

# ECE Lyon - ING1 Informatique — Devoir Surveillé n°2

## Pointeurs, structures, tableaux statiques, chaînes de caractères et aléatoire

Nom : ..... Prénom : .....

### Partie 1 : Pointeurs (7 points)

#### Exercice 1 - Passage par adresse/valeur (1 point)

Donnez les différences entre le passage de paramètres par valeur et par adresse. Dans quels cas privilégie-t-on plutôt l'un à l'autre ?

#### Exercice 2 - Trace (3 points)

Soit le programme suivant :

```
1 void doSomething1(int* a, int b) {
2     *a += 5;
3     b += 5;
4 }
5 int doSomething2(int a) {
6     return a + 10;
7 }
8 int main(void) {
9     int a = 0,
10     b = 0;
11     int* pA = &a;
12     doSomething1(pA, b);
13     doSomething2(a);
14     return 0;
15 }
```

Faites la trace d'exécution de ce programme. Vous pouvez pour cela faire un tableau à 3 colonnes : adresse, valeur, variable (comme vu en cours). Vous laisserez la trace de toute valeur ayant été écrite en rayant proprement les lignes supprimées. Vous pourrez également ajouter des commentaires quand vous jugez cela utile (quand et pourquoi la ligne est supprimée...).

#### Exercice 3 - Pointeurs (3 points)

Soit le programme suivant :

```
1 int main(void) {
2     int a = 3;
3     int* pA = &a;
4     int** ppA = &pA;
5     return 0;
6 }
```

Nom : ..... Prénom : .....

— Suite Exercice 3 —

Indiquez à quoi correspondent `a`, `&a`, `pA`, `&pA`, `*pA`, `ppA`, `**pA` et quelles sont leurs valeurs juste avant la fin du programme (ligne 5). Vous pouvez vous aider d'un schéma.

## Partie 2 : Structures et tableaux (7 points)

### Exercice 4 - Structure ou tableau ? (1 point)

Selon vous, dans quel cas privilégie-t-on plutôt un tableau ou une structure ?

### Exercice 5 - Structure (3 points)

Créez une structure “**Pion**” qui contient une lettre en guise de nom, un numéro d'équipe et deux entiers représentant sa position sur un plateau à deux dimensions (`x` et `y`).

Déclarez un **Pion** ayant pour nom “**E**” (“**Éléphant**”), appartenant à l'équipe 1 et se trouvant en position (5,0). Ce pion ne peut se déplacer que verticalement (sur l'axe `y`).

Créez la fonction `deplacementVertical` prenant en paramètres un pion, la nouvelle coordonnée `x` et nouvelle coordonnée `y` du pion. Cette fonction devra retourner 1 si le déplacement du pion passé en tant que premier paramètre est bien vertical, 0 sinon.

### Exercice 6 - Tableau à deux dimensions (3 points)

Créez un programme créant un tableau à deux dimensions de taille MAX par MAX (avec MAX une constante de valeur 10).

Initialisez ce tableau en affectant à chacune de ses cases le résultat de la multiplication de ses deux indices (`ligne x colonne`). Affichez le résultat de la multiplication `3x4` en utilisant le tableau créé. Il suffit pour cela d'afficher le contenu de la bonne case du tableau.

## Partie 3 : Chaînes de caractères (3 points)

### Exercice 7 - Palindrome (3 points)

Écrire un programme qui vérifie si une chaîne de caractères introduite au clavier est un palindrome. Vous pouvez utiliser la fonction `strlen()` de la bibliothèque `string.h` dont le prototype est le suivant :

```
size_t strlen(const char *str);
```

avec `size_t` un entier (`~int`) correspondant au nombre de caractères présents dans la chaîne de caractères `str` passée en paramètre (pour rappel, `char *str` est équivalent à `char str[]`).

NB : un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche : “PIERRE” n'est pas palindrome, alors que “OTTO”, “kayak” et “23432” le sont.

## Partie 4 : Aléatoire (3 points)

### Exercice 8 - Qui commence ? (3 points)

Nous souhaitons créer une fonction qui tire aléatoirement un numéro pour savoir qui commence à jouer dans un jeu. Créez une fonction `quiCommence(int nbJoueurs)` qui retourne un nombre entier aléatoire entre 1 et `nbJoueurs`. Cette fonction devra retourner des valeurs différentes à chaque exécutions afin de ne pas permettre de prédiction du tirage.