



## Éléments de programmation 2– LU1IN002

**Devoir à la maison - remplace le partiel du 16 mars 2020**

**Tous documents autorisés.**

*Les calculatrices, baladeurs et autres appareils électroniques sont interdits. Les téléphones mobiles doivent être éteints et rangés dans les sacs.*

Le barème sur 42 points (9 questions) n'a qu'une valeur indicative.

### Pointeurs

#### Question 1 (11 points)

On se donne les déclarations suivantes

```
int n;  
int* a;
```

Pour chacune des séquences d'instructions ci-dessous vous indiquerez si elle est correcte et le résultat de l'affichage, ou si elle est incorrecte et pourquoi elle est incorrecte.

1. `n = 42; a = &n; printf("%d_%d\n", n, *a);`
2. `n = 42; a = &n; printf("%d_%d\n", n, a);`
3. `n = 42; *a = n; printf("%d_%d\n", n, *a);`
4. `n = 42; *a = &n; printf("%d_%d\n", n, *a);`
5. `n = 42; a = 42; printf("%d_%d\n", n, *a);`
6. `a = &n; n = 42; printf("%d_%d\n", n, *a);`
7. `a = &n; *a = 42; printf("%d_%d\n", n, *a);`
8. `a = malloc(sizeof(int)); *a = 42; printf("%d_%d\n", n, *a);`
9. `a = malloc(sizeof(int)); n = 42; printf("%d_%d\n", n, *a);`
10. `a = malloc(sizeof(int)); n = 42; *a = n; printf("%d_%d\n", n, *a);`
11. `a = malloc(sizeof(int)); *a = 42; n = *a; printf("%d_%d\n", n, *a);`

Les anomalies peuvent être de plusieurs sortes : erreur de syntaxe, erreur de type, erreur à l'exécution ou affichage indéterminé.

**Question 2** (3 points)

```
1 #include<stdio.h>
2 void f(int a, int* b) {
3     *b = *b + a;
4 }
5 int main() {
6     int x = 6;
7     f(x, &x);
8     f(x, &x);
9     return 0;
10 }
```

Dessinez l'évolution de l'état de la pile lors de l'exécution du programme ci-dessus. Vous considèrerez les étapes suivantes :

1. la déclaration **int** x = 6; (ligne 1)
2. les passages de paramètre au 1er appel f(x, &x); (ligne 7)
3. la 1ère exécution de \*b = \*b + a; (ligne 3)
4. les passages de paramètre au 2nd appel f(x, &x); (ligne 8)
5. la 2nde exécution de \*b = \*b + a; (ligne 3)
6. l'état de la pile avant le **return** (ligne 9)

## Alternatives

**Question 3** (5 points)

Pour calculer le classement des joueurs de tennis, on calcule pour chacun des matchs gagnés un nombre de points selon les règles suivantes<sup>1</sup> :

- 0 point pour une victoire face à quelqu'un situé 4 échelons en dessous ou plus
- 15 points si 3 échelons en dessous
- 20 points si 2 échelons en dessous
- 30 points si 1 échelon en dessous
- 60 points si vous êtes d'échelon égal
- 90 points si 1 échelon au-dessus
- 120 points pour une victoire face à un adversaire situé 2 échelons au-dessus ou plus.

Définir la fonction de signature **int** points(**int** e1, **int** e2) où e1 est l'échelon du vainqueur et e2 l'échelon du perdant et qui donne le nombre de points attribués au vainqueur du match.

---

1. source : wearetennis.bnpparibas

## Tableaux

Dans cet exercice, les questions sont indépendantes même si chacune d'elles utilise la fonction définie dans la question précédente. Vous pouvez donc répondre à la question  $i + 1$  même si vous n'avez pas su répondre à la question  $i$ .

### Question 4 (4 points)

Définir une fonction `val_pos` qui prend en paramètre un entier `n`, un tableau d'entiers `t` et sa longueur `len` et qui donne 1 si `t` contient au moins `n` valeurs strictement positives, et 0 sinon. Vous veillerez à ne pas faire de calculs inutiles.

### Question 5 (4 points)

Définir la fonction `indice_min` qui prend en argument un tableau d'entiers non vide `t` et sa longueur `len` et qui renvoie l'indice d'une valeur minimale de `t`. Si cette valeur est présente plusieurs fois dans le tableau, la fonction peut donner l'indice de la première ou de la dernière occurrence de la valeur. Vous indiquerez laquelle des deux options vous avez choisie.

### Question 6 (2 points)

En utilisant `indice_min`, définir la fonction `replace_min_if_sup` qui prend en argument un entier `n`, un tableau d'entiers `t` non vide et sa longueur `len` et qui remplace une valeur minimale dans `t` par `n`, si `n` est supérieur à cette valeur.

### Question 7 (5 points)

En utilisant `replace_min_if_sup`, définir la fonction `t_best` qui prend en argument un entier `n`, un tableau d'entiers `t` et sa longueur `len` et qui renvoie un tableau contenant les `n` valeurs les plus grandes de `t`. On fait l'hypothèse que  $0 < n \leq len$ .

## Tableaux et adresses

### Question 8 (5 points)

Définir la fonction `echange` qui prend en argument deux tableaux d'entiers `t1` et `t2` et leur longueur commune `len` et qui échange les valeurs de `t1` et `t2` selon la règle suivante : si `t1[i] < t2[i]` alors échanger `t1[i]` et `t2[i]`.

Exemple :

si `t1` est le tableau { 7, 3, 9, 4, 5, 7 }

si `t2` est le tableau { 6, 4, 8, 3, 6, 5 }

alors, après exécution de `echange(t1, t2, 6)` ont aura que

`t1` est le tableau { 7, 4, 9, 4, 6, 7 }

`t2` est le tableau { 6, 3, 8, 3, 5, 5 }.

### Question 9 (3 points)

Soit le tableau d'entiers `t3 = { 7, 3, 9, 4, 5, 7, 6, 4, 8, 3, 6, 5 }`. Que vaut `t3` si l'on exécute `echange(t3, &t3[6], 6)` ? Expliquez votre réponse.