

---

 PROGRAMMATION IMPÉRATIVE - LANGAGE C
 

---

## EXAMEN

 2 heures  
 sans documents
 

---

**N.B.** : - Les réponses aux questions doivent être argumentées et aussi concises que possible.

- Le barème est donné à titre indicatif.

**Question 1**

(2 points)

En quoi le langage C peut-il être considéré comme un langage de bas niveau ? En quoi peut-il être considéré comme un langage de haut niveau ?

**Question 2**

(3 points)

Écrivez la définition des variables ou des prototypes de fonctions suivants :

- (a) `p` est un pointeur sur un tableau de chaînes de caractères ;
- (b) `t` est un tableau unidimensionnel de 5 `int` dont les trois premiers sont initialisés avec les valeurs 1, 2 et 3 ;
- (c) `f` est une fonction prenant en paramètre un `int` et un `double` et renvoyant un pointeur sur un `int` ;
- (d) `t` est un tableau bidimensionnel de 3 lignes de 2 colonnes d'`ints`, dont la première colonne est initialisée à 0 et la deuxième à 1 ;
- (e) `f` est un pointeur sur une fonction prenant en paramètre un `double` et renvoyant un `int` ;
- (f) `p` est un pointeur sur un tableau de pointeurs de fonctions prenant en paramètre un `int` et ne renvoyant rien.

**Question 3**

(3 points)

Analysez le programme suivant et donnez la séquence des affichages produits.

```
int f1 (int i) { return (i + 1); }
int f2 (int i) { return (i ++); }
int f3 (int i) { printf ("%d\n", (i == 0)); return (i); }
int f4 (int i) { printf ("%d\n", (i = 0)); return (i); }
int f5 (int * i) { return (++ (*i)); }
int f6 (int * i) { (*i) ++; return ((*i) ++); }
int main () {
    int a, b;
    a = f1 (0); b = f2 (1); printf ("a = %d, b = %d\n", a, b);
    a = f3 (a); b = f4 (a); printf ("a = %d, b = %d\n", a, b);
    a = f5 (&a); b = f6 (&a); printf ("a = %d, b = %d\n", a, b);
}
```

**Question 4**

(4 points)

Répertorier toutes les erreurs se trouvant dans le programme ci-dessous. Indiquer brièvement la façon de les corriger.

```
1. #include <stdio.h>
2.
3. int main (void)
4. {
5.     char * chaine, mot_de_passe = "Mot2Pa's";
6.     scanf ("%s", chaine);
7.     if (chaine == mot_de_passe)
8.         printf ("Vous pouvez passer.\n");
9.     else
10.        printf ("C'est pas ça!\n");
11.        return (1);
12. }
```

**Question 5**

(8 points)

Lors d'une course de relais, une équipe est composée de 4 relayeurs. Le temps effectué par chaque participant est exprimé sous forme d'un nombre de minutes, de secondes et de centièmes de secondes. On demande d'écrire un programme qui permette de saisir les temps effectués par chaque relayeur d'une équipe et qui donne alors le temps total de l'équipe ainsi que le meilleur et le moins bon temps. Un temps (minutes, secondes et centièmes) est représenté par une structure à définir.

(5.1) (1 point)

Donnez la définition du type structuré **Chrono**.

(5.2) (2 points)

Écrivez une fonction **chronoAjoute()** qui permette d'additionner deux temps stockés dans de telles structures. Pour cela, définissez la façon dont les paramètres devront être passés à la fonction, et justifiez la.

(5.3) (2 points)

Écrivez une fonction **chronoCompare()** qui compare deux temps stockés dans de telles structures, et qui renvoie  $-1$  si le premier temps est inférieur au deuxième,  $+1$  s'il est supérieur, et  $0$  s'ils sont identiques.

(5.4) (3 points)

Utilisez ces deux fonctions pour calculer la somme des temps des relayeurs ainsi que le meilleur et le moins bon temps au fur et à mesure de la saisie de chacun de ces temps. Afficher la somme des temps, le meilleur et le moins bon temps à la fin de la saisie. Créez des fonctions annexes si nécessaire.