

Introduction à C/C++

TD5: Structures, Fichiers.

Vincent GAUDILLIERE. vincent.gaudilliere@inria.fr.

1 Milieu d'un segment.

On définit ci-dessous les types Point et Segment :

```
typedef struct{                typedef struct{
    double x;                    Point P1;
    double y;                    Point P2;
}Point;                          }Segment;
```

Donner la définition complète des 3 fonctions suivantes, qui permettent de calculer les coordonnées du milieu d'un segment passé en paramètre.

```
Point milieu1(Segment s)
Point milieu2(Segment * s)
void milieu3(Segment * s, Point * milieu)
```

Vérifier que l'appel de ces 3 fonctions donne bien le même résultat.

2 Intersection de deux disques.

On définit la structure Disque suivante :

```
typedef struct{
    Point C; // centre du disque
    double r; // rayon du disque
}Disque;
```

Ecrire une fonction `int inter(Disque * D1, Disque * D2)` qui prend en entrée deux pointeurs sur Disque, et renvoie 1 si l'intersection des deux disques est non vide, 0 si les deux disques sont tangents, et -1 sinon.

Votre programme doit demander à l'utilisateur d'entrer au clavier les paramètres de chacun des disques, puis appeler la fonction `inter`, et enfin indiquer à l'utilisateur la situation des deux disques.

3 Âge d'un utilisateur.

Ecrire un programme qui demande à l'utilisateur son âge, puis écrit dans un fichier `age.txt` la phrase suivante : "L'utilisateur de ce programme a x ans" (avec l'âge de l'utilisateur à la place du "x").

4 Sauvegarde d'un jeu Pacman.

On considère un jeu *Pacman* très simplifié, dont l'état aurait été sauvegardé dans un fichier `Pacman.txt` (reçu par ailleurs). Ici, l'état du jeu est entièrement représenté par une variable `Partie1` de type `Etat` :

```
typedef struct{           typedef struct{           typedef struct{
    Pacman Pm;             int X;                       int X;
    Fantome F[5];         int Y;                       int Y;
    int score;            int VX;                      int VX;
}Etat;                   int VY;                      int VY;
                           }Pacman;                    int actif;
                           }Fantome;
```

Le fichier `Pacman.txt` est écrit selon le format suivant :

```
[score]
[paramètres du Pacman séparés par un espace (dans l'ordre de la structure)]
[paramètres du fantôme n°1 séparés par un espace (dans l'ordre de la structure)]
[paramètres du fantôme n°2 séparés par un espace (dans l'ordre de la structure)]
[paramètres du fantôme n°3 séparés par un espace (dans l'ordre de la structure)]
[paramètres du fantôme n°4 séparés par un espace (dans l'ordre de la structure)]
[paramètres du fantôme n°5 séparés par un espace (dans l'ordre de la structure)]
```

Ecrire un programme qui lit le fichier `Pacman.txt` et remplit les structures correspondantes. Ces structures doivent bien sûr être déclarées au préalable.

5 Opérations élémentaires.

Le fichier `operations.txt` (reçu par ailleurs) contient quelques opérations élémentaires impliquant des entiers. Les lignes du fichier sont toutes écrites selon le format suivant :

```
[un entier][un espace][une opérande][un espace][un entier]
```

Ecrire un programme qui lit le fichier `operations.txt`, effectue les opérations décrites dans celui-ci, et écrit les résultats dans un fichier `resultats.txt`. Plus précisément, la ligne i du second fichier doit contenir (uniquement) le résultat de l'opération décrite à la ligne i du premier fichier.

(Utile : l'instruction `switch` : [https://fr.wikipedia.org/wiki/Switch_\(instruction\)](https://fr.wikipedia.org/wiki/Switch_(instruction)))