

NOM :
Prénom :



Examen de langage C

11 Janvier 2016

Durée : 1h30

Aucun document autorisé. Calculatrice interdite.

L'examen est constitué d'une partie QCM (Questions à Choix Multiples) et d'un exercice à rendre sur feuille à part. Pour répondre au QCM, vous devez remplir la feuille de réponse en p.7 et rendre l'ensemble de l'énoncé. Il y a 0,5 point par bonne réponse, 0 point s'il n'y a pas de réponse et -0,25 point par réponse fausse.

Question 1

Qu'affiche l'instruction suivante :

```
printf("j'aime le C %d fois", 30/20) ;
```

- 1) j'aime le C 0 fois
- 2) j'aime le C 1 fois
- 3) j'aime le C 1.5 fois
- 4) je n'aime pas le C

Question 2

Qu'affichent les instructions suivantes :

```
int x=9 ;  
int y=x+10 ;  
printf("%d : %d : %d", x, y, y) ;
```

- 1) 9:19 :
- 2) 9:19:19
- 3) 9:19:%d
- 4) %d:%d:%d

Question 3

Qu'affichent les instructions suivantes :

```
char a ;  
a=65 ;  
printf("%d %c", a, a) ;
```

- 1) A A
- 2) 65 A
- 3) A 65
- 4) 65 65

Question 4

Qu'affichent les instructions suivantes :

```
int i=10 ;  
while (i>0) {  
    i=i-4 ;  
    printf("%d ", i) ;  
}
```

- 1) 1 0 6 2
- 2) 6 2 -2
- 3) 6 2
- 4) C'est une boucle infinie

Question 5

Le type « char » correspond à :

- 1) Un entier codé sur un unique octet
- 2) Un réel
- 3) Une chaîne de caractères
- 4) Aucune de ces trois réponses.

Question 6

Quelle est la valeur de S après exécution des instructions suivantes :

```
int i ;  
int S=0 ;  
for (i=1 ; i<6 ; i=i+2)  
    S=S+i ;
```

- 1) 6
- 2) 7
- 3) 8
- 4) 9

Question 7

Quelle est l'occupation mémoire de la constante symbolique définie par

```
#define A 10.5 ?
```

- 1) 0 octet
- 2) 1 octet
- 3) 2 octets
- 4) 4 octets

Question 8

Le type « sentence » correspond à

- 1) Un entier codé sur un unique octet
- 2) Un réel
- 3) Une chaîne de caractères
- 4) Aucune de ces trois réponses.

Question 9

Dans une structure if ...

- 1) Les parenthèses encadrant la condition logique sont obligatoires
- 2) Le mot clé « else » est obligatoire
- 3) La condition, énoncée juste après if, est suivie d'un point virgule.

Question 10

Si le nombre d'itérations est connu, il est conseillé d'utiliser :

- 1) while ...
- 2) do ... while
- 3) for ...

Question 11

L'instruction « switch » sert à éviter des instructions :

- 1) while ... imbriquées
- 2) do ... while imbriquées
- 3) if ... else ... imbriquées
- 4) for ... imbriquées

Question 12

Si le nombre d'itérations n'est a priori pas connu et que l'on désire passer au moins une fois dans la boucle, on utilise :

- 1) while ...
- 2) do ... while
- 3) for ...

Question 13

En programmation en langage C, quel signe utilise-t-on pour l'affectation ?

- 1) =
- 2) :=
- 3) ==

Question 14

On considère deux tableaux T1 et T2. Entourez les propositions correctes. Il est possible de copier le contenu de T2 dans T1 sans perdre d'information :

- 1) directement si T1 et T2 sont de même taille : on utilise l'instruction T1=T2
- 2) directement si la taille de T1 est supérieure à la taille de T2 : on utilise l'instruction T1=T2
- 3) directement si la taille de T2 est supérieure à la taille de T1 : on utilise l'instruction T1=T2
- 4) élément par élément à l'aide d'une boucle dès que la taille de T1 est supérieure ou égale à la taille de T2.

Question 15

Soit un tableau int T[10] ; pour accéder à la troisième case du tableau T, on utilise :

- 1) T[3]
- 2) T[2]
- 3) T{2}
- 4) T(3)

Question 16

On définit les variables de la façon suivante :

int i ;

```
int tab[10] ;
char c ;
int *pti ;
char *ptc ;

Cocher ce qui est juste (pour les lignes de code, ce qui compile sans warning) :
```

- 1) pti=&i ; *pti=12 ;
- 2) pti=&tab ; *pti=4 ;
- 3) ptc=&c ; *ptc='a' ;
- 4) tab[i] est équivalent à *(tab+i) ;
- 5) tab[i] est équivalent à &(tab+i) ;

Question 17

On considère la déclaration suivante : char *ptc ; ptc peut contenir :

- 1) des valeurs de variables de type caractère
- 2) des adresses de variables de type caractère

Question 18

On définit les constantes et les variables suivantes :

```
#define A 5
#define B 7
#define G 5.6
float c, d ;
int e=5, f=7 ;
```

Lesquelles de ces déclarations sont-elles justes :

- 1) int T1[10][B] ;
- 2) float T2[10][B] ;
- 3) int T3[A][B] ;
- 4) int T4[G][B] ;
- 5) char T5[1][c] ;
- 6) float T6[e][f] ;

Question 19

Lesquelles de ces déclarations de fonctions sont correctes si elles sont sensées calculer « x » élevé à la puissance « n » entière ?

- 1) float puissance (float x, int n) ;
- 2) puissance (float x, int n) ;
- 3) void puissance (float x, int n) ;
- 4) puissance() ;

Question 20

Il est conseillé d'utiliser des variables :

- 1) Locales
- 2) Globales

Question 21

Quelles sont les manières correctes pour passer le tableau d'entiers tab en argument ?

- 1) void fonction (int *tab, int dim)
- 2) void fonction (int tab[], int dim)
- 3) void fonction (int &tab, int dim)

Question 22

Entourez les réponses correctes. Dans un fichier d'entête .h, on met :

- 1) Les définitions des fonctions
- 2) Les entêtes ou prototypes des fonctions
- 3) Les définitions des structures
- 4) La fonction main
- 5) Toutes les variables du projet

Question 23

La fonction free de la bibliothèque stdlib.h est utilisée pour affecter la valeur zéro à un pointeur.

- 1) Vrai
- 2) Faux

Question 24

Que fait la fonction malloc(n) ?

- 1) Fournit l'adresse du premier octet d'un bloc en mémoire de n octets libres
- 2) Fournit l'adresse du premier octet d'un bloc en mémoire de n octets libres ou le pointeur NULL s'il n'y a pas assez de mémoire
- 3) Fournit un bloc en mémoire à partir de l'adresse n

Question 25

En C, on peut utiliser les variables sans pour autant avoir à les déclarer.

- 1) Vrai
- 2) Faux

Question 26

Solient les lignes d'instruction suivantes :

```
struct timbre {
    int prix ;
    int annee ;
    char origine[20] ;
    char image[20] ;
} ;
struct timbre COLLECTION[10] ;
```

Comment accède-t-on à l'année du 3^{ème} timbre de la collection ?

- 1) COLLECTION[2,2]
- 2) COLLECTION[2].annee
- 3) COLLECTION.annee[2]
- 4) COLLECTION.annee
- 5) (COLLECTION+2)->annee

Question 27

On considère l'entête (ou prototype) de fonction suivante :

```
void fiche(float *x, float *y, int i, char z, char c)
```

On considère les déclarations suivantes :

```
float a, c ;
```

```
int j ;
char b, h ;
```

Quels sont les appels de fonction corrects ?

- 1) fiche (a, c ; j ; b, h) ;
- 2) fiche (&a, &b, c, j, h) ;
- 3) fiche (&a, &c, 3, 'b', b) ;
- 4) fiche (&a, &c, j, b, h) ;
- 5) fiche (a ; j ; b ; h) ;

Question 28

Soit le pointeur de fichier défini par FILE *fp ; on veut ouvrir le fichier « C:/Mesdocuments/essai.txt » en mode lecture. Entourez les propositions correctes.

- 1) "C:/Mesdocuments/essai.txt"=fopen(fp, "r") ;
- 2) fp=fopen("C:/Mesdocuments/essai.txt", "r") ;
- 3) fp=fopen(C:/Mesdocuments/essai.txt, "r") ;
- 4) fopen(fp, "C:/Mesdocuments/essai.txt", "r") ;
- 5) le fichier «C:/Mesdocuments/essai.txt » doit exister
- 6) Si le fichier «C:/Mesdocuments/essai.txt » n'existe pas, il est créé.

Exercice à faire sur feuille à part

Faites une fonction qui sauvegarde dans un fichier au format ASCII un tableau de doubles avec le format suivant : « la composante i vaut tab[i] », suivi d'un passage à la ligne.

Faites une fonction qui lit un fichier écrit dans ce format.

Comment organiseriez-vous ce projet sous Code Blocks ? Précisez en particulier les fichiers sources, les fichiers d'entête, les inclusions de fichiers.