

Examen Programmation Structurée 2017- 2018 : Normale

Génie Informatique BTS I (GL I)

**Examineur : Nyabeye Doris-Khöler**

**Exercice 1 : Question à choix multiples (8 pts)**

Répondre sur votre copie d'examen en mentionnant juste le numéro de la question et les éléments de réponse.

Réponse juste (0.5 pt) – réponse fausse (-0.25 pt) – pas de réponse (0 pt). Question à réponse double (0.5 pt\*2).

1. Qu'affiche l'instruction suivante : `printf("j'aime le C %d fois", 30/20);`

- a. j'aime le C 0 fois
- b. j'aime le C 1 fois
- c. j'aime le C 1.5 fois
- d. je n'aime pas le C

2. 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 ;`

- a. 6
- b. 7
- c. 8
- d. 9

3. Qu'affichent les instructions suivantes :

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

- a. A A
- b. 65 A
- c. A 65
- d. 65 65

4. Qu'affichent les instructions suivantes :

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

- a. 1 0 6 2
- b. 6 2 -2
- c. 6 2
- d. C'est une boucle infinie

5. Dans une structure if ...
  - \* a. Les parenthèses encadrant la condition logique sont obligatoires
  - b. Le mot clé « else » est obligatoire
  - c. La condition, énoncée juste après if, est suivie d'un point-virgule.
6. Si le nombre d'itérations est connu, il est conseillé d'utiliser :
  - a. while ...
  - b. do ... while
  - \* c. for ...
7. L'instruction « switch » sert à éviter des instructions :
  - a. while ... imbriquées
  - b. do ... while imbriquées
  - \* c. if ... else ... imbriquées
  - d. for ... imbriquées
8. 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 :
  - \* a. while ...
  - b. do ... while
  - c. for ...
9. En programmation en langage C, quel signe utilise-t-on pour l'affectation ?
  - a. =
  - b. :=
  - c. ==
10. Lesquelles de ces déclarations de fonctions sont correctes si elles sont supposées calculer « x » élevé à la puissance « n » entière ?
  - \* a. float puissance (float x, int n) ;
  - b. puissance (float x, int n) ;
  - \* c. void puissance (float x, int n) ;
  - d. puissance()
11. Quelles sont les manières correctes pour passer le tableau d'entiers tab en argument ?
  - \* a. void fonction (int \*tab, int dim)
  - \* b. void fonction (int tab[ ], int dim)
  - c. void fonction (int &tab, int dim)
12. Soit un tableau int T[10] ; pour accéder à la troisième case du tableau T, on utilise :
  - a. T{2}
  - b. T(3)
  - \* c. T[3]
  - d. T[2]



13. Soient 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 3ème timbre de la collection ?

- a. COLLECTION[2,2]
  - b. COLLECTION[2].annee
  - c. COLLECTION.annee[2]
  - d. COLLECTION.annee
  - e. (COLLECTION+2)->annee
14. En C, on peut utiliser les variables sans pour autant avoir à les déclarer.
- a. Vrai
  - b. Faux

#### Exercice 4 : Tableaux, Structures et énumérations (12 pts)

1. Déclarer une énumération **jour** permettant de nommer les jours de la semaine en toutes lettres. (1 pt)
2. Créer un type **jdl** comme alias de cette énumération. (1 pt)
3. Créer une structure **periode** qui contient 4 champs : (1 pt)
  - J pour les jours (de type **jour**)
  - H pour les heures
  - M pour les minutes
  - S pour les secondes
4. Définir le type **Periode** permettant de désigner un alias pour cette structure. (1 pt)
5. Ecrire la fonction lirePeriode qui permet de lire 50 périodes et de les stocker dans un tableau.  
**Note Bien** : le nombre de seconde et le nombre de minutes entrés ne doivent pas dépasser 59 (2.5 pts)
6. Ecrire la fonction affichePeriode qui permet d'afficher les périodes du tableau sous la forme : **Jour - x h : y min : z sec.**  
**Exemple** : lundi - 8 h : 30 min : 50 sec. (2.5 pts)
7. Ecrire la fonction main permettant de tester les fonctions écrites. (3 pts)