

Programmation C

TD3 : Structures - Pointeurs

Pré-requis : chapitres 1 à 7 du cours de Programmation Bas Niveau.

Exercice 1 : Référencement / Déréférencement

Soit le code suivant :

```
int i = 3;  
int * p1;  
int ** p2;
```

Soient x l'adresse de i , et y l'adresse de $p1$. Pour chacune des instructions suivantes, dire si elle est valable et quel en est le résultat (considérez le code de manière impérative, et une instruction non valable comme ayant été supprimée) :

```
p1 = &i;  
p2 = *p1;  
p2 = &i;  
p2 = &p1;  
int j = *p2;  
int j = *p1;  
int j = **p2 + 1;
```

Exercice 2 : Structure point2D

1. Définissez une structure **point2D** qui encode les coordonnées d'un point (abscisse et ordonnée) sous forme de float. Définissez également un type **Point2D** pour simplifier l'appel à `struct point2D`.
2. Écrivez une fonction **float distance2D (Point2D p1, Point2D p2)** qui calcule la distance entre deux points. Rappel : la distance cartésienne entre deux points A et B se calcule comme suit : $\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$.
3. Écrivez une fonction **distance2DOrigine** qui calcule et renvoie la distance d'un point à l'origine de son repère.
4. Écrivez une fonction **Point2D milieuSegment (Point2D p1, Point2D p2)** qui renvoie le milieu du segment $[p1p2]$.
5. Écrivez une fonction **Point2D symetrieOrigine (Point2D p1)** qui renvoie le symétrique de $p1$ par rapport à l'origine du repère.
6. Écrivez une fonction **void symetrieOriginePointeur (Point2D * p)** qui modifie le **Point2D** pointé par p par son symétrique par rapport à l'origine du repère.

7. Écrivez une fonction **void reinitialisePoint (Point2D * p)** qui réinitialise le Point2D pointé par *p* à (0,0).
8. Écrivez une fonction **void setPoint (Point2D * p1, Point2D p2)** qui modifie le Point2D pointé par *p1* en lui donnant les valeurs de *p2*.
9. Écrivez une fonction **void affichePoint2D (Point2D p)** qui affiche le Point2D *p* passé en paramètre.
10. Écrivez une fonction **Point2D * symetrieTableauPoint2D (Point2D * tab, int taille)** qui renvoie un tableau alloué dynamiquement qui contient les symétriques par rapport à l'origine de chaque Point2D du tableau *tab* de taille *taille*.
11. Réalisez un main pour tester chacune de ces fonctions.

Exercice 3 : initialisation et affichage d'une table de multiplication

1. Écrivez une fonction **int * initTable (int taille, int chiffre)** qui renvoie un pointeur vers un tableau d'entiers de taille **taille** après avoir initialisé les valeurs de ce tableau à la table de multiplication du chiffre **chiffre** de la première valeur à la valeur **taille**. Par exemple, **initTable (5, 2)** renverra un pointeur vers un tableau qui contient {2, 4, 6, 8, 10 }.
2. Écrivez une fonction **void afficheTable (int * table, int taille)** qui affiche la table de multiplication **table**.