mémoire de travail
\Box les fichiers du disque
\Box en temps d'accès
20. Le code suivant :
<pre>int age = 18; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); }</pre>
affichera:
\square rien
☐ Mineur
☐ Mineur

Majeur

□ Majeur

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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 < 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
- 2. Un enregistrement permet de grouper plusieurs valeurs dans :
 - \square ses chants
 - \Box ses champs
 - \square ses blocs
 - \square ses cases
- 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 :
 - \Box dans lequel ces fonctions sont appelées dans le main
 - \Box un ordre quel conque
 - \Box alphabétique
 - □ dans lequel vous avez déclaré ces fonction

```
4. 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 4
- \square 3
- \Box 0
- 5. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
 - ☐ #include <stdio.h>
 - ☐ #include <studio.h>
 - ☐ #appart <stdlib.h>
 - \square #include <studlib.h>
- 6. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :
 - \square struct int factorielle(int n);
 - \square int factorielle(double n);
 - \Box int factorielle(int x);
 - \square int factorielle();
- 7. Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max', que doit-on chercher dans le programme?
 - \Box une fonction déclarée mais non définie
 - ☐ un désaccord entre la déclaration et la définition d'une fonction
 - \square une fonction appelée avant sa déclaration
 - $\Box\,$ une directive préprocesseur $\verb"#include"$ man quante

8. Les lignes

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

- \Box comportent une erreur qui sera détectée au cours de l'édition de lien
- \Box 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
- 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=8, b=5
- \Box f(a,b)=8, a=3, b=5
- 10. Vous utilisez une boucle while quand :
 - $\hfill \square$ 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 la boucle à l'avance

```
11. Le code suivant :
                                                             14. Le code suivant :
                                                                                                                           18. Le type des réels en C est :
                                                                  int age = 18;
                                                                                                                                □ double
     int i;
     for (i = 0; i < 5; i = i + 1)
                                                                  if (age < 18)
                                                                                                                                □ real
                                                                  {
                                                                                                                                □ char
         printf("%d ", i);
                                                                       printf("Mineur\n");
                                                                                                                                \square int
                                                                  }
     printf("\n");
                                                                  else
                                                                                                                           19. Après exécution jusqu'à la ligne 15 du programme C:
                                                                  ₹
    affichera:
                                                                                                                                  int main() {
                                                                       printf("Majeur\n");
      \Box 01234
                                                                                                                            11
                                                                                                                                       int x = 5;
                                                                  }
                                                                                                                            12
      \square 0 1 2 3
                                                                                                                                       int y = 3;
                                                                                                                            13
      \square 4 3 2 1
                                                                 affichera:
                                                                                                                            14
                                                                                                                                       x = y;
      \Box \ 4\ 3\ 2\ 1\ 0
                                                                   ☐ Mineur
                                                                                                                            15
                                                                      Majeur
                                                                                                                            16
12. Sous unix (ou linux), la commande 1s permet de :
                                                                                                                            17
                                                                                                                                 }
                                                                   □ Majeur
      □ afficher le contenu d'un fichier texte
                                                                   \square rien
      \square afficher la liste de fichiers contenus dans un
                                                                                                                                 \square la variable y vaut 5
         répertoire
                                                                   ☐ Mineur
                                                                                                                                \square la variable x vaut 5 et la variable y vaut 3
      □ voir des clips musicaux
                                                             15. Si pgcd est une fonction prenant en entrée deux entiers
                                                                                                                                \Box la variable x vaut 3
                                                                 et renvoyant un entier, il est correct d'écrire :
      □ compiler un programme
                                                                                                                                 □ le programme affiche "Faux"
                                                                   \square n = pgcd(int p, int q);
13. Soit un programme contenant les lignes suivantes :
                                                                   \square int pgcd(2);
                                                                                                                           20. On souhaite faire une boucle de contrôle de saisie : tant
     int i = 0;
                                                                                                                               que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                   \square int n = pgcd();
     int j = 0;
                                                                                                                              recommence la saisie de n. Soit le programme suivant :
                                                                   \square n = pgcd(n, 3);
     for (i = 0; i < 3; i = i + 1)
                                                                                                                                int a = 0;
                                                             16. Sur unix (ou linux), la commande mkdir permet de :
                                                                                                                                int b = 20;
         for (j = 0; j < 5; j = j + 1)
                                                                   □ créer un fichier texte
                                                                                                                                int n;
                                                                   □ ouvrir un fichier texte
                                                                                                                                scanf("%d", &n);
                 . . .
                                                                   □ créer un répertoire
                                                                                                                                while(cond)
         }
                                                                                                                                {
                                                                   \square changer de répertoire courant
                                                                                                                                  scanf("%d", &n);
     printf("j = %d\n", j);
                                                             17. Soient deux variables entières x et y initialisées à 4 et
                                                                 5 respectivement. L'affichage x=4 et y=5 est obtenu
                                                                                                                               Quelle est la condition cond :
    qu'est ce qui sera affiché par ce printf?
                                                                 avec la commande :
                                                                                                                                □ (a<=n) && (n<=b)
      \Box j = 5
                                                                   \square printf("x=%d et y=%d\n",x,y);
                                                                   \Box printf("x=%d et y=%d\n",x y);
      \Box j = %d
                                                                                                                                □ a<=n<=b
     \Box j = 4
                                                                   \square printf("x=%d et y=%d\n,x,y");
                                                                                                                                \square (n<=a) && (n<=b)
                                                                   \Box printf("x=%x et y=%y\n");
      \Box j = 0
                                                                                                                                 □ (a<n) || (n>b)
```

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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
     \square int tab[] = 5;
     ☐ int toto[taille=5];
     \square int[] new tableau(5);
     \square int toto[5];
     □ char tableau[5]:
2. 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 1 2 1 2 3
     \Box 0 0 0 1 1 1
     \Box 0 1 2 0 1 2
3. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses chants
     \square ses blocs
     \square ses cases
     \square ses champs
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 j = 5
    \Box i = 0
    □ j = %d
    \Box j = 4
5. 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);
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 :
    \square 5
    \Box 4
    \Box 1
    \Box 0
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:
      □ b
     \Box C
      \square A
     □В
 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:
     □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
      □ un ordre quelconque
      □ dans lequel vous avez déclaré ces fonction
 9. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise :
      ☐ int afficher_menu();
     ☐ int afficher_menu(int char);
     ☐ char afficher_menu(printf("menu"));
     □ void afficher_menu();
     ☐ double afficher_menu();
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, y);
     \Box int pgcd(int x, int x);
     \square void pgcd(int x, int y);
     \Box int pgcd(int y, int x);
11. 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
12. Le type des réels en C est :
      \square char
     \square int
      □ double
      □ real
```

13. Le code suivant :	15. Pour compiler un programme prog.c, on utilise la	\Box une fonction déclarée mais non définie
<pre>int age = 15; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); }</pre>	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. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : □ #appart <stdlib.h></stdlib.h>	<pre>□ une fonction appelée avant sa déclaration 19. Le code suivant : int somme = 0; int i; for (i = 1; i < 4; i = i + 1) { somme = somme + i; } printf("%d", somme);</pre>
affichera: Mineur Mineur Majeur Majeur rien	 ☐ #include <studio.h></studio.h> ☐ #include <studio.h></studio.h> ☐ #include <studlib.h></studlib.h> 17. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : ☐ analyse harmonique ☐ analyse syntaxique 	affichera: □ 0 □ 42 □ 6 □ 1
 14. Un fichier source est : □ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un document illisible pour les humains □ un fichier texte qui sera traduit en instructions processeur □ un document qui doit être protégé □ un document de référence du système 	□ analyse lexicale □ analyse sémantique 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 □ un désaccord entre la déclaration et la définition d'une fonction	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) ☐ for(i=1;i<=5;i=i+1) ☐ for(i=0;i<5;i=i+1) ☐ for(i=1;i<5;i=i+1)

Éléments d'informatique – contrôle continue

 \Box 5

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

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(a,b)=13, a=8, b=5
     \Box f(3,5)=8, a=3, b=5
2. Pour déclarer une variable qui sera utilisée comme va-
  riable de boucle on peut utiliser l'instruction
     \square loop i;
     \square int %d;
    \square int loop n;
     \square int k;
3. 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:
     ☐ ABCDEF
    \square A
    □ cccccc
     \sqcap i
```

```
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
     \square 0
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 6
     \Box 1
     \Box 0
     \square 42
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 void pgcd(int x, int y);
     \square int pgcd(int x, y);
    \Box int pgcd(int x, int x);
     \square int pgcd(int y, int x);
7. L'écriture 101 en binaire correspond au nombre natu-
  rel:
     \square 101
     \square 3
     \Box 4
```

```
8. 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
9. 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
10. 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
     \square l'incrément de la variable de boucle n'est pas 1
      □ vous n'avez pas déclaré de fonction
11. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \Box 0
```

 \square 0.5

 \square 1

<pre>12. Si le code : struct toto_s { int n; double x; }; précède la fonction main(), alors on peut écrire en début de main() : □ toto_s n, x; □ int toto.n = 3; □ struct toto_s toto; □ toto_s struct z = {3, 0.5}; □ int struct toto_s = {3, -1e10};</pre>	15. 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 16 □ le terminal affiche 8 □ le bus explose	 □ avoir défini une constante symbolique de la taille de cette fonction □ l'avoir définie 18. Un fichier source est : □ un document de référence du système □ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un document qui doit être protégé □ un fichier texte qui sera traduit en instructions processeur □ un document illisible pour les humains
13. 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 □ b □ A 14. Sur un ordinateur avec un seul processeur, habituellement les processus sont exécutés : □ chacun son tour, après que le processus précédent a terminé □ tour à tour, un petit peu à chaque fois □ tous ensemble □ en parallèle, chacun dans un registre	□ la case mémoire 8 contiendra 0 16. 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 □ 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 17. Avant de faire appel à une fonction il est nécessaire de : □ l'avoir déclarée □ l'avoir déclarée et définie	19. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses chants □ ses blocs □ ses cases □ ses champs 20. 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 □ retourner un bloc □ sélectionner entre deux blocs à l'aide d'une condition □ mettre les blocs en séquence les uns à la suite des autres

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

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

r r	eponse fausse. Duree : 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:
	$\hfill\Box$ alphabétique
	\Box dans lequel ces fonctions sont appelées dans le main
	$\hfill\Box$ un ordre quel conque
	\Box dans lequel vous avez déclaré ces fonction
2.	Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
	\square #include <studlib.h></studlib.h>
	\square #include <studio.h></studio.h>
	\square #include <stdio.h></stdio.h>
	☐ #appart <stdlib.h></stdlib.h>
3.	Un bit est:
	\Box un battement d'horloge processeur
	\Box la longueur d'un mot mémoire
	$\hfill\Box$ l'instruction qui met fin à un programme
	\square un chiffre binaire (0 ou 1)
4.	Si a et b sont deux variables de type :
	<pre>struct toto_s { int n;</pre>
	<pre>double x; };</pre>
	Alors pour tester l'égalité de a et de b on utilise la condition :
	□ a = b
	□ a == b
	$\square (a.n == b.n) && (a.x == b.x)$

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

```
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 int n = carre();
     \square n = carre(int n);
     \square int carre(2);
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 des processus
     □ les fichiers du disque
     □ en temps d'accès
7. 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 une accolade en trop
     □ un point-virgule en trop
8. Quel est l'opérateur de différence en C :
     \square \neq
     □!=
     □!
     □ <>
9. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     ☐ Mineur
     □ Majeur
     \square rien
     ☐ Mineur
        Majeur
```

```
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:
     □ b
     □В
      □ C
      \square A
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) && (n<=b)
      \square (n<=a) && (n<=b)
      \square (a<n) || (n>b)
     □ a<=n<=b
12. 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 0
      \Box 1
```

 \Box 6

```
13. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square x = mccarthy(n);
     \Box int mccarthy(int 2);
      \square n = mccarthy(p, q);
      \square n = mccarthy();
14. Afin de représenter la taille d'un tableau, définir une
    constante symbolique N valant 3.
      \square #define taille = 3
      \square #define N 3
      \square #define N = 3
      \square #define taille = N
15. Le langage C est un langage
      □ composé
     \square compilé
      □ lu, écrit, parlé
      □ interprété
16. 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);
```

```
17. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
    {
      printf("coucou\n");
      □ il n'affiche rien
     \Box\,il comporte une boucle infinie
     \square il ne compile pas
     □ il risque d'afficher bonjour à la place de coucou
18. 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
      \Box 0 1 2 0 1 2
```

```
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 int pgcd(int y, int x);
     \Box int pgcd(int x, int x);
      \Box int pgcd(int x, y);
     \square void pgcd(int x, int y);
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 7
      \Box 0
```

 \Box 5

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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 y vaut 5
     \square la variable x vaut 5 et la variable y vaut 3
     \square la variable x vaut 3
     ☐ le programme affiche "Faux"
2. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \square 8
     \Box 7
     □ 111
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 = b
     \square a{n, x} == b{n, x}
     \square a == b
     \square (a.n == b.n) && (a.x == b.x)
4. L'ordonnancement par tourniquet permet :
     \square d'afficher des ronds colorés à l'écran
     ☐ 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. Quels calculs peut-on programmer en programmation
  structurée?
    □ en programmation structurée on peut program-
       mer tous les calculs programmables en langage
       machine
    \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
    ☐ il y a des calculs programmables en langage ma-
       chine et qui ne sont pas programmables en pro-
       grammation structurée
6. Le code suivant :
   int age = 20;
   if (age < 18)
   {
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
  affichera:
    □ rien
    □ Mineur
       Majeur
    ☐ Mineur
    □ Majeur
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 int n = carre():
    \square n = carre(n);
```

 \square n = carre(int n);

```
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(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
9. Sur unix (ou linux), la commande mkdir permet de :
     □ ouvrir un fichier texte
     □ créer un répertoire
     □ changer de répertoire courant
     □ créer un fichier texte
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 :
     □ a<=n<=b
     □ (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
      □ (a<n) || (n>b)
```

 11. 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 	15. 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 harmonique ☐ l'analyse des entrées clavier ☐ l'analyse sémantique	18. Sous unix (ou linux), la commande cd permet de : □ ouvir un bureau partagé (common desktop) □ récupérer un programme arrêté avec la commande ab □ détruire un fichier □ jouer de la musique
12. Si x est une variable réelle (de type double) alors x = 3/2 lui affecte la valeur :	☐ l'édition de liens 16. Soit la fonction f définie par : int f(int a) { printf("a = \n", %d);	☐ changer de répertoire courant 19. Les lignes int i; int x=0; for(i=0,i<5,i=i+1)
□ 1.5 13. 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 -Wall prog.c -o prog.exe □ gcc prog.c -o -Wall prog.exe 14. Quel est le problème d'un programme comportant les lignes suivantes? while (1)	<pre>if (a > 0) { return 3; } return 4; } Alors l'expression f(0) prendra la valeur : □ 0 □ 3 □ 4</pre>	 x=x+1; comportent une erreur qui ne sera pas détectée comportent une erreur qui sera détectée au cours de l'édition de lien ne comportent aucune erreur comportent une erreur qui sera détectée au cours de l'analyse syntaxique
{ printf("coucou\n"); } □ il risque d'afficher bonjour à la place de coucou □ il ne compile pas □ il n'affiche rien □ il comporte une boucle infinie	17. 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(int n); ☐ void saisie_utilisateur(char c);	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=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)

Éléments d'informatique – contrôle continue

 \square 8 2

Prénom: Nom: N° etu :

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 4 add r2 r0 ecriture r0 8 stop 5 □ le terminal affiche 8 □ la case mémoire 8 contiendra 16 □ la case mémoire 8 contiendra 0 \square le bus explose 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)=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 3. 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.exe -Wall -o prog.c

☐ gcc prog.c -o -Wall prog.exe

```
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:
     □ a == b
     \square a{n, x} == b{n, x}
     \square a = b
     \Box (a.n == b.n) && (a.x == b.x)
5. L'écriture 101 en binaire correspond au nombre natu-
   rel:
     \Box 4
     \square 101
     \Box 5
     \square 3
6. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
     \square #define N = 3
     \square #define taille = 3
     \square #define taille = N
     □ #define N 3
7. 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
     \Box 8 6 4 2 0
```

```
8. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      □ ses chants
      □ ses cases
      \square ses champs
      \square ses blocs
 9. 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);
10. 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 factorielle(double n);
      \square int factorielle(int x);
11. 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
      \square 3
      \square 16
12. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \square 1.5
      \Box 0
      \square 0.5
```

 \square 1

```
13. 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
14. 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
     □ cccccc
     □ A
```

```
15. 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 comporte une boucle infinie
      \square il ne compile pas
16. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse lexicale
      \square analyse syntaxique
      □ analyse sémantique
      \square analyse harmonique
17. 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
18. 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>
```

```
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
      \square 7
      \Box 5
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(2/3);
      \square x - 1 = racine(x);
```

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

 \Box x = racine(x * x) - racine(x);

Éléments d'informatique – contrôle continue

Prénom: Nom:		
NTO .	Prénom :	Nom:
N° etu:	N° etu :	

Barème: 1 points par réponse juste (unique); -0.5 points par réponse fausse. Durée : 20 minutes. 1. Un fichier source est: □ un document de référence du système ☐ 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 fichier texte qui sera traduit en instructions processeur 2. Un programme en langage C doit comporter une et une seule définition de la fonction : \square begin □ include \square init □ main 3. Dans la commande gcc, l'option -Wall signifie : □ qu'il faut lancer un déboggueur ☐ que l'on veut voir tous les avertissements □ qu'il faut indenter le fichier source □ qu'on veut changer alétoirement de fond d'écran 4. Un enregistrement permet de grouper plusieurs valeurs dans: \square ses blocs \square ses champs \square ses cases \square ses chants 5. 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);

```
6. 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
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=0;i<=5;i=i+1)
     \square for(i=0;i<5;i=i+1)
     \square for(i=1;i<=5;i=i+1)
8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \square 8
     \square 7
     □ 111
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 0
     \Box 1
     \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 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();
      □ void afficher_menu();
     ☐ int afficher_menu(int char);
12. Sous unix (ou linux), la commande 1s permet de :
     \square afficher le contenu d'un fichier texte
      \square voir des clips musicaux
      □ compiler un programme
      \square afficher la liste de fichiers contenus dans un
        répertoire
13. Une segmentation fault est une erreur qui survient
   lorsque:
      \square 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

$\hfill \square$ le programme source a été enregistré sur le disque	16. Si cette erreur apparaît à la compilation :	\Box 6
dur en plusieurs morceaux et l'un d'entre eux ne peut pas être chargé par le compilateur	erreur: conflicting types for 'max', que doit- on chercher dans le programme?	\square 3
☐ le programme tente d'accèder à une partie de la mémoire qui ne lui est pas réservée	☐ une directive préprocesseur #include manquante ☐ une fonction déclarée mais non définie	19. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :
•	☐ une fonction appelée avant sa déclaration	☐ int pgcd(2);
14. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	un désaccord entre la déclaration et la définition	\square n = pgcd(n, 3);
, , , , , , , , , , , , , , , , , , ,	d'une fonction	\square int n = pgcd();
□ char c;	17. Si carre est une fonction prenant en entrée un en-	☐ n = pgcd(int p, int q);
☐ int char;	tier et renvoyant le carré de cet entier, et que n est une variable entière définie et initialisée, il est correct	20. Le code suivant :
□ char 'c';	d'écrire :	int age = 18;
☐ char "c";	☐ int carre(2);	if (age < 18)
15. Le code suivant :	\square n = carre(n);	{
19. De code barvani .	\square n = carre(int n);	<pre>printf("Mineur\n");</pre>
int i;	☐ int n = carre();	else
for (i = 4; i >= 0; i = i - 1) {	18. Pour l'extrait de programme suivant :	{
printf("%d ", i);	int somme = 0;	<pre>printf("Majeur\n");</pre>
}	int somme - 0; int serie[4] = {2, 4, 10, 4};	}
<pre>printf("\n");</pre>	for (i = 0; i < 4; i = i + 1)	
affichera:	{	affichera:
	somme = somme + serie[i];	☐ Mineur
$\square \ 4\ 3\ 2\ 1$	}	☐ Majeur
$\Box \ 0\ 1\ 2\ 3\ 4$	<pre>printf("somme = %d",somme);</pre>	□ rien
$\Box \ 4\ 3\ 2\ 1\ 0$	La valeur de somme affichée est :	
□ 1 2 3 4	□ 16	☐ Mineur
$\sqcup 1234$	\square 20	Majeur

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

ar reponse lausse. Duree: 20 minutes.
1. 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
☐ des processus
□ les fichiers du disque
\Box en temps d'accès
2. 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
$\hfill \square$ mettre les blocs en séquence les uns à la suite des autres
\square retourner un bloc
3. Le code suivant :
int age = 15;
if (age < 18)
{
<pre>printf("Mineur\n"); }</pre>
else
{
<pre>printf("Majeur\n"); }</pre>
affichera:
☐ Mineur
Majeur
☐ Majeur
☐ Mineur
□ rien
4. Le type des réels en C est :
□ int
□ real
□ char
□ double I

```
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 ces fonctions sont appelées dans le
        main
     □ un ordre quelconque
     □ alphabétique
     □ dans lequel vous avez déclaré ces fonction
6. Un programme en langage C doit comporter une et une
   seule définition de la fonction :
     \square include
     \square init
     □ begin
     \square main
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 0
     \square 3
     \Box 4
8. 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 et définie
     □ l'avoir définie
     □ l'avoir déclarée
```

```
9. 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
      \Box 6
      \square 3
      \square 20
10. 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
      □ un désaccord entre la déclaration et la définition
        d'une fonction
      □ une fonction déclarée mais non définie
11. Un fichier source est:
     ☐ un fichier que l'ont doit citer dans les documents
        produits sur l'ordinateur
      □ un document qui doit être protégé
      □ un fichier texte qui sera traduit en instructions
        processeur
      un document illisible pour les humains
      □ un document de référence du système
12. 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
```

13.	Si <i>n</i> est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :	
	☐ printf("Valeur de n ? %g\n", n);	
	□ un débogueur	
	☐ printf("Valeur de n ? %d\n", n);	
	☐ scanf("%d", &n);	
14.	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 déclarée mais non définie	
	\Box une fonction appelée avant sa déclaration	
	$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction	
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 B)	
	☐ (A == TRUE) && (B == TRUE)	
	☐ !(!A B) == (A && !B)	
	□ A && B	

```
16. 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();
17. Quel est l'opérateur de différence en C :
      \square \neq
      □ !=
      □ !
      □ <>
18. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse syntaxique
      \square analyse harmonique
      □ analyse sémantique
      \square analyse lexicale
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(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
20. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses chants
      \square ses blocs
      \square ses champs
```

 \square ses cases

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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

1. 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 \square tous ensemble \square en parallèle, chacun dans un registre 2. Un fichier source est: □ un fichier texte qui sera traduit en instructions processeur □ un document qui doit être protégé □ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un document illisible pour les humains \square un document de référence 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) □ a<=n<=b \square (a<=n) && (n<=b) \square (n<=a) && (n<=b) 4. L'ordonnancement par tourniquet permet : ☐ de doubler la mémoire disponible \square d'entretenir l'illusion que les processus tournent en parallèle □ d'afficher des ronds colorés à l'écran \square de ne pas perdre de temps avec la commutation

de contexte

```
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 :
    \Box 4
    \Box 5
    \Box 1
    \square 0
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
    \square 3
    \Box 4
7. 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);

```
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:
     \Box a = b
     \square a\{n, x\} == b\{n, x\}
     \square (a.n == b.n) & (a.x == b.x)
     \square a == b
 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)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
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 - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
```

 \square x = racine(2/3);

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

. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit-on chercher dans le programme?	14. Quels calculs peut-on programmer en programmation structurée?	entier et renvoyant un entier, il est correct d'écrire :	
☐ une fonction appelée avant sa déclaration	☐ il y a des calculs programmables en langage ma- chine et qui ne sont pas programmables en pro-	int factorielle(int 2);	
☐ une directive préprocesseur #include manquante	grammation structurée	□ n = factorielle(p, q);	
☐ une fonction déclarée mais non définie	\square il y a des calculs programmables en programma-	□ n = factorielle();	
☐ un désaccord entre la déclaration et la définition d'une fonction	tion structurée qui ne sont pas programmables en langage machine	18. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	
. Le code suivant :	□ certains programmes sont de vrais plats de spaghetti	□ int char;	
<pre>int age = 18; if (age < 18) {</pre>	☐ en programmation structurée on peut programmer tous les calculs programmables en langage machine	□ char c;□ char "c";□ char 'c';	
<pre>printf("Mineur\n"); }</pre>	15. Un enregistrement permet de grouper plusieurs valeurs dans :	<pre>19. Pour l'extrait de programme suivant : int i = 0;</pre>	
else {	□ ses champs	int j = 0;	
<pre>printf("Majeur\n"); }</pre>	□ ses cases	for (i = 0; i < 2; i = i + 1)	
	□ ses blocs	for $(j = 0; j < 3; j = j + 1)$	
affichera:	\square ses chants	{ printf("%d ", j);	
☐ Mineur	16. Pour l'extrait de programme suivant :	} }	
☐ Mineur Majeur	int produit = 1; int serie[4] = {2, 2, 2, 2};	qu'est ce qui sera affiché?	
□ rien	for (i = 0; i < 4; i = i + 1) {		
\square Majeur	<pre>produit = produit * serie[i];</pre>		
. Le bus système sert à :	<pre>} printf("produit = %d", produit);</pre>	\Box 0 1 2 3 0 1 2	
$\hfill \square$ Écrire des données sur le dique dur	La valeur affichée est :	20. Si x est une variable réelle (de type double) alors	
\Box Arriver à l'heure en cours		x = 3/2 lui affecte la valeur :	
\Box transporter les processus du tourniquet au processeur		\square 1.5 \square 0	
☐ Transférer des données et intructions entre pro-	□ 8	□ 1	
cesseur et mémoire	\Box 4	$\square \ \ 0.5$	

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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
     \Box 0
     \square 7
2. 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
3. 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
     □ une fonction déclarée mais non définie
     \square un désaccord entre la déclaration et la définition
        d'une fonction
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)
     ☐ for(i=1;i<=5;i=i+1)
```

```
5. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses chants
     □ ses cases
     \square ses blocs
6. 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);
7. 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
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:
     \square (a<=n) && (n<=b)
     \square (a<n) || (n>b)
     \square (n<=a) && (n<=b)
```

□ a<=n<=b</p>

```
9. 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
10. 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 programma-
        tion structurée qui ne sont pas programmables en
        langage machine
     \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
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 == TRUE) && (B == TRUE)
     □ A && B
     \square (!A || B)
```

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

```
12. 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 0 1 2 3 4 5 6
      \Box 0246
13. Sous unix (ou linux), la commande 1s permet de :
      \square afficher le contenu d'un fichier texte
      □ voir des clips musicaux
      \Box afficher la liste de fichiers contenus dans un
         répertoire
      \square compiler un programme
14. Le type des réels en C est :
      \Box int
      \square real
      ☐ double
      □ char
15. Un registre du processeur est :
      \square un composant qui contient la liste des fichiers du
         système
      \square une unité de calcul spécialisée de l'ordinateur
      \square une gamme de fréquence de fonctionnement du
         processeur
      □ une case mémoire interne au processeur qui sera
         manipulée directement lors des calculs
```

```
16. Pour compiler un programme prog.c, on utilise la
                                                                        printf("%d ", j);
   ligne de commande :
                                                                   }
                                                               }
     ☐ gcc -Wall prog.exe -o prog.c
     ☐ gcc prog.c -o -Wall prog.exe
                                                              qu'est ce qui sera affiché?
     ☐ gcc prog.exe -Wall -o prog.c
                                                                \Box 0 1 2 0 1 2 3
     ☐ gcc -Wall prog.c -o prog.exe
                                                                \Box 0 1 2 3 0 1 2
17. Le code suivant :
     int age = 20;
                                                                \Box 0 0 1 1 2 2 3
     if (age < 18)
                                                                \Box 0 1 2 0 1 2
     {
         printf("Mineur\n");
                                                          19. Si x est une variable réelle (de type double) alors
     }
                                                              x = 3/2 lui affecte la valeur :
     else
    {
                                                                \Box 1
         printf("Majeur\n");
    }
                                                                \Box 0
                                                                \square 0.5
   affichera:
                                                                \square 1.5
     □ Mineur
        Majeur
                                                          20. Vous avez déclaré préalablement un ensemble de fonc-
     □ Majeur
                                                              tions utilisées par votre programme principal. L'ordre
     □ Mineur
                                                              dans lequel vous devez maintenant définir ces fonctions
                                                              est l'ordre:
     □ rien
18. Pour l'extrait de programme suivant :
                                                                □ un ordre quelconque
     int i = 0;
                                                                \square dans lequel vous avez déclaré ces fonction
     int j = 0;
                                                                □ alphabétique
     for (i = 0; i < 2; i = i + 1)
                                                                □ dans lequel ces fonctions sont appelées dans le
         for (j = 0; j < 3; j = j + 1)
                                                                   main
```

{

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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

```
1. 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)
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 = 0
     \Box j = %d
     \Box i = 5
     \Box j = 4
3. 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
     \square un désaccord entre la déclaration et la définition
        d'une fonction
     □ une fonction déclarée mais non définie
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
     \Box 7
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. 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>
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 ne compile pas
     □ il comporte une boucle infinie
```

```
8. Le bus système sert à :
      □ Transférer des données et intructions entre pro-
         cesseur et mémoire
      □ transporter les processus du tourniquet au pro-
         cesseur
      ☐ Arriver à l'heure en cours
      ☐ Écrire des données sur le dique dur
 9. 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 3
      \Box 0 1 2 0 1 2
10. 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 n = pgcd(int p, int q);
     \square int n = pgcd();
     \square int pgcd(2);
11. 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:

```
12. Si cette erreur apparaît à la compilation :
                                                               appelant la fonction f ainsi définie :
                                                                                                                            \Box j = 0
   error: expected ';' before '}' token que doit-
                                                               int f(int a, int b)
                                                                                                                            \Box j = %d
   on chercher dans le programme?
                                                                                                                            □ j = 5
      ☐ un point-virgule en trop
                                                                 a = a + b;
      ☐ un point-virgule manquant
                                                                                                                            \Box j = 4
                                                                 return a;
      \square une accolade en trop
                                                                                                                      19. Avant de faire appel à une fonction il est nécessaire
      \square une accolade manquante
                                                               L'affichage dans le main est le suivant :
13. L'écriture 111 en binaire correspond au nombre natu-
                                                                 \Box f(a,b)=13, a=8, b=5
                                                                                                                            □ l'avoir déclarée
   rel:
                                                                \Box f(a,b)=8, a=8, b=5
     \Box 7
                                                                                                                            \square avoir défini une constante symbolique de la taille
                                                                 \Box f(a,b)=8, a=3, b=5
      □ 111
                                                                                                                               de cette fonction
                                                                 \Box f(3,5)=8, a=3, b=5
     \square 3
                                                                                                                            □ l'avoir définie
                                                           17. Lorsqu'un programme utilise printf ou scanf il faut
     \square 8
                                                                                                                            □ l'avoir déclarée et définie
                                                               qu'il contienne l'instruction préprocesseur :
14. Pour déclarer une fonction saisie_utilisateur qui
   demande à l'utilisateur d'entrer un entier au clavier et
                                                                 ☐ #include <stdio.h>
                                                                                                                      20. Pour l'extrait de programme suivant :
   renvoie cet entier on écrit :
                                                                 ☐ #include <studlib.h>
                                                                                                                           int i;
      □ void saisie_utilisateur(char c);
                                                                 ☐ #include <studio.h>
                                                                                                                           int j;
     □ void saisie_utilisateur(int n);
                                                                 ☐ #appart <stdlib.h>
                                                                                                                           for(i=4;i>0;i=i-1)
     ☐ int saisie_utilisateur();
                                                           18. Soit un programme contenant les lignes suivantes :
     ☐ saisie_utilisateur(scanf(%d));
                                                                                                                              for(j=i;j<6;j=j+1)
                                                                int i = 0;
15. Sur unix (ou linux), la commande mkdir permet de :
                                                                int j = 0;
      \square créer un fichier texte
                                                                                                                                printf("*");
                                                                for (i = 0; i < 0; i = i + 1)
     □ ouvrir un fichier texte
                                                                                                                             printf(" ");
      □ changer de répertoire courant
                                                                    for (j = 0; j < 5; j = j + 1)
      □ créer un répertoire
16. Soit le programme principal suivant :
                                                                    }
                                                                                                                          qu'est ce qui sera affiché?
   int main()
   {
                                                                                                                             ***** *** ***
                                                                printf("j = %d\n", j);
     int a = 3;
     int b = 5;
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
                                                               qu'est ce qui sera affiché?
```

};

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	1,0111
n etu:	

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
     \square il ne compile pas
     ☐ il comporte une boucle infinie
     \square il risque d'afficher bonjour à la place de coucou
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)=8, a=3, b=5
     \Box f(a,b)=13, a=8, b=5
     \Box f(a,b)=8, a=8, b=5
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}
    □ a == b
    \Box a = b
    \square (a.n == b.n) && (a.x == b.x)
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=%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");
5. Le code suivant :
    int i;
    for (i = 1; i < 5; i = i + 1)
        printf("%d ", i);
    printf("\n");
   affichera:
    \square 1 2 3 4
    \square 4 3 2 1
    \Box 01234
    \Box 4 3 2 1 0
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)
    □ (a<=n) && (n<=b)
     □ (n<=a) && (n<=b)
```

```
7. 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);
 8. Une variable booléenne est un variable :
      ☐ réelle positive
      □ qui est vraie ou fausse
      □ NaN (not a number, qui n'est pas un nombre)
      ☐ jamais nulle
      □ à laquelle une valeur vient d'être affectée
 9. 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
10. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      □ ses chants
      \square ses cases
      \square ses champs
      \square ses blocs
11. Le type des réels en C est :
      □ char
      □ int
      \square real
      □ double
```

```
18. Un programme en langage C doit comporter une et une
12. Après exécution jusqu'à la ligne 15 du programme C:
                                                               15. Avant de faire appel à une fonction il est nécessaire
                                                                   de:
                                                                                                                                  seule définition de la fonction :
       int main() {
                                                                     □ l'avoir déclarée
  11
            int x = 5;
                                                                                                                                    \square main
                                                                     □ l'avoir déclarée et définie
  12
            int y = 3;
                                                                                                                                    \square init
  13
                                                                     □ l'avoir définie
  14
            x = y;
                                                                     □ avoir défini une constante symbolique de la taille
                                                                                                                                    □ begin
  15
                                                                        de cette fonction
  16
                                                                                                                                    \square include
                                                               16. Le bus système sert à :
  17
       }
                                                                     □ Transférer des données et intructions entre pro-
                                                                                                                              19. Si pgcd est une fonction prenant en entrée deux entiers
      \square la variable x vaut 5 et la variable y vaut 3
                                                                        cesseur et mémoire
                                                                                                                                  et renvoyant un entier, il est correct d'écrire :
      \square la variable x vaut 3
                                                                     ☐ Arriver à l'heure en cours
      \square la variable y vaut 5
                                                                                                                                    \square n = pgcd(n, 3);
                                                                     ☐ Écrire des données sur le dique dur
      \square le programme affiche "Faux"
                                                                     □ transporter les processus du tourniquet au pro-
                                                                                                                                    \square int pgcd(2);
13. Sur un ordinateur avec un seul processeur, habituelle-
                                                                        cesseur
                                                                                                                                    \square int n = pgcd();
    ment les processus sont exécutés :
                                                               17. Soit un programme contenant les lignes suivantes :
                                                                                                                                    \Box n = pgcd(int p, int q);
      \square chacun son tour, après que le processus précédent
                                                                    int i = 0;
         a terminé
                                                                    int j = 0;
                                                                                                                              20. Pour l'extrait de programme suivant :
      \square tour à tour, un petit peu à chaque fois
                                                                    for (i = 0; i < 0; i = i + 1)
      \square en parallèle, chacun dans un registre
                                                                                                                                    int somme = 0;
                                                                         for (j = 0; j < 5; j = j + 1)
      \square tous ensemble
                                                                                                                                    int serie[4] = \{2, 4, 10, 4\};
                                                                                                                                    for (i = 0; i < 4; i = i + 1)
14. Après exécution jusqu'à la ligne 14 du programme C :
                                                                         }
       int main() {
                                                                                                                                       somme = somme + serie[i];
  11
            int x = 5;
  12
                                                                    printf("j = %d\n", j);
                                                                                                                                    printf("somme = %d",somme);
  13
            printf(" x = %d\n", 2);
                                                                    }
  14
                                                                                                                                  La valeur de somme affichée est :
  15
  16
       }
                                                                                                                                    \square 3
                                                                   qu'est ce qui sera affiché?
      □ le terminal affiche "Faux"
                                                                     \Box j = 4
                                                                                                                                    \square 20
      \square le terminal affiche x = 5
                                                                     \Box j = 0
                                                                                                                                    \Box 6
      \square le terminal affiche 5
                                                                     \Box j = 5
                                                                                                                                    \Box 16
      \square le terminal affiche x = 2
                                                                     \Box j = %d
```

т	•	-1
- 1	acence	- 1

Prénom:	Nom:	
N° etu:		

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 n = pgcd(int p, int q); \square int pgcd(2); \square int n = pgcd(); \square n = pgcd(n, 3); 2. Un enregistrement permet de grouper plusieurs valeurs dans: \square ses cases \square ses blocs \square ses champs \square ses chants 3. 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'; 4. 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 □ une accolade manquante \square une accolade en trop 5. 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 □ le terminal affiche 8

□ la case mémoire 8 contiendra 0

6. 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 des entrées clavier ☐ l'analyse harmonique □ l'édition de liens 7. Un programme en langage C doit comporter une et une seule définition de la fonction : \square init \square main \square include □ begin 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(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 9. 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

□ un ordre quelconque □ dans lequel ces fonctions sont appelées dans le main □ dans lequel vous avez déclaré ces fonction 10. 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 directive préprocesseur #include manquante □ un désaccord entre la déclaration et la définition d'une fonction \square une fonction déclarée mais non définie 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 \square le programme affiche **** 12. 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 0 \Box 6 \Box 1

13. Le langage C est un langage
\square composé
\square interprété
\Box lu, écrit, parlé
\square compilé
$14.\ {\rm Pour}\ {\rm d\'eclarer}$ une procédure ${\tt afficher_menu}\ {\rm sans}\ {\rm ar}$
gument et qui ne renvoie rien on utilise:
\square void afficher_menu();
☐ int afficher_menu(int char);
☐ char afficher_menu(printf("menu"));
☐ int afficher_menu();
\square double afficher_menu();
15. Pour déclarer une fonction ${\tt factorielle}$ qui prend en
argument un entier et renvoie sa factorielle on écrit :
\Box int factorielle(int x);
☐ int factorielle();
\square int factorielle(double n);
\square struct int factorielle(int n);
16. 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 :
□ C
□ A
□В
□ Ъ

```
17. 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?
       **** **** ****
18. 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
	□ 7
	□ 5
19.	Si a et b sont deux variables de type :
	<pre>struct toto_s { int n;</pre>
	<pre>double x; };</pre>
	Alors pour tester l'égalité de a et de b on utilise l condition :
	\square a{n, x} == b{n, x}
	□ (a.n == b.n) && (a.x == b.x)
	□ a = b
	□ a == b
20.	Sous unix (ou linux), la commande cd permet de :
	\Box jouer de la musique
	\Box ouvir un bureau partagé (common desktop)
	$\hfill \square$ récupérer un programme arrêté avec la command ab
	\Box changer de répertoire courant

 \square détruire un fichier

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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
     ☐ le programme affiche ****
     \square le programme affiche x
     \square la variable x vaut -\frac{1}{2}
     \square la variable x vaut 16
2. 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
     \square un ordre quelconque
     □ alphabétique
3. Le code suivant :
    int i;
    for (i = 0; i < 7; i = i + 2)
        printf("%d ", i);
   printf("\n");
  affichera:
     \Box 02468
     \Box 0123456
     \Box 0 1 2 3 4 5 6 7
     \Box 0246
```

```
4. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <stdio.h>
     ☐ #appart <stdlib.h>
     ☐ #include <studio.h>
     ☐ #include <studlib.h>
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=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)
6. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut indenter le fichier source
    □ qu'on veut changer alétoirement de fond d'écran
    □ qu'il faut lancer un déboggueur
     ☐ que l'on veut voir tous les avertissements
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;
    printf("%c\n", b);
   Alors l'affichage sera :
     □ A
     \Box C
     \square B
     □ b
```

```
9. 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
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 fonction déclarée mais non définie
      ☐ une directive préprocesseur #include manquante
      □ un désaccord entre la déclaration et la définition
         d'une fonction
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 int exposant(double n, int x);
      \square double exposant(double x, int n);
      \square exposant(double x, int n, int r);
12. 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(int p, int q);
      \square n = pgcd(n, 3);
```

```
15. Sous unix (ou linux), la commande cd permet de :
13. Si le code :
                                                                                                                             affichera:
                                                                  \hfill \squarerécupérer un programme arrêté avec la commande
    struct toto_s
                                                                                                                               \Box 6
                                                                                                                               \Box 0
      int n;
                                                                  □ ouvir un bureau partagé (common desktop)
      double x;
                                                                                                                               \Box 42
                                                                  \square détruire un fichier
    };
                                                                  □ changer de répertoire courant
                                                                                                                               \Box 1
    précède la fonction main(), alors on peut écrire en
                                                                  □ jouer de la musique
    début de main():
                                                                                                                         19. Le code suivant :
                                                            16. Un enregistrement permet de grouper plusieurs valeurs
      \square int struct toto_s = {3, -1e10};
                                                                 dans:
                                                                                                                               int i;
      \square toto_s n, x;
                                                                                                                               for (i = 1; i < 5; i = i + 1)
                                                                  \square ses blocs
      \square toto_s struct z = {3, 0.5};
                                                                  \square ses champs
      □ struct toto_s toto;
                                                                                                                                   printf("%d ", i);
                                                                  □ ses chants
      \Box int toto.n = 3;
                                                                  \square ses cases
                                                                                                                              printf("\n");
14. Pour l'extrait de programme suivant :
                                                            17. Pour déclarer une fonction saisie_utilisateur qui
                                                                                                                             affichera:
     int i;
                                                                 demande à l'utilisateur d'entrer un entier au clavier et
     int j;
                                                                renvoie cet entier on écrit:
                                                                                                                               \square 4 3 2 1
     for(i=4;i>0;i=i-1)
                                                                  □ void saisie_utilisateur(char c);
                                                                                                                               \Box \ 4\ 3\ 2\ 1\ 0
                                                                  □ void saisie_utilisateur(int n);
       for(j=i;j<6;j=j+1)
                                                                                                                               \square 1 2 3 4
                                                                  ☐ saisie_utilisateur(scanf(%d));
                                                                                                                               \Box \ 0\ 1\ 2\ 3\ 4
         printf("*");
                                                                  ☐ int saisie_utilisateur();
                                                                                                                         20. Vous utilisez une boucle while quand:
                                                             18. Le code suivant :
       printf(" ");
                                                                  int somme = 0;
                                                                                                                               □ vous avez déjà fait un for dans le même pro-
                                                                  int i;
                                                                                                                                  gramme principal
                                                                 for (i = 1; i < 4; i = i + 1)
    qu'est ce qui sera affiché?
                                                                                                                               \square vous n'avez pas déclaré de fonction
                                                                                                                               \square vous ne connaissez pas le nombre d'itérations de
                                                                    somme = somme + i;
                                                                                                                                  la boucle à l'avance
                                                                  printf("%d", somme);
                                                                                                                                ☐ l'incrément de la variable de boucle n'est pas 1
          ***** *** ***
```

	ıce.	

Prónom :	Nom ·
Prénom :	NOIII .
No otu ·	
n etu:	

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

1. Le bus système sert à:

```
☐ Transférer des données et intructions entre processeur et mémoire
```

```
\square Écrire des données sur le dique dur
```

- \Box transporter les processus du tourniquet au processeur
- ☐ Arriver à l'heure en cours
- 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");
qu'est ce qui sera affiché?

    □ 0 1 0 1 0 1
    □ 0 1 2 0 1 2
    □ 1 2 3 1 2</pre>
```

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 :

 \Box 0 0 1 1 2 2

```
4. 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.exe -Wall -o prog.c
    ☐ gcc prog.c -o -Wall prog.exe
5. Sous unix (ou linux), la commande 1s permet de :
     □ compiler un programme
     □ voir des clips musicaux
     □ afficher le contenu d'un fichier texte
     □ afficher la liste de fichiers contenus dans un
        répertoire
6. Le langage C est un langage
     □ lu, écrit, parlé
     □ compilé
     □ interprété
     □ composé
7. Après la déclaration : int mccarthy(int n);, il est
   correct d'écrire :
     \square n = mccarthy(p, q);
    \square x = mccarthy(n);
    \square n = mccarthy();
     \Box int mccarthy(int 2);
8. 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
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 ne compile pas
      \square il comporte une boucle infinie
10. 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
      ☐ une fonction appelée avant sa déclaration
      □ un désaccord entre la déclaration et la définition
         d'une fonction
11. Sous unix (ou linux), la commande cd permet de :
      □ ouvir un bureau partagé (common desktop)
      □ détruire un fichier
      ☐ récupérer un programme arrêté avec la commande
         ab
      □ jouer de la musique
      □ changer de répertoire courant
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 0
      \Box 1
      \Box 5
      \Box 4
```

13. 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): \square int n = carre(); \square int carre(2); 14. Le code suivant : int age = 20; if (age < 18) printf("Mineur\n"); printf("Majeur\n"); affichera: □ Majeur \square rien □ Mineur Majeur ☐ Mineur 15. Le code suivant : int age = 15; if (age < 18) { printf("Mineur\n"); } else {

```
printf("Majeur\n");
    }
   affichera:
     □ Mineur
     □ Mineur
        Majeur
     \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 0
     \Box 5
     \Box 7
17. 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));
```

```
18. 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 struct z = {3, 0.5};
     \Box int struct toto_s = {3, -1e10};
     \square toto_s n, x;
     \square int toto.n = 3;
19. 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();
20. 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";

Lacence	· I

Prénom:	Nom:
N° etu :	

Barème: 1 points par réponse juste (unique); -0.5 points par 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 : □ void saisie_utilisateur(char c); □ saisie_utilisateur(scanf(%d)); ☐ int saisie_utilisateur(); □ void saisie_utilisateur(int n); 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 } □ le terminal affiche 5 \square le terminal affiche x = 2 \square le terminal affiche x = 5 □ le terminal affiche "Faux" 3. 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); □ printf("Valeur de n ? %d\n", n); □ scanf("%d", &n); 4. Vous utilisez une boucle while quand: □ vous avez déjà fait un for dans le même programme principal □ vous n'avez pas déclaré de fonction \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

5. Un programme en langage C doit comporter une et une seule définition de la fonction :
□ init
□ include
□ main
□ begin
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 un désaccord entre la déclaration et la définition d'une fonction
\Box une fonction appelée avant sa déclaration
\Box une fonction déclarée mais non définie
7. Pour déclarer une fonction factorielle qui prend en argument un entier et renvoie sa factorielle on écrit :
☐ int factorielle(double n);
☐ int factorielle(int x);
☐ int factorielle();
☐ struct int factorielle(int n);
8. L'écriture <u>111</u> en binaire correspond au nombre naturel :
□ 111
\Box 3
□ 8
□ 7
9. Avant de faire appel à une fonction il est nécessaire de :
□ l'avoir déclarée
☐ l'avoir définie
\square avoir défini une constante symbolique de la taille de cette fonction
□ l'avoir déclarée et définie

```
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 comporte une boucle infinie
      □ il n'affiche rien
      \Boxil ne compile pas
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:
      □ un ordre quelconque
      □ alphabétique
      □ dans lequel ces fonctions sont appelées dans le
         main
      □ dans lequel vous avez déclaré ces fonction
12. Après exécution jusqu'à la ligne 15 du programme C :
       int main() {
 11
            int x = 5;
            int y;
 12
 13
 14
            y = x;
 15
 16
 17
      \square la variable x vaut 5 et la variable y vaut 0
      \Box la variable x vaut 0
      □ le programme affiche "Faux"
      \square la variable y vaut 5
13. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse lexicale
      □ analyse sémantique
      \square analyse harmonique
      \square analyse syntaxique
```

14. Le code suivant :	$\hfill\Box$ un document qui doit être protégé	$\hfill \square$ l'analyse des entrées clavier
int i;	\Box un fichier texte qui sera traduit en instructions	☐ l'édition de liens
for $(i = 0; i < 5; i = i + 1)$	processeur	19. Pour compiler un programme prog.c, on utilise la
{	17. Le code suivant :	ligne de commande :
<pre>printf("%d ", i); }</pre>	int age = 20;	☐ gcc -Wall prog.c -o prog.exe
<pre>printf("\n");</pre>	if (age < 18)	☐ gcc prog.c -o -Wall prog.exe
affichera:	<pre>{ printf("Mineur\n");</pre>	☐ gcc prog.exe -Wall -o prog.c
\square 0 1 2 3	printi(mineur(n), }	☐ gcc -Wall prog.exe -o prog.c
$\square 4321$	else	
$\Box \ 4\ 3\ 2\ 1\ 0$	{	20. On souhaite faire une boucle de contrôle de saisie : tant que l'entier \mathbf{n} n'appartient pas à l'intervalle $[ab]$, on
\Box 0 1 2 3 4	<pre>printf("Majeur\n"); </pre>	recommence la saisie de n. Soit le programme suivant :
15. Si cette erreur apparaît à la compilation :	J	int a = 0;
Undefined symbols :"_prinft" ou	affichera :	int b = 20;
référence indéfinie vers « prinft » que doit-	□ rien	int n;
on chercher dans le programme?	□ Majeur	scanf("%d", &n);
\square une faute de frappe dans un appel de fonction	□ Mineur	<pre>while(cond) f</pre>
\square une directive préprocesseur #include manquante	□ Mineur	scanf("%d", &n);
\Box un caractère interdit en C	Majeur	}
\square une variable non déclarée	18. Quelle étape de la compilation vient d'échouer lors-	Quelle est la condition cond :
16. Un fichier source est:	qu'on a un message comme celui-ci :	☐ (a <n) (n="" ="">b)</n)>
\Box un document illisible pour les humains	Undefined symbols :"_prinft" ou	☐ (a<=n) && (n<=b)
$\hfill\Box$ un fichier que l'ont doit citer dans les documents	référence indéfinie vers « prinft »	
produits sur l'ordinateur	\square l'analyse harmonique	□ a<=n<=b
\square un document de référence du système	\square l'analyse sémantique	□ (n<=a) && (n<=b)

Prénom:	Nom:
N° etu :	
11 004.	

Licence 1 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 : \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) 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 indenter le fichier source
- □ qu'il faut lancer un déboggueur
- 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
```

- \square le terminal affiche x = 5
- \Box le terminal affiche x = 2
- □ le terminal affiche "Faux"
- □ le terminal affiche 5
- 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);
□ void afficher_date(date_s d);
☐ int afficher_date(date_s d);
```

□ struct date_s afficher_date(struct date_s d);

```
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 processeur
- \square une unité de calcul spécialisée de l'ordinateur
- \square un composant qui contient la liste des fichiers du système
- 6. 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 0 1 1 2 2 3
- \Box 0 1 2 0 1 2 3
- \Box 0 1 2 0 1 2
- \Box 0 1 2 3 0 1 2
- 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:
- \square int toto.n = 3:
- \square int struct toto_s = {3, -1e10};
- \square toto_s n, x;
- \Box toto_s struct z = {3, 0.5};

8. Vous utilisez une boucle while quand:

 \Box l'incrément de la variable de boucle n'est pas 1□ vous avez déjà fait un for dans le même programme principal

 $\square\,$ vous n'avez pas déclaré de fonction

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

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(x * x) - racine(x): \square x = racine(2/3):

 \square x - 1 = racine(x);

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

10. 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 des entrées clavier
- □ l'édition de liens
- □ l'analyse harmonique
- □ l'analyse sémantique

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

- \square 8
- \square 3
- \square 7
- \square 111

12. 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
- \square il ne compile pas
- \square il n'affiche rien

```
13. Pour compiler un programme prog.c, on utilise la
                                                                                                                          affichera:
                                                                 printf("somme = %d",somme);
   ligne de commande :
                                                                                                                            \Box \ 4\ 3\ 2\ 1\ 0
                                                               La valeur de somme affichée est :
      ☐ gcc -Wall prog.exe -o prog.c
                                                                                                                            \square 1 2 3 4
     ☐ gcc prog.exe -Wall -o prog.c
                                                                 \Box 10
     ☐ gcc -Wall prog.c -o prog.exe
                                                                 \Box 0
                                                                                                                            \square 4 3 2 1
                                                                 \square 15
     ☐ gcc prog.c -o -Wall prog.exe
                                                                                                                            \Box 01234
                                                                 \Box 6
14. Soit la fonction f définie par :
                                                           16. Une variable booléenne est un variable :
                                                                                                                      19. Sous unix (ou linux), pour créer un répertoire TP4
   int f(int a)
                                                                 □ réelle positive
                                                                                                                          dans le répertoire courant on peut utiliser la com-
   {
                                                                 □ qui est vraie ou fausse
                                                                                                                          mande:
      printf("a = \n", %d);
      if (a > 0)
                                                                 □ jamais nulle
                                                                                                                            ☐ yppasswd
                                                                 □ à laquelle une valeur vient d'être affectée
        return 3;
                                                                 □ NaN (not a number, qui n'est pas un nombre)
                                                                                                                            ☐ mkdir TP4
                                                           17. Laquelle de ces écritures correspond à la déclaration
                                                                                                                            □ new TP4
      return 4;
                                                               d'une variable de type caractère en langage C?
                                                                                                                            ☐ kwrite TP4
                                                                 \Box char c;
   Alors l'expression f(0) prendra la valeur :
                                                                 ☐ char "c";
      \Box 0
                                                                                                                       20. Pour déclarer une fonction pgcd qui calcule et renvoie
                                                                 □ char 'c';
                                                                                                                          le plus grand diviseur commun de deux entiers positifs
     \Box 4
                                                                 \square int char;
                                                                                                                          passés en arguments on écrit :
      \square 3
                                                           18. Le code suivant :
                                                                                                                            \square void pgcd(int x, int y);
15. Pour l'extrait de programme suivant :
                                                                int i;
                                                                for (i = 1; i < 5; i = i + 1)
      int somme = 0;
                                                                                                                            \square int pgcd(int x, int x);
      for (i = 0; i < 5; i = i + 1)
                                                                                                                            \Box int pgcd(int x, y);
                                                                     printf("%d ", i);
        somme = somme + i;
                                                                                                                            \square int pgcd(int y, int x);
        i = i + 1; /* attention ! */
                                                                printf("\n");
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes. 1. 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 2. Après la déclaration : int mccarthy(int n);, il est correct d'écrire : \square int mccarthy(int 2); \square n = mccarthy(p, q); \square n = mccarthy(); \square x = mccarthy(n); 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. 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 □ une variable non déclarée \square une faute de frappe dans un appel de fonction 5. Sous unix (ou linux), la commande cd permet de : ☐ récupérer un programme arrêté avec la commande ab □ détruire un fichier □ changer de répertoire courant □ jouer de la musique □ ouvir un bureau partagé (common desktop)

```
6. Sous unix (ou linux), la commande 1s permet de :
     □ voir des clips musicaux
    □ compiler un programme
    □ afficher le contenu d'un fichier texte
    □ afficher la liste de fichiers contenus dans un
        répertoire
7. 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
    \square un document illisible pour les humains
    □ un document qui doit être protégé
     □ un document de référence du système
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. 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);
```

```
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:
      \Box (a.n == b.n) && (a.x == b.x)
      \square a{n, x} == b{n, x}
      □ a == b
      \Box a = b
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. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      ☐ int toto[taille=5];
      \square int toto[5];
      □ char tableau[5];
      ☐ int[] new tableau(5);
      \square int tab[] = 5;
13. Quel est le problème d'un programme comportant les
    lignes suivantes?
    while (1)
      printf("coucou\n");
      \square 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
```

14. L'écriture <u>101</u> en binaire correspond au nombre naturel :	 17. 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); 	☐ Transférer des données et intructions entre pro cesseur et mémoire ☐ Arriver à l'heure en cours
\Box 4	\square int saisie_utilisateur();	20. Pour l'extrait de programme suivant :
\square 3	\square saisie_utilisateur(scanf(%d));	int somme = 0;
15. Laquelle de ces écritures correspond à la déclaration	☐ void saisie_utilisateur(int n);	int serie[4] = {2, 4, 10, 4}; for (i = 0; i < 4; i = i + 1)
d'une variable de type caractère en langage C?	18. Un enregistrement permet de grouper plusieurs valeurs	{
□ char c;	dans:	<pre>somme = somme + serie[i];</pre>
□ char "c";	☐ ses champs	}
☐ int char;	☐ ses blocs	<pre>printf("somme = %d",somme);</pre>
☐ char 'c';	□ ses chants	La valeur de somme affichée est :
16. Sur unix (ou linux), la commande mkdir permet de :	☐ ses cases	\Box 6
□ changer de répertoire courant	19. Le bus système sert à :	\square 3
□ créer un répertoire	☐ transporter les processus du tourniquet au pro-	□ 9
□ créer un fichier texte	cesseur	\Box 16
\square ouvrir un fichier texte	☐ Écrire des données sur le dique dur	\square 20

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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 définie
 avoir défini une constante symbolique de la taille
 - ☐ l'avoir déclarée et définie
- 2. Soit la fonction f définie par :

de cette fonction

☐ l'avoir déclarée

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

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

3. Le code suivant :

 $\Box 02468$

```
int i;
for (i = 8; i > 0; i = i - 2)
{
    printf("%d ", i);
}
printf("\n");
affichera:
    □ 8 6 4 2 0
    □ 8 2
    □ 8 6 4 2
```

```
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\}
     \square (a.n == b.n) \&\& (a.x == b.x)
     □ a == b
5. Dans la commande gcc, l'option -Wall signifie :
     ☐ que l'on veut voir tous les avertissements
     □ qu'il faut indenter le fichier source
     □ qu'on veut changer alétoirement de fond d'écran
     □ qu'il faut lancer un déboggueur
6. 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 ne pas perdre de temps avec la commutation
        de contexte
     ☐ de doubler la mémoire disponible
7. 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
8. 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 2 0 1 2
     \Box 0 1 0 1 0 1 0 1
      \Box 0 0 0 1 1 1
      □ 1 2 1 2 3
 9. 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 void pgcd(int x, int y);
      \Box int pgcd(int x, y);
      \square int pgcd(int x, int x);
10. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses blocs
      □ ses cases
     \square ses champs
      □ ses chants
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 (n<=a) && (n<=b)
      □ (a<=n) && (n<=b)
      □ a<=n<=b
```

□ (a<n) || (n>b)

12. Lorsqu'un programme utilise printf ou scanf il faut	16. Après exécution jusqu'à la ligne 15 du programme C :	\Box le programme affiche x
qu'il contienne l'instruction préprocesseur :	10 int main() {	☐ la variable x vaut 16
☐ #include <studlib.h></studlib.h>	11 int x = 5;	□ la variable x vaut 10
☐ #include <stdio.h></stdio.h>	12 int y;	☐ le programme affiche ****
☐ #appart <stdlib.h></stdlib.h>	13	
☐ #include <studio.h></studio.h>	14 y = x; 15	\square la variable x vaut $-\frac{1}{2}$
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 :	16 17 }	19. Pour déclarer une variable qui sera utilisée comme va-
\square printf("x=%d et y=%d\n,x,y");	\Box la variable y vaut 5	riable de boucle on peut utiliser l'instruction
\square printf("x=%d et y=%d\n",x y);	□ le programme affiche "Faux"	
\Box printf("x=%x et y=%y\n");	\Box la variable x vaut 0	\square loop i;
\Box printf("x=%d et y=%d\n",x,y);	\Box la variable x vaut 5 et la variable y vaut 0	□ int loop n
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 == TRUE) && (B == TRUE) □ (!A B) □ !(!A B) == (A && !B) 15. 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 programme principal □ l'incrément de la variable de boucle n'est pas 1	17. 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(int x); □ int factorielle(double n); 18. 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 }	 int loop n; int k; int %d; 20. Un programme en langage C doit comporter une et une seule définition de la fonction : init include main begin

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 :
     □ struct int factorielle(int n);
     ☐ int factorielle();
    \square int factorielle(int x):
    ☐ int factorielle(double n);
2. 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 = %d
     \Box j = 5
3. 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);
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 en trop
     \square une accolade manquante
     □ un point-virgule manquant
```

```
5. 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);
6. Un fichier source est:
    □ un document illisible pour les humains
    □ 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
    □ un fichier texte qui sera traduit en instructions
       processeur
7. 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);
8. 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 <studio.h>
9. L'ordonnancement par tourniquet permet :
    ☐ de ne pas perdre de temps avec la commutation
        de contexte
    □ d'afficher des ronds colorés à l'écran
    \square d'entretenir l'illusion que les processus tournent
       en parallèle
     \square de doubler la mémoire disponible
```

```
10. 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);
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. Pour déclarer une procédure afficher_menu sans ar-
    gument et qui ne renvoie rien on utilise:
     ☐ int afficher_menu();
     ☐ double afficher_menu();
     ☐ int afficher_menu(int char);
     ☐ char afficher_menu(printf("menu"));
     □ void afficher_menu();
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 7
     \Box 5
     \Box 0
14. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     \square 8
     □ 111
     \square 7
```

```
15. 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 ne compile pas
      \square il comporte une boucle infinie
16. 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 1
      \Box 0
      \Box 6
```

```
17. Laquelle des analyses suivantes ne fait pas partie des
    étapes de la compilation :
      \square analyse harmonique
     \square analyse syntaxique
      \square analyse lexicale
      □ analyse sémantique
18. Le code suivant :
     int age = 20;
     if (age < 18)
     {
          printf("Mineur\n");
     printf("Majeur\n");
    affichera:
      \square Mineur
         Majeur
      □ Majeur
      ☐ Mineur
      \square rien
19. 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
     □ j = %d
20. Un enregistrement permet de grouper plusieurs valeurs
    dans:
      \square ses champs
      \square ses cases
      \square ses blocs
```

 \square ses chants

Éléments d'informatique – contrôle continue

 \square 16

Prénom:	Nom:
N° etu :	

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 = racine(x * x) - racine(x);
     \square x = racine(2/3):
     \square x - 1 = racine(x);
     \square x = racine(racine(x)*racine(x));
2. 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
     \square comportent une erreur qui sera détectée au cours
        de l'édition de lien
3. Le code suivant :
    int age = 15;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
        printf("Majeur\n");
    }
  affichera:
     \square rien
     ☐ Mineur
```

Majeur

□ Mineur

□ Majeur

```
4. 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
    ☐ une accolade manquante
    \square une accolade en trop
5. 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
    \square une fonction appelée avant sa déclaration
    \square un désaccord entre la déclaration et la définition
        d'une fonction
    □ une directive préprocesseur #include manquante
6. 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();
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
    \Box 0
    \square 8
```

```
8. Une variable booléenne est un variable :
      \square iamais nulle
      □ à laquelle une valeur vient d'être affectée
      \square qui est vraie ou fausse
      ☐ réelle positive
      □ NaN (not a number, qui n'est pas un nombre)
 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
      \Box \ 4\ 3\ 2\ 1\ 0
      \Box 1 2 3 4
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
      □ la variable x vaut 16
      \square le programme affiche x
      \square la variable x vaut -\frac{1}{2}
      \square le programme affiche ****
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=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)

12. 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 :	 □ changer de répertoire courant □ ouvir un bureau partagé (common desktop) □ récupérer un programme arrêté avec la commande 	<pre>printf("%c", i); } printf("\n");</pre>
<pre>□ n = carre(int n); □ n = carre(n); □ int carre(2); □ int n = carre();</pre>	ab 16. 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 :	Alors l'affichage sera : ABCDEF i A
 13. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt : □ printf("Valeur de n ? %g\n", n); □ un débogueur □ printf("Valeur de n ? %d\n", n); □ scanf("%d", &n); 14. Un enregistrement permet de grouper plusieurs valeurs dans : □ ses champs □ ses champs □ ses cases □ ses blocs 15. Sous unix (ou linux), la commande cd permet de : □ détruire un fichier 	□ alphabétique □ dans lequel ces fonctions sont appelées dans le main □ dans lequel vous avez déclaré ces fonction □ un ordre quelconque 17. Le bus système sert à : □ Arriver à l'heure en cours □ transporter les processus du tourniquet au processeur □ Transférer des données et intructions entre processeur et mémoire □ Écrire des données sur le dique dur 18. Au début de la fonction main() on place le code : char i;	 □ ccccc 19. Vous utilisez une boucle while quand: □ 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 n'avez pas déclaré de fonction □ l'incrément de la variable de boucle n'est pas 1 20. 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, int x); □ int pgcd(int x, y);
\Box jouer de la musique	for (i = 'A'; i <= 'F'; i = i + 1) {	□ void pgcd(int x, int y);

т		-1
	acence	

Prénom:	Nom:
N° etu :	
n etu.	

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
     □ Majeur
     ☐ Mineur
     □ rien
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 - 1 = racine(x):
     \square x = racine(racine(x)*racine(x));
     \square x = racine(2/3):
     \square x = racine(x * x) - racine(x);
3. 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;
☐ int %d:
```

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:

```
□ alphabétique
```

```
□ dans lequel ces fonctions sont appelées dans le
       main
    □ dans lequel vous avez déclaré ces fonction
    □ un ordre quelconque
5. 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();
    ☐ char afficher_menu(printf("menu"));
    ☐ int afficher_menu();
6. 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
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 == b
    \Box a = b
    \square a{n, x} == b{n, x}
```

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. 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
      \Box la case mémoire 8 contiendra 16
      □ le terminal affiche 8
      □ la case mémoire 8 contiendra 0
10. 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";
11. 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
      \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
```

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

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

```
15. Après exécution jusqu'à la ligne 14 du programme C:
      □ il comporte une boucle infinie
                                                                                                                         18. Si le code:
     \square il ne compile pas
                                                                                                                             struct toto_s
                                                              10
                                                                    int main() {
                                                              11
                                                                         int x = 5;
     □ il risque d'afficher bonjour à la place de coucou
                                                                                                                                int n;
                                                              12
     \square il n'affiche rien
                                                                                                                                double x;
                                                              13
                                                                         printf(" x = %d\n", 2);
                                                                                                                             };
                                                              14
13. Le code suivant :
                                                              15
                                                                         . . .
                                                                                                                             précède la fonction main(), alors on peut écrire en
                                                                   }
                                                              16
                                                                                                                             début de main() :
     int i;
     for (i = 0; i < 7; i = i + 2)
                                                                                                                               □ struct toto_s toto;
                                                                   \square le terminal affiche x = 5
                                                                                                                               \square int struct toto_s = {3, -1e10};
         printf("%d ", i);
                                                                  \square le terminal affiche 5
                                                                                                                               \square int toto.n = 3;
                                                                  \square le terminal affiche x = 2
    printf("\n");
                                                                                                                               \Box toto_s struct z = {3, 0.5};
                                                                  □ le terminal affiche "Faux"
                                                                                                                               \square toto_s n, x;
   affichera:
                                                                                                                         19. Soit la fonction f définie par :
                                                            16. Un bit est:
     \Box 02468
                                                                                                                             int f(int a)
     \Box 0 1 2 3 4 5 6 7
                                                                  ☐ l'instruction qui met fin à un programme
     \Box 0246
                                                                  □ la longueur d'un mot mémoire
                                                                                                                                printf("a = \n", %d);
                                                                                                                                if (a > 0)
     \Box 0123456
                                                                  □ un battement d'horloge processeur
                                                                  \square un chiffre binaire (0 ou 1)
14. Soit la fonction f définie par :
                                                                                                                                  return 3;
                                                            17. Le code suivant :
   int f(int a)
                                                                                                                                return 4;
                                                                  int age = 20;
      printf("a = \n", %d);
                                                                 if (age < 18)
                                                                                                                             Alors l'expression f(0) prendra la valeur :
      if (a > 0)
                                                                                                                               \square 3
                                                                      printf("Mineur\n");
        return f(a - 1) + 1;
                                                                                                                               \Box 4
                                                                  printf("Majeur\n");
                                                                                                                               \Box 0
      return 4;
                                                                 affichera:
                                                                                                                         20. Pour déclarer une fonction exposant qui prend en ar-
                                                                                                                             gument un réel x et un entier positif n et renvoie la
   Alors l'expression f(1) prendra la valeur :
                                                                  ☐ Mineur
                                                                                                                             valeur de x^n on écrit :
                                                                     Majeur
     \Box 4
                                                                                                                               \square void exposant(double x^n);
                                                                  \square rien
     \Box 1
                                                                                                                               \square exposant(double x, int n, int r);
                                                                  ☐ Mineur
     \Box 5
                                                                                                                               \square int exposant(double n, int x);
                                                                   □ Majeur
                                                                                                                               \square double exposant(double x, int n);
     \Box 0
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu :		

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 manquant \square une accolade manquante □ un point-virgule en trop \square une accolade en trop 2. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : \square analyse lexicale \square analyse harmonique \square analyse syntaxique □ analyse sémantique 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 □ 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 4. 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;

```
5. 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 void pgcd(int x, int y);
    \Box int pgcd(int x, int x);
    \Box int pgcd(int x, y);
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 || B)
    □ A && B
    ☐ (A == TRUE) && (B == TRUE)
    \square !(!A || B) == (A && !B)
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 :
    \square 0
    \Box 1
    \Box 5
    \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 :
      \Box 4
      \square 3
      \Box 0
 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);
     \Box printf("x=%d et y=%d\n",x y);
10. Si x est une variable réelle (de type double) alors
    x = 3/2 lui affecte la valeur :
      \square 1.5
      \Box 0
     \square 0.5
      \Box 1
11. 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
      ☐ une directive préprocesseur #include manquante
      □ un caractère interdit en C
      \square une faute de frappe dans un appel de fonction
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));
     □ void saisie_utilisateur(char c);
     ☐ int saisie_utilisateur();
      □ void saisie_utilisateur(int n);
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 :
     \square x = racine(2/3);
     \square x = racine(x * x) - racine(x):
     \square x = racine(racine(x)*racine(x));
      \square x - 1 = racine(x):
```

4.	Soit la fonction g définie par :
	<pre>int g(int a) {</pre>
	<pre>printf("a = \n", %d);</pre>
	if (1 > 0)
	{
	return 5;
	}
	return 7;
	}
	Alors l'expression g(0) prendra la valeur :
	\Box 7
	\Box 0
	\Box 5
5.	Le type des réels en C est :
	\square double
	\square real
	\square int
	□ char
6.	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 :

14. Soit la fonction g définie par :	☐ dans lequel ces fonctions sont appelées dans le main	18. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doit
<pre>int g(int a) {</pre>	un ordre quelconque	on chercher dans le programme?
printf("a = \n", %d);	☐ dans lequel vous avez déclaré ces fonction	$\hfill\Box$ une fonction déclarée mais non définie
if (1 > 0) {	\Box alphabétique	☐ un désaccord entre la déclaration et la définitio d'une fonction
return 5;	17. Le code suivant :	☐ une directive préprocesseur #include manquant
return 7;	int age = 20;	$\hfill\Box$ une fonction appelée avant sa déclaration
}	if (age < 18) {	19. Si pgcd est une fonction prenant en entrée deux entier et renvoyant un entier, il est correct d'écrire :
Alors l'expression g(0) prendra la valeur :	<pre>printf("Mineur\n");</pre>	,
□ 7 □ 0	else	□ n = pgcd(int p, int q);
	{	☐ int pgcd(2);
	<pre>printf("Majeur\n");</pre>	□ n = pgcd(n, 3);
15. Le type des réels en C est :	}	\square int n = pgcd();
double		20. Pour déclarer un tableau d'entiers de taille 5, on peu
□ real	affichera:	utiliser l'instruction
□ int	☐ Mineur	\square int toto[5];
□ char	Majeur	☐ int toto[taille=5];
16. Vous avez déclaré préalablement un ensemble de fonc-	☐ Mineur	☐ char tableau[5];
tions utilisées par votre programme principal. L'ordre dans lequel vous devez maintenant définir ces fonctions	☐ Majeur	\square int[] new tableau(5);
est l'ordre :	\square rien	<pre>□ int tab[] = 5;</pre>

18. Si cette erreur apparaît à la compilation :
erreur: conflicting types for 'max', $\operatorname{que}\operatorname{doit}$
on chercher dans le programme?
$\hfill \square$ une fonction déclarée mais non définie
$\hfill \square$ un désaccord entre la déclaration et la définition d'une fonction
$\hfill \square$ une directive préprocesseur $\mbox{\tt\#include}$ man quante
\Box une fonction appelée avant sa déclaration
19. 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);
☐ int pgcd(2);
\square n = pgcd(n, 3);
\square int n = pgcd();
20. Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction
☐ int toto[5];
☐ int toto[taille=5];
☐ char tableau[5];
☐ int[] new tableau(5);
☐ int tab[] = 5;

Éléments d'informatique – contrôle continue

Prénom:	Nom:	
N° etu:		

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: \square 4 3 2 1 \Box 4 3 2 1 0 \square 1 2 3 4 $\Box 01234$ 2. Vous utilisez une boucle while quand: □ vous avez déjà fait un for dans le même programme principal \square vous n'avez pas déclaré de fonction \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 3. 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 } \square le terminal affiche x = 2 \square le terminal affiche x = 5 □ le terminal affiche 5 □ le terminal affiche "Faux" 4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : \square analyse lexicale □ analyse sémantique \square analyse syntaxique

 \square analyse harmonique

5. L'écriture <u>101</u> en binaire correspond au nombre naturel :
□ 5
□ 101
□ 4
6. Si racine est une fonction prenant en entrée un rée et renvoyant la racine carrée de cet réel, et que x es une variable réelle définie et initialisée, il est incorrec 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);
7. Soit le programme principal suivant :
<pre>int main()</pre>
<pre>{ int a = 3; 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>
<pre>a = a + b; 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(3,5)=8, a=3, b=5
\Box f(a,b)=13, a=8, b=5
8. Quel est le problème d'un programme comportant le lignes suivantes?
while (1)
{
<pre>printf("coucou\n"); }</pre>

```
\square il comporte une boucle infinie
      \square il ne compile pas
      □ il n'affiche rien
      \square il risque d'afficher bonjour à la place de coucou
 9. 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
      □ retourner un bloc
      □ sélectionner entre deux blocs à l'aide d'une condi-
         tion
      □ répéter un bloc tant qu'une condition est vérifée
10. Un registre du processeur est :
      □ une gamme de fréquence de fonctionnement du
         processeur
      □ 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
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 5
      \square 7
      \square 0
```

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.	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);
	☐ int exposant(double n, int x);
	\square double exposant(double x, int n);
	☐ void exposant(double x^n);
14.	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
	□ les fichiers du disque
	□ certaines données de la mémoire de travail
	□ en temps d'accès
15.	Soit la fonction f définie par :
	<pre>int f(int a) {</pre>
	printf("a = \n", %d);
	if (a > 0) {
	return f(a - 1) + 1;
	}
	return 4;
	}

```
Alors l'expression f(1) prendra la valeur :
                                                             18. Au début de la fonction main() on place le code :
      \Box 4
                                                                  char b = 'A';
                                                                  b = b + 2;
      \Box 5
                                                                  printf("%c\n", b);
      \Box 0
                                                                 Alors l'affichage sera :
      \Box 1
                                                                   \square B
                                                                   \Box C
16. Le code suivant :
                                                                   \square A
     int i;
                                                                   □ъ
     for (i = 8; i > 0; i = i - 2)
                                                             19. Après la déclaration : int mccarthy(int n);, il est
          printf("%d ", i);
                                                                 correct d'écrire :
                                                                   ☐ int mccarthy(int 2);
     printf("\n");
                                                                   \square n = mccarthy(p, q);
    affichera:
                                                                   \square n = mccarthy();
                                                                   \square x = mccarthy(n);
      \square 8 2
                                                             20. Les lignes
      \Box 8 6 4 2 0
                                                                 int i;
      \Box 02468
                                                                 int x=0;
     \square 8 6 4 2
                                                                 for(i=0,i<5,i=i+1)
                                                                 {
17. Quelle étape de la compilation vient d'échouer lors-
                                                                    x=x+1;
    qu'on a un message comme celui-ci :
                                                                 }
    Undefined symbols :"_prinft" ou
    référence indéfinie vers « prinft »
                                                                   \square ne comportent aucune erreur
                                                                   \Box comportent une erreur qui sera détectée au cours
      ☐ l'édition de liens
                                                                      de l'édition de lien
     ☐ l'analyse des entrées clavier
                                                                   \Box comportent une erreur qui sera détectée au cours
      ☐ l'analyse harmonique
                                                                      de l'analyse syntaxique
                                                                   \Box comportent une erreur qui ne sera pas détectée
      ☐ l'analyse sémantique
```

Éléments d'informatique – contrôle continue

Prénom: Nom: N° etu :

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

```
1. 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
     \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
2. L'ordonnancement par tourniquet permet :
     □ 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
     ☐ d'entretenir l'illusion que les processus tournent
        en parallèle
3. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     □ 111
     \square 7
     \square 8
     \square 3
4. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses chants
     \square ses blocs
```

□ ses cases

```
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 6
    \Box 0
6. Dans la commande gcc, l'option -Wall signifie :
    □ qu'il faut indenter le fichier source
    □ qu'on veut changer alétoirement de fond d'écran
    □ qu'il faut lancer un déboggueur
    ☐ que l'on veut voir tous les avertissements
7. 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 01234
    \Box \ 4\ 3\ 2\ 1\ 0
    \square 4 3 2 1
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 : \Box x = racine(racine(x)*racine(x)): \square x - 1 = racine(x); \square x = racine(2/3); \square x = racine(x * x) - racine(x);

```
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. Le langage C est un langage
     □ compilé
     □ composé
     □ interprété
     □ lu, écrit, parlé
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)=13, 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)=8, a=8, b=5
12. Pour compiler un programme prog.c, on utilise la
   ligne de commande :
     ☐ gcc prog.exe -Wall -o prog.c
     ☐ gcc -Wall prog.c -o prog.exe
     ☐ gcc -Wall prog.exe -o prog.c
```

☐ gcc prog.c -o -Wall prog.exe

13. Si carre est une fonction prenant en entrée un en-	\square une accolade en trop	\square double exposant(double x, int n);
tier et renvoyant le carré de cet entier, et que n est	\square un point-virgule en trop	☐ void exposant(double x^n);
une variable entière définie et initialisée, il est correct d'écrire :	□ un point-virgule manquant	☐ int exposant(double n, int x);
\square n = carre(n);	17. Soit un programme contenant les lignes suivantes :	19. Pour déclarer une procédure afficher_menu sans ar-
☐ int n = carre();	int i = 0;	gument et qui ne renvoie rien on utilise :
\square n = carre(int n);	int j = 0; for (i = 0; i < 0; i = i + 1)	☐ int afficher_menu();
☐ int carre(2);	101 (1 - 0; 1 < 0; 1 - 1 + 1) {	☐ double afficher_menu();
14. L'écriture <u>101</u> en binaire correspond au nombre natu-	for (j = 0; j < 5; j = j + 1)	☐ int afficher_menu(int char);
rel:	i	☐ char afficher_menu(printf("menu"));
□ 101 □ a	}	□ void afficher_menu();
\square 3	}	□ void afficher_menu(),
\square 5	printf("j = %d\n", j);	20. Au début de la fonction main() on place le code :
\Box 4		char i;
15. Avant de faire appel à une fonction il est nécessaire de :	}	for (i = 'A'; i <= 'F'; i = i + 1)
\square avoir défini une constante symbolique de la taille	qu'est ce qui sera affiché?	<pre>printf("%c", i);</pre>
de cette fonction	□ j = 0	}
\square l'avoir déclarée et définie	□ j = %d	<pre>printf("\n");</pre>
□ l'avoir définie	□ j = 5	Alors l'affichage sera :
□ l'avoir déclarée	□ j = 4	☐ ABCDEF
16. Si cette erreur apparaît à la compilation :	18. Pour déclarer une fonction exposant qui prend en ar-	□i
error: expected ';' before '}' token que doit- on chercher dans le programme?	gument un réel x et un entier positif n et renvoie la valeur de x^n on écrit :	□ A
$\hfill\Box$ une accolade man quante	\square exposant(double x, int n, int r);	□ сссссс

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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 = 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 \square la variable y vaut 5 \square le programme affiche "Faux" 2. Le code suivant : int age = 18; if (age < 18) { printf("Mineur\n"); } else { printf("Majeur\n"); } affichera: □ Majeur □ Mineur Majeur ☐ Mineur □ rien 3. 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 char); □ void afficher_menu();

☐ int afficher_menu();

```
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:
    □ a == b
    \Box a = b
    \square a{n, x} == b{n, x}
    \square (a.n == b.n) \&\& (a.x == b.x)
5. Le type des réels en C est :
    □ char
    ☐ double
    \square real
    □ int.
6. 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":
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(struct date_s d);
    ☐ int afficher_date(date_s d);
    □ void afficher_date(date_s d);
    ☐ struct date_s afficher_date(struct date_s
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:
    □ b
    □ A
    \square B
    \Box C
```

9.	Vous utilisez une boucle while quand :
	\Box vous avez déjà fait un for dans le même programme principal
	\Box l'incrément de la variable de boucle n'est pas 1
	$\hfill \square$ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
	\square vous n'avez pas déclaré de fonction
10.	Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :
	\Box analyse syntaxique
	\Box analyse sémantique
	\Box analyse lexicale
	$\hfill\Box$ analyse harmonique
11.	Un bit est:
	\Box l'instruction qui met fin à un programme
	\Box un chiffre binaire (0 ou 1)
	\Box la longueur d'un mot mémoire
	\Box un battement d'horloge processeur
12.	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);
1);	\square exposant(double x, int n, int r);
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 :
	\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");
	\Box printf("x=%d et y=%d\n",x y);

14. Un enregistrement permet de grouper plusieurs valeurs	16. Si n est une variable entière pour demander sa valeur
dans:	à l'utilisateur, on utilise plutôt :
\square ses chants	<pre>□ scanf("%d", &n);</pre>
\square ses blocs	☐ printf("Valeur de n ? %d\n", n);
\square ses champs	\square un débogueur
□ ses cases	☐ printf("Valeur de n ? %g\n", n);
15. Soit un programme contenant les lignes suivantes :	17. Quel est l'opérateur de différence en C :
int i = 0;	□ !=
int j = 0;	□ !
for (i = 0; i < 0; i = i + 1)	□≠
{ for $(j = 0; j < 5; j = j + 1)$	
{	18. Pour l'extrait de programme suivant :
•••	<pre>int somme = 0;</pre>
}	int serie[4] = {2, 4, 10, 4};
}	for $(i = 0; i < 4; i = i + 1)$
printf("j = %d\n", j);	{
•••	somme = somme + serie[i];
}	}
	<pre>printf("somme = %d",somme);</pre>
qu'est ce qui sera affiché?	La valeur de somme affichée est :
□ j = 5	\Box 6
□ j = %d	\square 3
□ j = 4	\Box 16
□ j = 0	\square 20

```
19. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #include <stdio.h>
     ☐ #appart <stdlib.h>
     \square #include <studlib.h>
     ☐ #include <studio.h>
20. 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
     \Box 10
     \Box 15
```

 \Box 0

 \Box 16

 \square 8

Éléments d'informatique – contrôle continue

 \Box 5

Prénom:	Nom:
N° etu :	

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 □ ses cases 2. Un fichier source est: □ un fichier texte qui sera traduit en instructions processeur ☐ un document illisible pour les humains □ 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. 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 □ les fichiers du disque □ en temps d'accès 4. 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

```
5. 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
    \Box l'incrément de la variable de boucle n'est pas 1
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 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
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 4
    \Box 1
    \square 0
```

```
8. Une variable booléenne est un variable :
      □ jamais nulle
      □ NaN (not a number, qui n'est pas un nombre)
      ☐ réelle positive
      □ à laquelle une valeur vient d'être affectée
      \square qui est vraie ou fausse
 9. Le code suivant :
     int age = 20;
     if (age < 18)
     {
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
    affichera:
     □ Mineur
     □ Majeur
      ☐ Mineur
        Majeur
      □ rien
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 :
     \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);
11. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
     ☐ int toto[taille=5];
     \square int tab[] = 5;
     \square int toto[5];
      ☐ char tableau[5];
      \square int[] new tableau(5);
```

12. L'écriture <u>111</u> en binaire correspond au nombre natu-	15. Le bus système sert à :	18. L'écriture $\underline{101}$ en binaire correspond au nombre natu-
rel : □ 8	☐ Transférer des données et intructions entre processeur et mémoire	rel: \Box 5
	\Box transporter les processus du tourniquet au pro-	□ 4
□ 7	cesseur □ Écrire des données sur le dique dur	□ 3 □ 101
□ 111	☐ Arriver à l'heure en cours	19. Si racine est une fonction prenant en entrée un réel
13. Un programme en langage C doit comporter une et une seule définition de la fonction :	16. Au début de la fonction main() on place le code :	et renvoyant la racine carrée de cet réel, et que x est une variable réelle définie et initialisée, il est incorrect
\Box main	char b = 'A'; b = b + 2;	d 'écrire: $\Box x = racine(racine(x)*racine(x));$
\Box begin	<pre>printf("%c\n", b);</pre>	
□ init	Alors l'affichage sera :	\Box x = racine(x * x) - racine(x);
\Box include	□ b	\Box x - 1 = racine(x);
14. Quel est le problème d'un programme comportant les	□ A	20. Au début de la fonction main() on place le code :
lignes suivantes?	□ C	char i;
while (1)	□ В	for (i = 'A'; i <= 'F'; i = i + 1) {
<pre>t printf("coucou\n");</pre>	17. Si carre est une fonction prenant en entrée un entier et renvoyant le carré de cet entier, et que n est	<pre>printf("%c", i);</pre>
}	une variable entière définie et initialisée, il est correct d'écrire : □ int carre(2);	<pre>} printf("\n");</pre>
\square il risque d'afficher bonjour à la place de coucou		Alors l'affichage sera :
☐ il comporte une boucle infinie ☐ il ne compile pas		□ A
	<pre>□ n = carre(int n);</pre> □ n = carre(n);	□i
□ il n'affiche rien	☐ in = carre(n); ☐ int n = carre();	
	□ int n - carre(),	□ ABCDEF

Éléments d'informatique – contrôle continue

□ rien

Prénom: Nom: N° etu :

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
     ☐ int toto[taille=5];
     \square int tab[] = 5;
     ☐ int[] new tableau(5);
     \square int toto[5]:
     □ char tableau[5];
2. 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
     \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
     ☐ 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
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:
     □ A
     Πi
     ☐ ABCDEF
     □ cccccc
4. Si n est une variable entière pour demander sa valeur
   à l'utilisateur, on utilise plutôt :
     □ scanf("%d", &n);
     \square printf("Valeur de n ? %g\n", n);
     \square un débogueur
```

□ printf("Valeur de n ? %d\n", n);

```
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 :
    \Box 5
    \Box 7
    \square 0
6. Lorsqu'un programme utilise printf ou scanf il faut
  qu'il contienne l'instruction préprocesseur :
    ☐ #include <studlib.h>
    ☐ #appart <stdlib.h>
    ☐ #include <stdio.h>
    ☐ #include <studio.h>
7. Le code suivant :
   int age = 18;
   if (age < 18)
        printf("Mineur\n");
   }
   else
   {
        printf("Majeur\n");
   }
  affichera:
    ☐ Mineur
    □ Mineur
       Majeur
    □ Majeur
```

```
8. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 3
      \Box 5
      \square 101
      \Box 4
 9. Le code suivant :
     int i;
     for (i = 4; i \ge 0; i = i - 1)
          printf("%d ", i);
     }
     printf("\n");
    affichera:
     \Box 1234
     \Box 43210
     \square 4 3 2 1
     \Box 01234
10. Le type des réels en C est :
      ☐ double
      □ int
      □ real
      □ char
11. 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 y vaut 5
      \square la variable x vaut 3
      ☐ le programme affiche "Faux"
      \square la variable x vaut 5 et la variable y vaut 3
```

```
12. Soit un programme contenant les lignes suivantes :
                                                                    □ il n'affiche rien
                                                                                                                           18. Le code suivant :
                                                                    \square il ne compile pas
     int i = 0;
                                                                                                                                int somme = 0;
                                                                   \square il comporte une boucle infinie
     int j = 0;
                                                                                                                                int i;
     for (i = 0; i < 0; i = i + 1)
                                                                    □ il risque d'afficher bonjour à la place de coucou
                                                                                                                                for (i = 1; i < 4; i = i + 1)
                                                              15. Laquelle de ces écritures correspond à la déclaration
         for (j = 0; j < 5; j = j + 1)
                                                                                                                                   somme = somme + i;
                                                                  d'une variable de type caractère en langage C?
                                                                    □ char 'c';
                                                                                                                                printf("%d", somme);
                                                                    ☐ char "c";
                                                                    □ char c;
     printf("j = %d\n", j);
                                                                                                                               affichera:
                                                                    ☐ int char;
                                                                                                                                 \square 42
                                                              16. On souhaite faire une boucle de contrôle de saisie : tant
                                                                 que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                                                                                 \Box 6
                                                                  recommence la saisie de n. Soit le programme suivant :
    qu'est ce qui sera affiché?
                                                                                                                                  \square 1
                                                                   int a = 0;
      \Box j = 0
                                                                   int b = 20;
                                                                                                                                 \Box 0
      \Box j = 4
                                                                   int n;
      \Box j = %d
                                                                   scanf("%d", &n);
                                                                                                                           19. Laquelle des analyses suivantes ne fait pas partie des
      \Box j = 5
                                                                                                                                étapes de la compilation :
                                                                   while(cond)
13. Vous avez déclaré préalablement un ensemble de fonc-
                                                                                                                                 \square analyse syntaxique
                                                                     scanf("%d", &n);
    tions utilisées par votre programme principal. L'ordre
                                                                                                                                 \square analyse harmonique
    dans lequel vous devez maintenant définir ces fonctions
    est l'ordre :
                                                                  Quelle est la condition cond :
                                                                                                                                 \square analyse lexicale
      \square un ordre quelconque
                                                                    \square (n<=a) && (n<=b)
                                                                                                                                 □ analyse sémantique
      □ alphabétique
                                                                    \square (a<n) || (n>b)
      \square dans lequel vous avez déclaré ces fonction
                                                                    \square (a<=n) && (n<=b)
                                                                                                                           20. Vous utilisez une boucle while quand:
      \square dans lequel ces fonctions sont appelées dans le
                                                                    □ a<=n<=b
                                                                                                                                 □ vous ne connaissez pas le nombre d'itérations de
         main
                                                             17. Une variable booléenne est un variable :
                                                                                                                                    la boucle à l'avance
14. Quel est le problème d'un programme comportant les
                                                                   □ à laquelle une valeur vient d'être affectée
                                                                                                                                 □ vous avez déjà fait un for dans le même pro-
    lignes suivantes?
                                                                    ☐ réelle positive
                                                                                                                                    gramme principal
    while (1)
                                                                   □ NaN (not a number, qui n'est pas un nombre)
                                                                                                                                  □ vous n'avez pas déclaré de fonction
                                                                    □ jamais nulle
      printf("coucou\n");
                                                                                                                                  ☐ l'incrément de la variable de boucle n'est pas 1
                                                                    □ qui est vraie ou fausse
```

т		-1
	acence	

while (1)

printf("coucou\n");

Éléments d'informatique – contrôle continue

 \square A

□ b

Prénom:	Nom:
N° etu:	

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 = racine(2/3): \square x = racine(x * x) - racine(x); \square x = racine(racine(x)*racine(x)): \square x - 1 = racine(x): 2. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise: □ void afficher_menu(); ☐ char afficher_menu(printf("menu")); ☐ int afficher_menu(); ☐ int afficher_menu(int char); ☐ double afficher_menu(); 3. 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 déclarée mais non définie □ un désaccord entre la déclaration et la définition d'une fonction \square une fonction appelée avant sa déclaration 4. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation : \square analyse syntaxique \square analyse harmonique □ analyse sémantique \square analyse lexicale 5. Quel est le problème d'un programme comportant les lignes suivantes?

```
\square il ne compile pas
     ☐ il risque d'afficher bonjour à la place de coucou
     □ il comporte une boucle infinie
    □ il n'affiche rien
6. 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
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);
    \square printf("x=%d et y=%d\n,x,y");
    \Box printf("x=%x et y=%y\n");
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:
    \Box C
    □В
```

```
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 4
      \Box 0
     \Box 5
     \Box 1
10. 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
     \square en parallèle, chacun dans un registre
11. Le langage C est un langage
     □ lu, écrit, parlé
     □ composé
     □ interprété
     □ compilé
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)
                                                            15. Pour l'extrait de programme suivant :
                                                                  int somme = 0;
      a = a + b;
                                                                  int serie[4] = \{2, 4, 10, 4\};
      return a;
                                                                  for (i = 0; i < 4; i = i + 1)
                                                                  {
                                                                     somme = somme + serie[i];
    L'affichage dans le main est le suivant :
     \Box f(a,b)=8, a=8, b=5
                                                                  printf("somme = %d",somme);
     \Box f(3,5)=8, a=3, b=5
                                                                La valeur de somme affichée est :
     \Box f(a,b)=8, a=3, b=5
                                                                  \square 3
     \Box f(a,b)=13, a=8, b=5
                                                                  \square 20
                                                                  \Box 16
13. Le type des réels en C est :
                                                                  \Box 6
      \square int
                                                            16. Dans la commande gcc, l'option -Wall signifie :
      □ char
                                                                  \hfill\Box qu'il faut indenter le fichier source
      \square double
                                                                  \square que l'on veut voir tous les avertissements
      \square real
                                                                  \Box qu'on veut changer alétoirement de fond d'écran
14. Lorsqu'un programme utilise printf ou scanf il faut
                                                                  \square qu'il faut lancer un déboggueur
    qu'il contienne l'instruction préprocesseur :
                                                            17. Sur unix (ou linux), la commande mkdir permet de :
      ☐ #include <studio.h>
                                                                                                                              □ void afficher_date(date_s d);
                                                                  \square changer de répertoire courant
      ☐ #include <studlib.h>
                                                                                                                              ☐ struct date_s afficher_date(struct date_s d);
                                                                  □ créer un fichier texte
      ☐ #include <stdio.h>
                                                                                                                              ☐ int afficher_date(date_s d);
                                                                  \Box ouvrir un fichier texte
      ☐ #appart <stdlib.h>
                                                                  □ créer un répertoire
                                                                                                                              □ void afficher_date(struct date_s d);
```

18.	Pour déclarer un tableau d'entiers de taille 5, on peut utiliser l'instruction
	☐ int toto[5];
	☐ int tab[] = 5;
	☐ int[] new tableau(5);
	☐ char tableau[5];
	☐ int toto[taille=5];
19.	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 carre(2);
	☐ int n = carre();
	\square n = carre(n);
	□ n = carre(int n);
20.	Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu du struct, on écrit :

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

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

réponse fausse. Durée : 20 minutes.
1. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.
\square #define taille = 3
☐ #define N 3
□ #define N = 3
\square #define taille = N
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. 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 y, int x);
☐ int pgcd(int x, int x);
☐ void pgcd(int x, int y);
4. Un enregistrement permet de grouper plusieurs valeurs dans :
\square ses champs
□ ses cases
\square ses chants
\square ses blocs
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 :
☐ for(i=0;i<5;i=i+1)
☐ 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)

```
6. 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();
 7. 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();
 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 :
      □В
      \square A
      \Box C
      □ b
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 n = carre(int n);
     \square n = carre(n);
      \square int n = carre();
      \square int carre(2);
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. 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. 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. 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 :	16. Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur : ☐ #include <studio.h> ☐ #include <studlib.h></studlib.h></studio.h>	<pre>18. Après la déclaration : int mccarthy(int n);, il est correct d'écrire : □ int mccarthy(int 2); □ x = mccarthy(n);</pre>
□ alphabétique □ dans lequel vous avez déclaré ces fonction □ un ordre quelconque	☐ #appart <stdlib.h> ☐ #include <stdio.h> 17. Le code suivant :</stdio.h></stdlib.h>	□ n = mccarthy(p, q); □ n = mccarthy(); 19. Pour déclarer une fonction exposant qui prend en ar-
☐ dans lequel ces fonctions sont appelées dans le main 14. Sous unix (ou linux), la commande 1s permet de :	<pre>int age = 20; if (age < 18) { printf("Mineur\n");</pre>	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);
 □ compiler un programme □ voir des clips musicaux □ afficher le contenu d'un fichier texte □ afficher la liste de fichiers contenus dans un 	<pre>} else { printf("Majeur\n");</pre>	 □ double exposant(double x, int n); □ void exposant(double x^n); □ exposant(double x, int n, int r); 20. Si cette erreur apparaît à la compilation :
répertoire 15. Le langage C est un langage compilé interprété lu, écrit, parlé	affichera: Mineur Majeur rien Majeur	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
□ composé	☐ Mineur	d'une fonction

т		-1
	acence	

Prénom :	Nom:	
N° etu :		

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:
     \square 0 1 2 3 4 5 6 7
     \Box 02468
     \Box 0123456
     \Box 0246
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 :
     □ certaines données de la mémoire de travail
    \Box en temps d'accès
     \square des processus
     □ les fichiers du disque
4. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut lancer un déboggueur
     □ 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
```

```
5. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 3
     □ 111
     \square 8
     \square 7
6. 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);
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)
     \square for(i=0:i<5:i=i+1)
     \square for(i=0;i<=5;i=i+1)
8. 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 une fonction déclarée mais non définie
     □ un désaccord entre la déclaration et la définition
        d'une fonction
     ☐ une fonction appelée avant sa déclaration
9. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \Box 1
     \Box 0
     \square 1.5
     \square 0.5
```

```
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 :
      \square 5
      \square 7
      \square 0
11. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \square 101
      \square 5
      \square 3
      \Box 4
12. Avant de faire appel à une fonction il est nécessaire
    de:
      ☐ l'avoir déclarée et définie
      ☐ l'avoir définie
      □ l'avoir déclarée
      \square avoir défini une constante symbolique de la taille
         de cette fonction
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();
      ☐ int factorielle(int 2);
```

□ printf("%d", factorielle(n));

□ un document qui doit être protégé return EXIT_SUCCESS; 16	
□ un fichier texte qui sera traduit en instructions processeur 17 17 18 19 19 19	y vaut 0
□ un document de référence du système int f(int a, int b) □ la variable x vaut 0	
\Box un document illisible pour les humains \Box la variable y vaut 5	
□ un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ le programme affiche "Faux" □ le programme affiche "Faux"	
15. Le langage C est un langage 20. Soit un programme contenant les lignes	suivantes :
L'affichage dans le main est le suivant : int i = 0;	
$\square \text{ compilé} \qquad \qquad \square \text{ f(a,b)=8, a=3, b=5} \qquad \qquad \text{int } j=0;$	
$\square \text{ lu, \'ecrit, parl\'e}$	
)
16. Si cet avertissement apparaît à la compilation :	,
warning: implicit declaration of function 'max', 18. Si pgcd est une fonction prenant en entrée deux entiers , que doit-on chercher dans le programme? 18. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire :	
□ une fonction déclarée mais non définie	
\Box un désaccord entre la déclaration et la définition \Box n = pgcd(n, 3);	
d'une fonction \Box int n = pgcd();	
☐ une fonction appelée avant sa déclaration ☐ int pgcd(2);	
□ une directive préprocesseur #include manquante 19. Après exécution jusqu'à la ligne 15 du programme C : qu'est ce qui sera affiché?	
17. Soit le programme principal suivant : 10 int main() {	
int main()	
int a = 3;	
int b = 5;	

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	1,0111
n etu:	

```
1. 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 j = 5
    \Box j = 0
     \Box j = %d
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);
    \square n = factorielle();
    ☐ printf("%d", factorielle(n));
     ☐ int factorielle(int 2);
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 :
     \square 3
     \Box 0
     \Box 4
```

```
4. 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
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 :
     \square 16
     \Box 4
     \Box 0
     \square 8
6. 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
     □ en parallèle, chacun dans un registre
     □ tour à tour, un petit peu à chaque fois
7. Quels calculs peut-on programmer en programmation
   structurée?
     □ certains programmes sont de vrais plats de spa-
        ghetti
     \square il y a des calculs programmables en langage ma-
        chine et qui ne sont pas programmables en pro-
        grammation structurée
    ☐ 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
```

8.	Un programme en langage C doit comporter une et une seule définition de la fonction :
	□ begin
	□ init
	\Box include
	\square main
9.	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
	□ъ
	□ A
	□ С
10.	Dans la commande gcc, l'option -Wall signifie :
	$\hfill\Box$ qu'il faut indenter le fichier source
	\Box qu'on veut changer alétoirement de fond d'écran
	\Box que l'on veut voir tous les avertissements
	$\hfill \square$ qu'il faut lancer un déboggueur
11.	Si cette erreur apparaît à la compilation : error: expected ';' before '}' token que doit-on chercher dans le programme?
	\Box une accolade man quante
	\Box un point-virgule man quant
	\Box une accolade en trop
	\Box un point-virgule en trop

```
12. Le code suivant :
     int age = 15;
     if (age < 18)
     {
         printf("Mineur\n");
     }
     else
     {
         printf("Majeur\n");
     }
    affichera:
      \square rien
      □ Majeur
      ☐ Mineur
         Majeur
      ☐ Mineur
13. Pour déclarer un tableau d'entiers de taille 5, on peut
    utiliser l'instruction
      ☐ int toto[taille=5];
      \square int[] new tableau(5);
      \square int tab[] = 5;
      □ char tableau[5];
      \square int toto[5]:
14. Après la déclaration : int mccarthy(int n);, il est
    correct d'écrire :
      \square n = mccarthy(p, q);
      \square x = mccarthy(n);
     ☐ int mccarthy(int 2);
      \square n = mccarthy();
```

```
15. Si pgcd est une fonction prenant en entrée deux entiers
                                                                       {
    et renvoyant un entier, il est correct d'écrire :
                                                                            printf("%d ", j);
                                                                       }
      \square int pgcd(2);
                                                                  }
     \square n = pgcd(int p, int q);
     \square n = pgcd(n, 3);
                                                                 qu'est ce qui sera affiché?
     \square int n = pgcd();
                                                                   \Box 0 1 2 0 1 2
16. Vous avez déclaré préalablement un ensemble de fonc-
                                                                   \Box 0 0 1 1 2 2 3
    tions utilisées par votre programme principal. L'ordre
    dans lequel vous devez maintenant définir ces fonctions
                                                                   \Box 0 1 2 0 1 2 3
    est l'ordre :
                                                                   \Box 0 1 2 3 0 1 2
     □ dans lequel vous avez déclaré ces fonction
      □ dans lequel ces fonctions sont appelées dans le
                                                             19. Laquelle de ces écritures correspond à la déclaration
         main
                                                                 d'une variable de type caractère en langage C?
     □ alphabétique
                                                                   ☐ char "c";
     \square un ordre quelconque
                                                                   □ char 'c';
17. Une variable booléenne est un variable :
                                                                   \Box char c;
     \Box à la
quelle une valeur vient d'être affectée
                                                                   ☐ int char;
      ☐ réelle positive
     □ NaN (not a number, qui n'est pas un nombre)
                                                             20. Pour déclarer une fonction pgcd qui calcule et renvoie
     □ jamais nulle
                                                                 le plus grand diviseur commun de deux entiers positifs
      □ qui est vraie ou fausse
                                                                 passés en arguments on écrit :
18. Pour l'extrait de programme suivant :
                                                                   \square int pgcd(int y, int x);
     int i = 0;
                                                                   \square int pgcd(int x, y);
     int j = 0;
                                                                   \Box int pgcd(int x, int x);
     for (i = 0; i < 2; i = i + 1)
     {
                                                                   \square void pgcd(int x, int y);
         for (j = 0; j < 3; j = j + 1)
```

т	•	-1
- 1	acence	- 1

Prénom:	Nom:
N° etu :	

```
1. Le code suivant :
    int age = 20;
    if (age < 18)
        printf("Mineur\n");
   printf("Majeur\n");
  affichera:
    ☐ Mineur
    □ Majeur
    \square rien
    ☐ Mineur
       Majeur
2. Sous unix (ou linux), pour créer un répertoire TP4
  dans le répertoire courant on peut utiliser la com-
  mande:
    ☐ kwrite TP4
    □ mkdir TP4
    ☐ yppasswd
    □ new TP4
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
    \square 3
    \square 0
```

```
4. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses blocs
     □ ses cases
     \square ses champs
     \square ses chants
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) && (n<=b)
     □ a<=n<=b
     □ (a<n) || (n>b)
     □ (n<=a) && (n<=b)
6. Laquelle des analyses suivantes ne fait pas partie des
   étapes de la compilation :
     \square analyse syntaxique
     \square analyse lexicale
     □ analyse sémantique
     \square analyse harmonique
7. Un registre du processeur est :
     \Box une gamme de fréquence de fonctionnement du
        processeur
     \square une case mémoire interne au processeur qui sera
        manipulée directement lors des calculs
     \square un composant qui contient la liste des fichiers du
        système
     \square une unité de calcul spécialisée de l'ordinateur
```

```
8. Lorsqu'un programme utilise printf ou scanf il faut
   qu'il contienne l'instruction préprocesseur :
     ☐ #appart <stdlib.h>
     ☐ #include <studlib.h>
     ☐ #include <studio.h>
     ☐ #include <stdio.h>
9. 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 0 1 2 3
     \Box 0 1 2 0 1 2
     \Box 0 1 2 3 0 1 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 :
     \Box 0
     \Box 5
     \Box 1
     \Box 4
```

```
11. Si a et b sont deux variables de type:
                                                                 \Box 0 0 0 1 1 1
                                                                                                                          lignes suivantes?
                                                                 \Box 0 1 2 0 1 2
   struct toto_s
                                                                 \Box 1 2 1 2 3
                                                                                                                          while (1)
      int n;
                                                                 \Box 0 1 0 1 0 1 0 1
      double x;
                                                           14. Soit la fonction g définie par :
                                                                                                                             printf("coucou\n");
   };
                                                               int g(int a)
   Alors pour tester l'égalité de a et de b on utilise la
                                                               {
   condition:
                                                                 printf("a = \n", %d);
                                                                 if (1 > 0)
                                                                                                                            \square il ne compile pas
     \square a = b
                                                                 {
     \square a{n, x} == b{n, x}
                                                                                                                            □ il n'affiche rien
                                                                    return 5;
     \square a == b
     \square (a.n == b.n) && (a.x == b.x)
                                                                 return 7;
12. Après exécution du programme :
                                                               Alors l'expression g(0) prendra la valeur :
       lecture 8 r0
                                                                 \square 0
       valeur 3 r1
                                                                 \Box 7
      mult r1 r0
                                                                 \square 5
      valeur 1 r2
                                                           15. Pour déclarer une procédure afficher_date qui prend
       add r2 r0
                                                               en argument un struct date_s et affiche le contenu
       ecriture r0 8
                                                               du struct, on écrit:
       stop
                                                                 □ struct date_s afficher_date(struct date_s d);
      5
                                                                                                                            int i;
                                                                 ☐ int afficher_date(date_s d);
     □ la case mémoire 8 contiendra 16
                                                                                                                            int j;
                                                                 □ void afficher_date(date_s d);
     □ le terminal affiche 8
                                                                                                                            for(i=4;i>0;i=i-1)
                                                                 □ void afficher_date(struct date_s d);
     \square la case mémoire 8 contiendra 0
                                                           16. Vous utilisez une boucle while quand :
                                                                                                                              for(j=i;j<6;j=j+1)
     \square le bus explose
                                                                 □ vous n'avez pas déclaré de fonction
                                                                 □ vous ne connaissez pas le nombre d'itérations de
13. Soit un programme contenant les lignes suivantes :
                                                                                                                                printf("*");
                                                                    la boucle à l'avance
     int i = 0;
                                                                 □ vous avez déjà fait un for dans le même pro-
                                                                                                                              printf(" ");
     int j = 0;
                                                                    gramme principal
     for (i = 0; i < 2; i = i + 1)
                                                                 \Box l'incrément de la variable de boucle n'est pas 1
                                                           17. Avant de faire appel à une fonction il est nécessaire
         for (j = 0; j < 3; j = j + 1)
                                                                                                                          qu'est ce qui sera affiché?
                                                               de:
                                                                                                                              ** ** ** ** **
                                                                 □ l'avoir définie
              printf("%d ", i);
                                                                 □ avoir défini une constante symbolique de la taille
                                                                                                                              ** *** **** *****
    }
                                                                    de cette fonction
                                                                 \Boxl'avoir déclarée et définie
   qu'est ce qui sera affiché?
                                                                 □ l'avoir déclarée
```

```
18. Quel est le problème d'un programme comportant les
      \Box il comporte une boucle infinie
      ☐ il risque d'afficher bonjour à la place de coucou
19. 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
      \square que l'on veut voir tous les avertissements
20. Pour l'extrait de programme suivant :
```

т		-1
	100000	- 1
	acence	

Prénom:	Nom:	
N° etu :		

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 16 \square le bus explose □ le terminal affiche 8 □ la case mémoire 8 contiendra 0 2. 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 \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 \square 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. Le langage C est un langage □ composé □ interprété □ lu, écrit, parlé □ compilé 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: □ alphabétique

□ dans lequel vous avez déclaré ces fonction

```
□ dans lequel ces fonctions sont appelées dans le
        main
     \square un ordre quelconque
5. Quel est le problème d'un programme comportant les
   lignes suivantes?
   while (1)
     printf("coucou\n");
     □ il n'affiche rien
     □ il comporte une boucle infinie
     ☐ il risque d'afficher bonjour à la place de coucou
     \square il ne compile pas
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. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
     □ créer un fichier texte
     □ changer de répertoire courant
     □ ouvrir un fichier texte
8. Une variable booléenne est un variable :
     ☐ jamais nulle
     □ 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
```

```
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 = 0
     \Box j = %d
     \Box i = 5
     \Box j = 4
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
     ☐ des processus
     □ certaines données de la mémoire de travail
     □ en temps d'accès
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. 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);

```
13. Le code suivant :
    int age = 20;
    if (age < 18)
    {
         printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     ☐ Mineur
        Majeur
     □ Majeur
     ☐ Mineur
     \square rien
14. 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
15. 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);
```

```
16. Pour déclarer une fonction exposant qui prend en ar-
                                                             19. Après exécution jusqu'à la ligne 14 du programme C:
    gument un réel x et un entier positif n et renvoie la
                                                                    int main() {
                                                               10
    valeur de x^n on écrit :
                                                               11
                                                                         int x = 5;
      \square int exposant(double n, int x);
                                                               12
      \square double exposant(double x, int n);
                                                               13
                                                                         printf(" x = %d\n", 2);
      \square void exposant(double x^n);
                                                               14
      \square exposant(double x, int n, int r);
                                                               15
                                                                          . . .
17. Si cette erreur apparaît à la compilation :
                                                               16
                                                                    }
    erreur: conflicting types for 'max', que doit-
                                                                   \square le terminal affiche x = 2
    on chercher dans le programme?
      ☐ une fonction appelée avant sa déclaration
                                                                   \square le terminal affiche "Faux"
      \square un désaccord entre la déclaration et la définition
                                                                   \square le terminal affiche x = 5
         d'une fonction
                                                                   \square le terminal affiche 5
      □ une fonction déclarée mais non définie
      ☐ une directive préprocesseur #include manquante
                                                             20. Soit la fonction f définie par :
18. Pour l'extrait de programme suivant :
     int i = 0;
                                                                 int f(int a)
     int j = 0;
     for (i = 0; i < 3; i = i + 1)
                                                                   printf("a = \n", %d);
                                                                   if (a > 0)
          for (j = 0; j < 2; j = j + 1)
                                                                   {
                                                                      return 3;
              printf("%d ", i);
          }
                                                                   return 4;
                                                                 }
     printf("\n");
                                                                 Alors l'expression f(0) prendra la valeur :
    qu'est ce qui sera affiché?
      \Box 0 0 1 1 2 2
                                                                   \square 3
      \Box 1 2 3 1 2
                                                                   \Box 4
      \Box 0 1 2 0 1 2
                                                                   \Box 0
      \Box 0 1 0 1 0 1
```

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

□ comportent une erreur qui sera détectée au cours

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 :

```
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 (a<n) || (n>b)
    □ a<=n<=b
    \square (n<=a) && (n<=b)
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:
    \Box a = b
    □ a == b
    \square (a.n == b.n) && (a.x == b.x)
    \square a{n, x} == b{n, x}
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)=13, a=8, b=5
    \Box f(a,b)=8, a=8, b=5
    \Box f(a,b)=8, a=3, b=5
4. 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 0
    \Box 10
5. Avant de faire appel à une fonction il est nécessaire
    □ l'avoir déclarée
    □ l'avoir déclarée et définie
    ☐ l'avoir définie
    □ avoir défini une constante symbolique de la taille
       de cette fonction
6. Les lignes
  int i;
  int x=0:
  for(i=0,i<5,i=i+1)
     x=x+1;
```

 \square ne comportent aucune erreur

```
de l'analyse syntaxique
     □ comportent une erreur qui ne sera pas détectée
     □ comportent une erreur qui sera détectée au cours
        de l'édition de lien
7. Le code suivant :
    int age = 15;
    if (age < 18)
         printf("Mineur\n");
    else
    {
         printf("Majeur\n");
   affichera:
     □ Mineur
    □ Mineur
        Majeur
     □ Majeur
     \square rien
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 void pgcd(int x, int y);
     \square int pgcd(int y, int x);
     \square int pgcd(int x, int x);
     \square int pgcd(int x, y);
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 déclarer une procédure afficher_date qui prend	affichera :	18. Le code suivant :
en argument un struct date_s et affiche le contenu du struct, on écrit :	□ rien	int i;
<pre>void afficher_date(date_s d);</pre>	☐ Mineur	for $(i = 4; i \ge 0; i = i - 1)$
☐ struct date_s afficher_date(struct date_s	d); Majeur	{ printf("%d ", i);
☐ int afficher_date(date_s d);	□ Majeur	}
☐ void afficher_date(struct date_s d);	□ Mineur	<pre>printf("\n");</pre>
11. Sur un ordinateur avec un seul processeur, habituelle-	15. Pour afficher à l'aide de printf("%d\n",tab[i]);	affichera:
ment les processus sont exécutés :	le contenu d'un tableau de 5 entiers initialisé au	$\Box \ 4\ 3\ 2\ 1\ 0$
☐ chacun son tour, après que le processus précédent a terminé	préalable, on utilise plutôt :	$\square 4 3 2 1$
□ tour à tour, un petit peu à chaque fois	☐ for(i=1;i<5;i=i+1)	$\Box \ 0\ 1\ 2\ 3\ 4$
□ tous ensemble	☐ for(i=0;i<=5;i=i+1)	□ 1 2 3 4
□ en parallèle, chacun dans un registre	☐ for(i=1;i<=5;i=i+1)	
12. Laquelle de ces écritures correspond à la déclaration	☐ for(i=0;i<5;i=i+1)	19. Quel est le problème d'un programme comportant les lignes suivantes?
d'une variable de type caractère en langage C?	16. Vous avez déclaré préalablement un ensemble de fonc-	
☐ char 'c';	tions utilisées par votre programme principal. L'ordre	while (1)
\square int char;	dans lequel vous devez maintenant définir ces fonctions est l'ordre :	<pre>printf("coucou\n");</pre>
☐ char "c";		}
□ char c;	\square alphabétique	□ il n'affiche rien
13. Soient deux variables entières x et y initialisées à 4 et	\square un ordre quelconque	
5 respectivement. L'affichage x=4 et y=5 est obtenu avec la commande :	☐ dans lequel vous avez déclaré ces fonction	☐ il comporte une boucle infinie
avec la commande : □ printf("x=%d et y=%d\n,x,y");	\Box dans lequel ces fonctions sont appelées dans le	\Box il risque d'afficher bonjour à la place de coucou
☐ printf("x=%x et y=%y\n");	main	\square il ne compile pas
☐ printf("x=%d et y=%d\n",x,y);	17. Si carre est une fonction prenant en entrée un en-	20. La virtualisation de la mémoire permet notamment de
☐ printf("x=%d et y=%d\n",x y);	tier et renvoyant le carré de cet entier, et que n est	stocker des portions inactives de la mémoire de travail
14. Le code suivant :	une variable entière définie et initialisée, il est correct d'écrire :	sur le disque dur. Mais on perd :
int age = 20;	☐ int n = carre();	\square les fichiers du disque
if (age < 18)		□ en temps d'accès
<pre>{ printf("Mineur\n");</pre>	□ n = carre(n);	des processus
<pre>printi(mineur(n); }</pre>	\square int carre(2);	•
<pre>printf("Majeur\n");</pre>	\square n = carre(int n);	\Box certaines données de la mémoire de travail

т		-1
	acence	

Prénom:	Nom:
N° etu :	

Barème : 1 points par réponse juste (unique) ; -0,5 points par réponse fausse. Durée : 20 minutes. 1. Le type des réels en C est : \square int □ char ☐ double □ real 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 : \square 3 \Box 4 \square 0 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: □ A □ cccccc \square i ☐ ABCDEF 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

```
□ dans lequel vous avez déclaré ces fonction
     □ alphabétique
     □ dans lequel ces fonctions sont appelées dans le
       main
5. Le code suivant :
    int age = 15;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
     ☐ Mineur
    □ Mineur
       Majeur
    □ Majeur
    \square rien
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
    \square ses blocs
    □ ses chants
    □ ses cases
    \square ses champs
7. 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
```

```
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");
     \square printf("x=%d et y=%d\n",x y);
     \square printf("x=%d et y=%d\n",x,y);
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. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
    printf("Majeur\n");
   affichera:
     □ Mineur
     □ Majeur
     ☐ Mineur
        Majeur
     □ rien
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
      ☐ le programme affiche ****
     \square la variable x vaut -\frac{1}{2}
      □ la variable x vaut 16
      \square le programme affiche x
```

```
12. 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
                                                                      }
                                                                  }
13. 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
     □ 1 2 3 1 2
14. L'écriture 101 en binaire correspond au nombre natu-
   rel:
      \square 101
     \Box 5
     \square 3
     \Box 4
```

```
15. Pour l'extrait de programme suivant :
                                                                \Box f(a,b)=8, a=8, b=5
                                                               \Box f(a,b)=8, a=3, b=5
    int i = 0;
    int j = 0;
                                                          17. Le langage C est un langage
    for (i = 0; i < 2; i = i + 1)
                                                                □ composé
         for (j = 0; j < 3; j = j + 1)
                                                                □ interprété
                                                               □ compilé
             printf("%d ", j);
                                                                □ lu, écrit, parlé
                                                          18. Pour déclarer une procédure afficher_menu sans ar-
                                                              gument et qui ne renvoie rien on utilise :
   qu'est ce qui sera affiché?
     \Box 0 1 2 3 0 1 2
                                                                ☐ int afficher_menu();
     \Box 0 0 1 1 2 2 3
                                                               ☐ double afficher_menu();
     \Box 0 1 2 0 1 2 3
                                                               □ void afficher_menu();
     \Box 0 1 2 0 1 2
                                                                ☐ char afficher_menu(printf("menu"));
16. Soit le programme principal suivant :
                                                                ☐ int afficher_menu(int char);
   int main()
                                                          19. Lorsqu'un programme utilise printf ou scanf il faut
                                                              qu'il contienne l'instruction préprocesseur :
    int a = 3:
                                                                ☐ #include <studlib.h>
    int b = 5;
                                                                ☐ #appart <stdlib.h>
    printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
    return EXIT_SUCCESS;
                                                                ☐ #include <studio.h>
                                                                ☐ #include <stdio.h>
   appelant la fonction f ainsi définie :
                                                          20. Si racine est une fonction prenant en entrée un réel
   int f(int a, int b)
                                                              et renvoyant la racine carrée de cet réel, et que x est
                                                              une variable réelle définie et initialisée, il est incorrect
      a = a + b;
                                                              d'écrire :
      return a;
                                                               \square x = racine(x * x) - racine(x);
                                                               \square x - 1 = racine(x);
   L'affichage dans le main est le suivant :
                                                               \square x = racine(2/3);
     \Box f(a,b)=13, a=8, b=5
                                                               \square x = racine(racine(x)*racine(x));
     \Box f(3,5)=8, a=3, b=5
```

Prénom :	Nom:
N° etu:	
v cu.	

Licence 1 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 ☐ double \square real \square int 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 :

 \square 0 \Box 4 \square 3

3. Sur unix (ou linux), la commande mkdir permet de :

□ ouvrir un fichier texte

□ créer un répertoire

 \Box créer un fichier texte

□ 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, y); \square int pgcd(int y, int x); \square void pgcd(int x, int y); \square int pgcd(int x, int x);

5. 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

6. 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 \Box 1

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 || B)

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

□ A && B

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

8. 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 ****

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. 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

11. 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 0 1 2 3

 \square 4 3 2 1

12. Afin de représenter la taille d'un tableau, définir une constante symbolique N valant 3.

□ #define N 3

☐ #define taille = 3

☐ #define taille = N

 \square #define N = 3

13. 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
	\Box dans lequel vous avez déclaré ces fonction
	□ alphabétique
	\square un ordre quel conque
14.	L'écriture $\underline{111}$ en binaire correspond au nombre naturel :
	□ 111
	□ 7
	\square 3
	□ 8
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 :
	\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);
16.	Soit un programme contenant les lignes suivantes :
	<pre>int i = 0; int j = 0; for (i = 0; i < 2; i = i + 1) {</pre>

```
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 1 2 1 2 3
     \Box 0 0 0 1 1 1
     \square 0 1 0 1 0 1 0 1
17. Dans la commande gcc, l'option -Wall signifie :
     \square qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
     \square que l'on veut voir tous les avertissements
     \square qu'on veut changer alétoirement de fond d'écran
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
□ 16
19. Vous utilisez une boucle while quand :
$\hfill \square$ l'incrément de la variable de boucle n'est pas 1
□ vous avez déjà fait un for dans le même programme principal
\Box vous ne connaissez pas le nombre d'itérations d la boucle à l'avance
\square vous n'avez pas déclaré de fonction
20. Quel est le problème d'un programme comportant le lignes suivantes?
while (1)
{
<pre>printf("coucou\n"); }</pre>
\Box il risque d'afficher bonjour à la place de coucou
$\hfill\Box$ il comporte une boucle infinie
\square il ne compile pas
\square il n'affiche rien

1. Si le code :

Éléments d'informatique – contrôle continue

Prénom :	Nom:	
N° etu :		

```
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};
    \Box int struct toto_s = {3, -1e10};
    \square toto_s n, x;
    □ struct toto_s toto;
2. 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)
3. 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);
4. Sur unix (ou linux), la commande mkdir permet de :
     □ créer un répertoire
    \square changer de répertoire courant
     □ ouvrir un fichier texte
     □ créer un fichier texte
```

```
5. 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
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=3, b=5
    \Box f(3,5)=8, a=3, b=5
    \Box f(a,b)=8, a=8, b=5
7. 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:
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 0
     \Box 4
     \square 3
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. 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
     \Box 4
      \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 y, int x);
     \square int pgcd(int x, int x);
      \square int pgcd(int x, y);
     \square void pgcd(int x, int y);
```

<pre>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?</pre>	☐ jamais nulle ☐ à laquelle une valeur vient d'être affectée ☐ qui est vraie ou fausse 17. Si pgcd est une fonction prenant en entrée deux entiers et renvoyant un entier, il est correct d'écrire : ☐ n = pgcd(int, n, int, g); ☐ n = pgcd(int, n, int, g);	18. 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=%x et y=%y\n"); □ printf("x=%d et y=%d\n", x,y); 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(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); 20. 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 : □ 8 □ 0
 14. Un programme en langage C doit comporter une et un seule définition de la fonction : □ main 	$\Box n = \operatorname{ngcd}(\operatorname{int} n \operatorname{int} a).$	□ 0 □ 4 □ 16

т		-1
	100000	- 1
	acence	

Prénom:	Nom:	
N° etu :		

```
1. Le code suivant :
    int age = 15;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
         printf("Majeur\n");
    }
   affichera:
     □ Mineur
        Majeur
     □ rien
     □ Mineur
     □ Majeur
2. Pour déclarer une variable qui sera utilisée comme va-
   riable de boucle on peut utiliser l'instruction
     \square int loop n;
     □ loop i;
     \square int %d;
     \square int k;
3. 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
```

```
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};
    □ struct toto_s toto;
    \Box int toto.n = 3;
    \Box toto_s struct z = {3, 0.5};
    \square toto_s n, x;
5. 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, y);
    \Box int pgcd(int x, int x);
6. Le langage C est un langage
    □ compilé
    □ lu, écrit, parlé
    □ composé
    □ interprété
7. Pour déclarer une fonction factorielle qui prend en
   argument un entier et renvoie sa factorielle on écrit :
    □ struct int factorielle(int n);
    \Box int factorielle(int x);
     ☐ int factorielle(double n);
    ☐ int factorielle();
8. 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);
     \square un débogueur
     □ scanf("%d", &n);
```

```
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 0
     \Box 7
10. 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);
     ☐ int saisie_utilisateur();
     □ 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 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
```

```
15. Afin de représenter la taille d'un tableau, définir une
12. Le code suivant :
                                                                                                                                La valeur de somme affichée est :
                                                                  constante symbolique N valant 3.
     int i;
                                                                                                                                  \Box 6
     for (i = 4; i \ge 0; i = i - 1)
                                                                    \square #define N = 3
                                                                                                                                  \square 16
                                                                    \square #define taille = N
          printf("%d ", i);
                                                                                                                                  \square 20
                                                                    ☐ #define taille = 3
                                                                                                                                  \square 3
     printf("\n");
                                                                    □ #define N 3
    affichera:
                                                                                                                            19. Si racine est une fonction prenant en entrée un réel
                                                              16. Si cette erreur apparaît à la compilation :
      \square 4 3 2 1
                                                                                                                                et renvoyant la racine carrée de cet réel, et que x est
                                                                  Undefined symbols : "_prinft" ou
                                                                                                                                une variable réelle définie et initialisée, il est incorrect
      \Box \ 4\ 3\ 2\ 1\ 0
                                                                  référence indéfinie vers « prinft » que doit-
                                                                  on chercher dans le programme?
                                                                                                                                d'écrire :
      \Box 01234
      \Box 1 2 3 4
                                                                    □ une directive préprocesseur #include manquante
                                                                                                                                  \square x = racine(racine(x)*racine(x));
13. Soit la fonction f définie par :
                                                                    □ un caractère interdit en C
                                                                                                                                  \square x = racine(x * x) - racine(x);
    int f(int a)
                                                                    □ une variable non déclarée
                                                                                                                                  \square x = racine(2/3):
                                                                    \square une faute de frappe dans un appel de fonction
      printf("a = \n", %d);
                                                                                                                                  \square x - 1 = racine(x);
      if (a > 0)
                                                              17. Un enregistrement permet de grouper plusieurs valeurs
                                                                                                                            20. Au début de la fonction main() on place le code :
                                                                  dans:
        return 3;
                                                                    \square ses blocs
                                                                                                                                  char i;
      }
                                                                                                                                 for (i = 'A'; i \le 'F'; i = i + 1)
                                                                    \square ses chants
      return 4;
                                                                    \square ses champs
                                                                                                                                    printf("%c", i);
    Alors l'expression f(0) prendra la valeur :
                                                                    \square ses cases
      \square 3
                                                                                                                                 printf("\n");
                                                              18. Pour l'extrait de programme suivant :
      \Box 0
                                                                                                                                Alors l'affichage sera :
                                                                     int somme = 0;
      \Box 4
                                                                     int serie[4] = \{2, 4, 10, 4\};
14. Dans la commande gcc, l'option -Wall signifie :
                                                                                                                                  \square A
                                                                     for (i = 0; i < 4; i = i + 1)
      \square qu'on veut changer alétoirement de fond d'écran
                                                                                                                                  \Box i
      \square que l'on veut voir tous les avertissements
                                                                       somme = somme + serie[i];
                                                                                                                                  □ cccccc
      □ qu'il faut indenter le fichier source
                                                                                                                                  ☐ ABCDEF
      □ qu'il faut lancer un déboggueur
                                                                    printf("somme = %d".somme):
```

Éléments d'informatique – contrôle continue

Prénom :	Nom:
N° etu :	

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

r reponse fausse. Duree : 20	J minutes.
1. Si cet avertissement appa warning: implicit deci , que doit-on chercher dan	laration of function 'max
\square une directive préprod	cesseur #include manquante
\square un désaccord entre la d'une fonction	a déclaration et la définition
\Box une fonction déclarée	e mais non définie
\Box une fonction appelée	avant sa déclaration
2. Dans la commande gcc, l'	option -Wall signifie:
□ qu'il faut lancer un o	léboggueur
☐ qu'on veut changer a	alétoirement de fond d'écran
□ qu'il faut indenter le	fichier source
☐ que l'on veut voir to	us les avertissements
3. Si factorielle est une for entier et renvoyant un ent	onction prenant en entrée un tier, il est correct d'écrire :
☐ n = factorielle()	;
\square printf("%d", fact	orielle(n));
☐ int factorielle(i	nt 2);
☐ n = factorielle(p	, q);
4. Après exécution jusqu'à la	a ligne 15 du programme C :
10	
11 int main() { 12 int x = 5;	
13 Int x = 5,	
x = 3 * x + 1;	
15	
16	
17 }	
\Box le programme affiche	· ***
\Box la variable x vaut 16	
\Box la variable x vaut $-$	$\frac{1}{2}$
☐ le programme affiche	e x

```
5. 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;
6. Pour déclarer un tableau d'entiers de taille 5, on peut
   utiliser l'instruction
    □ char tableau[5]:
    \square int tab[] = 5;
    \square int toto[5];
    ☐ int toto[taille=5];
    ☐ int[] new tableau(5);
7. Le type des réels en C est :
    \square int
    □ double
    \square real
     □ char
8. Le code suivant :
    int i;
    for (i = 8; i > 0; i = i - 2)
        printf("%d ", i);
   printf("\n");
   affichera:
    \Box 02468
    \square 8 2
    \Box 8 6 4 2 0
    \Box 8 6 4 2
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 vous avez déclaré ces fonction
    □ un ordre quelconque
     □ dans lequel ces fonctions sont appelées dans le
```

main

```
10. Sur unix (ou linux), la commande mkdir permet de :
      □ ouvrir un fichier texte
      □ créer un fichier texte
      □ changer de répertoire courant
      □ créer un répertoire
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. L'ordonnancement par tourniquet permet :
      □ d'entretenir l'illusion que les processus tournent
        en parallèle
      \square 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
13. 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 <studio.h>
```

14. Au début de la fonction main() on place le code :	_ **** **** ****	\square 0 1 2 3 4 5 6 7
char i;	_ ** *** ****	□ 0 2 4 6 8
for (i = 'A'; i <= 'F'; i = i + 1)	_ **** *** ***	$\Box 0246$
{	16. Pour déclarer une fonction saisie_utilisateur qui	$\Box \ 0\ 1\ 2\ 3\ 4\ 5\ 6$
<pre>printf("%c", i); }</pre>	demande à l'utilisateur d'entrer un entier au clavier et renvoie cet entier on écrit :	19. Pour l'extrait de programme suivant :
<pre>printf("\n");</pre>	☐ void saisie_utilisateur(char c);	int produit = 0;
Alors l'affichage sera :	☐ saisie_utilisateur(scanf(%d));	<pre>int serie[4] = {2, 2, 2, 2}; for (i = 0; i < 4; i = i + 1)</pre>
□ ABCDEF	☐ void saisie_utilisateur(int n);	{
□i	☐ int saisie_utilisateur();	<pre>produit = produit * serie[i];</pre>
□ A □ cccccc	17. Pour déclarer une procédure afficher_date qui prend en argument un struct date_s et affiche le contenu	<pre>} printf("produit = %d", produit);</pre>
15. Pour l'extrait de programme suivant :	du struct, on écrit :	La valeur affichée est :
int i;	☐ int afficher_date(date_s d);	□ 8
int j;	\square void afficher_date(date_s d);	\square 4
for(i=4;i>0;i=i-1)	\square void afficher_date(struct date_s d);	□ 16
{ for(j=i;j<6;j=j+1)	☐ struct date_s afficher_date(struct date_s	d); 0
{	18. Le code suivant :	20. Vous utilisez une boucle while quand:
<pre>printf("*"); </pre>	int i;	□ vous n'avez pas déclaré de fonction
<pre>printf(" "); }</pre>	for (i = 0; i < 7; i = i + 2) { printf("%d ", i);	□ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
qu'est ce qui sera affiché?	<pre>print(//d</pre>	□ vous avez déjà fait un for dans le même programme principal
** ** ** ** **	affichera:	$\hfill\Box$ l'incrément de la variable de boucle n'est pas 1

Éléments d'informatique – contrôle continue

 \square 8

Prénom:	Nom:
N° etu :	

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 : \square 7 \Box 0 \Box 5 2. 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 des entrées clavier □ l'édition de liens ☐ l'analyse harmonique □ l'analyse sémantique 3. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise: ☐ int afficher menu(int char): ☐ int afficher_menu(); ☐ char afficher_menu(printf("menu")); □ void afficher_menu(); ☐ double afficher menu(): 4. Si cette erreur apparaît à la compilation : erreur: conflicting types for 'max', que doiton chercher dans le programme? □ 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

```
5. Si x est une variable réelle (de type double) alors
   x = 3/2 lui affecte la valeur :
     \square 1.5
     \Box 0
     \square 0.5
     \Box 1
6. 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 ****
     \Box la variable x vaut -\frac{1}{2}
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 15
     \square 10
     \Box 6
     \Box 0
8. L'écriture 111 en binaire correspond au nombre natu-
   rel:
     \square 111
     \square 3
     \square 7
```

```
9. Une variable booléenne est un variable :
      □ NaN (not a number, qui n'est pas un nombre)
      □ réelle positive
      \square qui est vraie ou fausse
      □ à laquelle une valeur vient d'être affectée
      \square jamais nulle
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:
     □ A
      \square B
      \Box C
      □b
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():
     ☐ int factorielle(int 2);
      \square n = factorielle(p, q);
12. 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
      \square il comporte une boucle infinie
```

13. La virtualisation de la mémoire permet notamment de	\square 3	19. Le code suivant :
stocker des portions inactives de la mémoire de travail sur le disque dur. Mais on perd :		int i;
□ certaines données de la mémoire de travail	\Box 4	for (i = 0; i < 7; i = i + 2) {
□ les fichiers du disque	16. Laquelle des analyses suivantes ne fait pas partie des étapes de la compilation :	<pre>printf("%d ", i); }</pre>
☐ des processus ☐ en temps d'accès	\square analyse harmonique	<pre>printf("\n");</pre>
14. Pour déclarer un tableau d'entiers de taille 5, on peut	$\hfill\Box$ analyse sémantique	affichera : $\square \ 0\ 1\ 2\ 3\ 4\ 5\ 6$
utiliser l'instruction	\square analyse lexicale	$\square \ 0\ 2\ 4\ 6$
☐ int[] new tableau(5);	\square analyse syntaxique	$\Box \ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7$
☐ int tab[] = 5;	17. Un programme en langage C doit comporter une et une	$\Box \ 0\ 2\ 4\ 6\ 8$
☐ int toto[5];	seule définition de la fonction :	20. Le code suivant :
□ char tableau[5];	\Box include	int age = 20;
☐ int toto[taille=5];	□ init	if (age < 18)
15. Soit la fonction f définie par :	□ begin	<pre>{ printf("Mineur\n");</pre>
<pre>int f(int a) {</pre>	\Box main	}
<pre>printf("a = \n", %d); if (a > 0)</pre>	18. Laquelle de ces écritures correspond à la déclaration d'une variable de type caractère en langage C?	<pre>printf("Majeur\n"); affichera:</pre>
{ return 3;	□ char c;	☐ rien ☐ Mineur
} return 4;	□ char 'c';	□ mineur Majeur
}	\square int char;	\square Mineur
Alors l'expression f(0) prendra la valeur :	□ char "c";	\square Majeur

т		-1
	100000	- 1
	acence	

Prénom:	Nom:
N° etu :	

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 : $\square \neq$ □ <> \Box ! □ != 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 16 \Box 0 \Box 4 3. L'écriture 111 en binaire correspond au nombre naturel: \square 8 \Box 7 □ 111 \square 3 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=0;i<=5;i=i+1) \square for(i=1;i<=5;i=i+1)

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

```
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. Le code suivant :
    int age = 15;
    if (age < 18)
    {
        printf("Mineur\n");
    }
    else
    {
        printf("Majeur\n");
    }
   affichera:
    □ Majeur
    □ Mineur
     ☐ Mineur
       Majeur
    \square rien
7. Le langage C est un langage
    □ compilé
    □ interprété
    □ lu, écrit, parlé
    □ composé
8. Vous utilisez une boucle while quand:
    \square 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
```

9.	Un bit est:
	\Box la longueur d'un mot mémoire
	\Box l'instruction qui met fin à un programme
	\square un chiffre binaire (0 ou 1)
	\Box un battement d'horloge processeur
10.	Lorsqu'un programme utilise printf ou scanf il faut qu'il contienne l'instruction préprocesseur :
	☐ #include <stdio.h></stdio.h>
	☐ #appart <stdlib.h></stdlib.h>
	☐ #include <studio.h></studio.h>
	☐ #include <studlib.h></studlib.h>
11.	Après exécution jusqu'à la ligne 15 du programme C
10	o int main() {
1	•
1:	9
1: 1	
1	
10	
1'	7 }
	\square la variable y vaut 5
	\Box le programme affiche "Faux"
	\Box la variable x vaut 0
	\Box la variable x vaut 5 et la variable y vaut 0
12.	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);
	☐ int carre(2);
	☐ int n = carre();
13.	L'écriture <u>101</u> en binaire correspond au nombre natu-
	rel:
	\Box 4
	□ 101
	\square 3
	\square 5

14. Un enregistrement permet de grouper plusieurs valeur	rs \Box 6
dans:	□ 20
\square ses champs	□ 16
□ ses chants □ ses cases	17. Avant de faire appe
□ ses blocs	☐ l'avoir définie
 15. Afin de représenter la taille d'un tableau, définir un constante symbolique N valant 3. □ #define taille = 3 □ #define N = 3 	ne □ l'avoir déclarée □ l'avoir déclarée □ avoir défini une de cette fonctio
\square #define N 3	18. Si a et b sont deux v
\square #define taille = N	struct toto_s
<pre>16. 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 : □ 3</pre>	<pre>f int n; double x; }; Alors pour tester l'e condition:</pre>

```
el à une fonction il est nécessaire
et définie
constante symbolique de la taille
variables de type :
égalité de a et de b on utilise la
\{n, x\}
&& (a.x == b.x)
```

```
19. 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
      \square une fonction déclarée mais non définie
      \square un désaccord entre la déclaration et la définition
         d'une fonction
     □ une directive préprocesseur #include manquante
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 7

 \Box 0

Éléments d'informatique – contrôle continue

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

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 : \square n = factorielle(); ☐ int factorielle(int 2); □ printf("%d", factorielle(n)); \square n = factorielle(p, q); 2. Un fichier source est: □ un document qui doit être protégé □ un document illisible pour les humains un fichier que l'ont doit citer dans les documents produits sur l'ordinateur □ un fichier texte qui sera traduit en instructions processeur \square un document de référence du système 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(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

```
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 double exposant(double x, int n);
     \square int exposant(double n, int x);
    \square void exposant(double x^n);
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 01234
    \square 1 2 3 4
    \Box \ 4\ 3\ 2\ 1\ 0
6. Sous unix (ou linux), pour créer un répertoire TP4
   dans le répertoire courant on peut utiliser la com-
   mande:
    ☐ kwrite TP4
    ☐ yppasswd
    ☐ mkdir TP4
    □ new TP4
7. 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);
```

□ struct int factorielle(int n);

☐ int factorielle();

```
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:
      □ A
      \Box i
      \Box ccccc
      ☐ ABCDEF
 9. L'écriture 101 en binaire correspond au nombre natu-
    rel:
      \Box 5
      \Box 4
      \square 3
      \square 101
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. Avant de faire appel à une fonction il est nécessaire
    de:
      □ l'avoir déclarée
      □ avoir défini une constante symbolique de la taille
         de cette fonction
      □ l'avoir définie
      □ l'avoir déclarée et définie
```

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu :	

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

I. I	reponse lausse. Duree: 20 minutes.
1.	Si cet avertissement apparaît à la compilation : warning: implicit declaration of function 'max, que doit-on chercher dans le programme?
	$\hfill \square$ une fonction déclarée mais non définie
	\Box une fonction appelée avant sa déclaration
	\Box un désaccord entre la déclaration et la définition d'une fonction
	\Box une directive préprocesseur $\verb"#include"$ man quante
2.	Le type des réels en C est :
	\square double
	□ char
	□ real
	□ int
3.	Vous utilisez une boucle while quand :
	\Box vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
	\square vous n'avez pas déclaré de fonction
	\Box vous avez déjà fait un for dans le même programme principal
	\Box l'incrément de la variable de boucle n'est pas 1
4.	Le code suivant :
	<pre>int i; for (i = 8; i > 0; i = i - 2) { printf("%d ", i); }</pre>
	<pre>printf("\n");</pre>
	affichera:
	□ 8 2
	$\Box \ 0\ 2\ 4\ 6\ 8$
	\square 8 6 4 2
	$\Box \ 8 \ 6 \ 4 \ 2 \ 0$

```
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) && (n<=b)
     □ a<=n<=b
     □ (n<=a) && (n<=b)
     \square (a<n) || (n>b)
6. Un enregistrement permet de grouper plusieurs valeurs
   dans:
     \square ses champs
     \square ses chants
     \square ses blocs
     \square ses cases
7. Afin de représenter la taille d'un tableau, définir une
   constante symbolique N valant 3.
     ☐ #define taille = N
     \square #define taille = 3
     \square #define N = 3
     □ #define N 3
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
      \square 0
 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 6
      \Box 0
      \square 42
      \Box 1
10. 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);
      ☐ printf("Valeur de n ? %g\n", n);
      □ scanf("%d", &n);
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?
      ☐ (A == TRUE) && (B == TRUE)
     □ A && B
      \square (!A || B)
      \square !(!A || B) == (A && !B)
12. Le code suivant :
     int age = 15;
     if (age < 18)
          printf("Mineur\n");
```

```
}
     else
     {
         printf("Majeur\n");
     }
   affichera:\\
     \square Mineur
         Majeur
      \square rien
      \square Mineur
      \square Majeur
13. Le code suivant :
     int age = 18;
     if (age < 18)
     {
         printf("Mineur\n");
     else
     {
         printf("Majeur\n");
     }
    affichera:
      \square Mineur
      \square rien
      \square Mineur
         Majeur
      \square Majeur
```

14. Au début de la fonction main() on place le code :	□ transporter les processus du tourniquet au pro-
char i;	cesseur □ Arriver à l'heure en cours
for (i = 'A'; i <= 'F'; i = i + 1) {	
<pre>printf("%c", i);</pre>	17. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :
<pre>} printf("\n");</pre>	\square double afficher_menu();
Alors l'affichage sera :	\square char afficher_menu(printf("menu"));
Alors i allichage sera . □ A	\square int afficher_menu(int char);
□ ABCDEF	\square void afficher_menu();
□ i	\square int afficher_menu();
□ сссссс	18. Si factorielle est une fonction prenant en entrée un entier et renvoyant un entier, il est correct d'écrire :
15. Soit la fonction g définie par :	\square int factorielle(int 2);
int g(int a)	\square n = factorielle(p, q);
{ printf("a = \n", %d);	☐ n = factorielle();
if (1 > 0)	\Box printf("%d", factorielle(n));
{ return 5;	19. Un programme en langage C doit comporter une et une seule définition de la fonction :
} return 7;	\square main
}	\square include
Alors l'expression g(0) prendra la valeur :	□ begin
	\Box init
□ 7	20. Lorsqu'un programme utilise printf ou scanf il faut
\Box 0	qu'il contienne l'instruction préprocesseur :
16. Le bus système sert à :	\square #include <studio.h></studio.h>
☐ Transférer des données et intructions entre pro-	□ #appart <stdlib.h></stdlib.h>
cesseur et mémoire	\square #include <stdio.h></stdio.h>
\Box Écrire des données sur le dique dur	\square #include <studlib.h></studlib.h>

Éléments d'informatique – contrôle continue

Prénom:	Nom:
N° etu:	

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 :
 - \Box 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
- 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 7
- \Box 0
- 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:
 - \Box dans lequel vous avez déclaré ces fonction
 - □ alphabétique
 - \square un ordre quelconque
 - \Box dans lequel ces fonctions sont appelées dans le main

- 4. Vous utilisez une boucle while quand :
 - □ vous avez déjà fait un for dans le même programme principal
 - \Box l'incrément de la variable de boucle n'est pas 1
 - \square vous n'avez pas déclaré de fonction
 - □ vous ne connaissez pas le nombre d'itérations de la boucle à l'avance
- 5. 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 $\# {\tt include}$ man quante
 - \Box un désaccord entre la déclaration et la définition d'une fonction
 - \square une fonction déclarée mais non définie
- 6. 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
 - \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
- 7. Un registre du processeur est :
 - ☐ un composant qui contient la liste des fichiers du système
 - $\square\,$ une unité de calcul spécialisée de l'ordinateur
 - \Box une gamme de fréquence de fonctionnement du processeur
 - ☐ une case mémoire interne au processeur qui sera manipulée directement lors des calculs
- 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 processeur
 - ☐ Transférer des données et intructions entre processeur et mémoire

- 9. 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 int carre(2);
 - \square n = carre(n);
 - \square n = carre(int n);
- 10. Le code suivant :

```
int i;
for (i = 0; i < 7; i = i + 2)
{
    printf("%d ", i);
}
printf("\n");</pre>
```

- affichera:
- \Box 0 1 2 3 4 5 6
- \Box 0 1 2 3 4 5 6 7
- \Box 0 2 4 6 8
- $\Box \ 0\ 2\ 4\ 6$
- 11. 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(date_s d);
 - ☐ int afficher_date(date_s d);
 - $\hfill\Box$ void afficher_date(struct date_s d);
 - $\hfill\Box$ struct date_s afficher_date(struct date_s d);
- 12. Sur unix (ou linux), la commande mkdir permet de :
 - □ ouvrir un fichier texte
 - □ créer un fichier texte
 - □ changer de répertoire courant
 - □ créer un répertoire

13. Quels calculs peut-on programmer en programmation structurée?	16. Laqı d'un
□ il y a des calculs programmables en programmation structurée qui ne sont pas programmables en langage machine	
\Box certains programmes sont de vrais plats de spaghetti	
□ il y a des calculs programmables en langage ma- chine et qui ne sont pas programmables en pro- grammation structurée	17. Le c int for
□ en programmation structurée on peut programmer tous les calculs programmables en langage machine	} }
14. Pour déclarer une procédure afficher_menu sans argument et qui ne renvoie rien on utilise :	pri affici
☐ int afficher_menu(int char);	
☐ double afficher_menu();	
☐ char afficher_menu(printf("menu"));	
□ void afficher_menu();	
☐ int afficher_menu();	18. Si le
15. Si n est une variable entière pour demander sa valeur à l'utilisateur, on utilise plutôt :	stri {
☐ printf("Valeur de n ? %d\n", n);	ir
☐ scanf("%d", &n);	do };
\square un débogueur	préc
☐ printf("Valeur de n ? %g\n", n);	débu

```
uelle de ces écritures correspond à la déclaration
ne variable de type caractère en langage C?
char "c";
char 'c';
 int char;
char c;
code suivant :
ti;
r (i = 0; i < 5; i = i + 1)
 printf("%d ", i);
intf("\n");
hera:
4\ 3\ 2\ 1\ 0
0\ 1\ 2\ 3\ 4
0\ 1\ 2\ 3
4\ 3\ 2\ 1
code:
uct toto_s
nt n;
ouble x;
cède la fonction main(), alors on peut écrire en
ut de main() :
```

```
\Box int struct toto_s = {3, -1e10};
     \Box toto_s struct z = {3, 0.5};
     □ struct toto_s toto;
      \square toto_s n, x;
      \square int toto.n = 3;
19. 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);
      ☐ int factorielle(int 2);
      ☐ printf("%d", factorielle(n));
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, x\} == b\{n, x\}
      \Box a = b
      \square (a.n == b.n) && (a.x == b.x)
```

Éléments d'informatique – contrôle continue

Prénom : Nom : N° etu :

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

2. Le bus système sert à :

☐ Transférer des données et intructions entre processeur et mémoire

☐ Arriver à l'heure en cours

☐ Écrire des données sur le dique dur

 \Box transporter les processus du tourniquet au processeur

3. Un programme en langage C doit comporter une et une seule définition de la fonction :

```
□ main
□ init
```

 \square include

 \square begin

 \Box j = 5

 \Box j = 0

4. Soit un programme contenant les lignes suivantes :

```
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 :
     \Box 0
     \Box 5
     \Box 1
     \Box 4
6. Dans la commande gcc, l'option -Wall signifie :
     □ qu'il faut lancer un déboggueur
     □ qu'il faut indenter le fichier source
     ☐ que l'on veut voir tous les avertissements
     □ qu'on veut changer alétoirement de fond d'écran
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
     □ il comporte une boucle infinie
     \square il risque d'afficher bonjour à la place de coucou
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 7
      \Box 5
 9. Le code suivant :
     int age = 20;
     if (age < 18)
          printf("Mineur\n");
     }
     else
     {
          printf("Majeur\n");
     }
    affichera:
      \square rien
      □ Majeur
      □ Mineur
        Majeur
      ☐ Mineur
10. Quel est l'opérateur de différence en C:
      \square \neq
      □ !=
      □ <>
```

```
18. Vous avez déclaré préalablement un ensemble de fonc-
11. Soit le programme principal suivant :
                                                            14. Si a et b sont deux variables de type :
                                                                                                                            tions utilisées par votre programme principal. L'ordre
    int main()
                                                                struct toto_s
                                                                                                                            dans lequel vous devez maintenant définir ces fonctions
                                                                                                                            est l'ordre:
     int a = 3;
                                                                  int n;
     int b = 5;
                                                                  double x;
                                                                                                                              □ dans lequel ces fonctions sont appelées dans le
     printf("f(a,b)=%d, a=%d, b=%d\n",f(a,b),a,b);
     return EXIT_SUCCESS;
                                                                Alors pour tester l'égalité de a et de b on utilise la
                                                                                                                              □ dans lequel vous avez déclaré ces fonction
                                                                condition:
    appelant la fonction f ainsi définie :
                                                                 □ a == b
                                                                                                                              \square un ordre quelconque
    int f(int a, int b)
                                                                  \Box a = b
                                                                                                                              □ alphabétique
                                                                  \square (a.n == b.n) && (a.x == b.x)
      a = a + b;
                                                                 \square a\{n, x\} == b\{n, x\}
                                                                                                                        19. Le code suivant :
      return a;
                                                            15. Quels calculs peut-on programmer en programmation
                                                                                                                             int age = 18;
                                                                structurée?
    L'affichage dans le main est le suivant :
                                                                                                                             if (age < 18)
      \Box f(a,b)=13, a=8, b=5
                                                                  \square il y a des calculs programmables en programma-
                                                                    tion structurée qui ne sont pas programmables en
      \Box f(a,b)=8, a=8, b=5
                                                                                                                                  printf("Mineur\n");
                                                                     langage machine
      \Box f(a,b)=8, a=3, b=5
                                                                  □ certains programmes sont de vrais plats de spa-
      \Box f(3,5)=8, a=3, b=5
                                                                                                                             else
                                                                     ghetti
12. Lorsqu'un programme utilise printf ou scanf il faut
                                                                                                                             {
                                                                  \square il y a des calculs programmables en langage ma-
    qu'il contienne l'instruction préprocesseur :
                                                                                                                                  printf("Majeur\n");
      ☐ #include <studio.h>
                                                                     chine et qui ne sont pas programmables en pro-
                                                                                                                             }
                                                                     grammation structurée
      ☐ #include <stdio.h>
                                                                  □ en programmation structurée on peut program-
      ☐ #appart <stdlib.h>
                                                                                                                            affichera:
                                                                    mer tous les calculs programmables en langage
      ☐ #include <studlib.h>
                                                                     machine
13. On souhaite faire une boucle de contrôle de saisie : tant
                                                                                                                              □ rien
                                                            16. Pour déclarer une procédure afficher_date qui prend
    que l'entier n n'appartient pas à l'intervalle [a..b], on
                                                                                                                              ☐ Mineur
                                                                en argument un struct date_s et affiche le contenu
    recommence la saisie de n. Soit le programme suivant :
                                                                du struct, on écrit :
     int a = 0;
                                                                                                                              □ Majeur
                                                                  □ void afficher_date(date_s d);
     int b = 20;
                                                                                                                              ☐ Mineur
     int n;
                                                                  □ void afficher_date(struct date_s d);
                                                                                                                                Majeur
     scanf("%d", &n);
                                                                  ☐ int afficher_date(date_s d);
     while(cond)
                                                                 \square struct date_s afficher_date(struct date_s |d)20. L'écriture 111 en binaire correspond au nombre natu-
                                                            17. Avant de faire appel à une fonction il est nécessaire
                                                                                                                            rel:
       scanf("%d", &n);
                                                                de:
                                                                                                                              \square 3
                                                                  □ avoir défini une constante symbolique de la taille
    Quelle est la condition cond:
                                                                     de cette fonction
                                                                                                                             \square 8
      □ a<=n<=b
                                                                  ☐ l'avoir définie
      \square (a<=n) && (n<=b)
                                                                                                                              \square 7
                                                                  □ l'avoir déclarée
      \square (n<=a) && (n<=b)
                                                                                                                              □ 111
                                                                  □ l'avoir déclarée et définie
      \square (a<n) || (n>b)
```