

Examen Final

| | | | |
|--------------------|---|-------------------|-----------------|
| Matière | : Algorithmique et Programmation 1 | Classe | : TI1x |
| Enseignants | : J. Bokri, A. Elamraoui, O. Henia, Y. Landolsi | Durée | : 01h30 |
| Date | : 02/01/2020 | Documents | : Non autorisés |
| | | Nbre pages | : 3 |

QCM : (6 pts)

Q1/ 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

Q2/ 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

Q3/ 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

Q4/ Combien de fois passe-t-on dans la boucle suivante ?

```
int Cpt = 10 ;  
do {  
cpt++ ;  
printf("bonjour \n") ;  
} while ( Cpt < 10) ;
```

- A. 0
- B. 1
- C. 9
- D. 10

Q5/ Combien de fois passe-t-on dans la boucle suivante ?

```
for ( compteur = 2; compteur <  
9; compteur += 2 )
```

- A. 4
- B. 5
- C. 7
- D. 8

Q6/ Lequel de ses prototypes de fonction ne permet pas de faire passer un tableau ?

- A. void Mafonction (int
tableau[], int taille);
- B. void Mafonction (int
tableau, int taille);
- C. void Mafonction (int *
tableau, int taille);

Q7/ Soient les lignes d'instructions 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

Q8/ 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 ?

- A. fiche (a, c ; j ; b, h) ;
- B. fiche (&a, &b, c, j, h) ;
- C. fiche (&a, &c, 3, 'b', b) ;
- D. fiche (&a, &c, j, b, h) ;

Exercice 1 : (4 pts)

Soit la suite arithmétique suivante définie par :

$$S_0 = -2, S_1 = 1 \text{ et } S_n = 4 * S_{n-2} + (-1)^n * S_{n-1}$$

NB : L'exercice doit être traité en algorithmique.

1. Ecrire une fonction **réursive SuiteRec** qui prend en paramètre un entier N et retourne le **Nième** terme de cette suite. **(2pts)**
2. Ecrire une procédure **saisirEntier** qui permet de saisir un entier positif. **(1pt)**
3. Ecrire le **programme principal** qui demande à l'utilisateur de saisir un entier positif N et calcule et affiche le Nième terme de la suite (Utiliser les sous-programmes des questions précédentes). **(1pt)**

Problème : (10 pts)

On souhaite gérer les inscriptions à un congrès organisé dans un hôtel pour une journée et une nuit.

Les organisateurs proposent aux participants trois types d'inscriptions :

- Hébergement et petit déjeuner '**P**' avec un prix `PRIX_PDEJ=48D`
- Hébergement en demi-pension '**D**' avec un prix `PRIX_DP=54D`
- Hébergement et pension complète '**C**' avec un prix `PRIX_PC=60D`

Un participant peut être accompagné de son conjoint. Dans ce cas, le prix d'inscription sera doublé.

Un participant est caractérisé par :

- **nomPar** de type chaîne de caractères représentant le nom et prénom du participant,
- **typeIns** de type caractère représentant le type d'inscription (P, D, C),
- **conjoint** de type booléen indiquant si le participant est accompagné par son conjoint ou non,
- **montantPar** de type réel calculé selon le type d'inscription et la présence du conjoint.

NB : Le problème doit être traité en langage Python.

1. Déclarer la structure **Participant**. (1 pt)
2. Ecrire une procédure **lireParticipant** qui permet de saisir les informations d'un participant passé en paramètre et de calculer le **MontantPar** qu'il doit payer. (2 pts)
3. Ecrire une procédure **afficheParticipant** qui affiche les informations du participant passé en paramètre (1.5 pt)
4. Ecrire une procédure itérative **remplirTab** qui permet de remplir le tableau des participants en donnant la taille effective `n` en paramètre avec le tableau. (1pt)
5. Ecrire une procédure itérative **afficheTab** qui permet d'afficher toutes les informations des participants. (1pt)
6. Ecrire une fonction récursive **calculTotal** qui permet de calculer et de renvoyer le total des montants des tous les participants du tableau. (1.5 pt)
7. Ecrire un **programme principal** qui permet de (2pts)
 - a. Déclarer un tableau de 50 participants au maximum.
 - b. Demander à l'utilisateur le nombre de participants.
 - c. Remplir le tableau des participants
 - d. Afficher tous les participants
 - e. Afficher le montant total des participations.