Licence EEA Semestre 3 – Contrôle de TP – Janvier 2023

HAE303 – LANGAGE C

Durée : 1h.20

– Exercice : Labyrinthe –

I.1 - Préparation

1. Créez un répertoire dont le titre est votre nom et prénom, au format : `NOM_Prenom`. Par exemple : `DUPONT_Pierre`.
2. Placez vous dans ce répertoire avec le terminal. Vous écrirez la totalité de vos codes à l'intérieur de ce répertoire.
3. Vous devrez transmettre ce répertoire à l'enseignant, en suivant une procédure qui vous sera précisée à la fin de l'épreuve.

Toute copie qui sera rendue sans suivre cette procédure vaudra 0/20.

I.2 - Compilation

On donne le code ci-dessous. Téléchargez-le *(ne faites PAS un copier/coller à partir de ce document)* et copiez le code dans un fichier que vous appellerez "**Q1.cpp**". Ensuite, compilez le, constatez que le compilateur mentionne des erreurs, et corrigez ces erreurs.

```
#include <iostream>
#include <vector>

using namespace std;

struct Joueur {
   int positionX;
   int positionY;
};

struct Labyrinthe {
   vector<vector<char>> disposition;
};


void initialiserLabyrinthe(Labyrinthe* labyrinthe) {
   (*labyrinthe).disposition = {
      {'#', 'S', '#', '#', '#', '#', '#', '#', '#'},
      {'#', ' ', '#', ' ', '#', '#', ' ', '#', ' '},
      {'#', ' ', ' ', ' ', '#', ' ', ' ', ' ', ''},
      {' ', ' ', '#', ' ', ' ', ' ', '#', ' ', '#'},
      {'#', '#', '#', '#', '#', '#', '#', ' ', '#'},
      {'#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#'},
```

```
        {'#', '#', '#', '#', '#', '#', '#', ' ', '#'},
        {'#', '#', '#', '#', '#', '#', '#', 'E', '#'},
    };
}

void afficherLabyrinthe(Labyrinthe labyrinthe) {
    for (unsigned long i = 0; i < labyrinthe.disposition.size(); i++) {
        for (unsigned long j = 0; j < labyrinthe.disposition[i].size(); j
            ++) {
            cout << labyrinthe.disposition[i][j] << " ";
        }
        cout << "\n";
    }
}

int main() {
    Joueur joueur1;
    joueur1.positionX = 0;
    joueur1.positionY = 0;


    Labyrinthe labyrinthe;
    initialiserLabyrinthe(&labyrinthe);

    cout << "\nLabyrinthe Initial:" << "\n";
    afficherLabyrinthe(labyrinthe);


    return 0;
}
```

I.3 - Prise en main

1 - Compilez **Q1.cpp** et rassurez vous sur le fait qu'il fonctionne et ne comporte pas d'erreur. Regardez ce qu'il fait, faites la relation entre le programme et ce qui est affiché quand vous lancez le programme.

2 - Si vous avez compris, ce programme s'occupe de gérer un labyrinthe, contenu dans la structure `Labyrinthe`, où les "#" sont les murs, les " " sont les zones où le joueur peut circuler, "E" représente l'entrée et "S" la sortie. Les coordonnées du joueur dans le labyrinthe sont stockées à l'intérieur de la structure "Joueur".

La fonction d'affichage qui est donnée, `afficherLabyrinthe`, n'est pas satisfaisante parce qu'elle n'affiche pas la position du joueur dans le labyrinthe.

Dans **Q1.cpp**, modifiez la fonction `afficheLabyrinthe` pour que son prototype devienne :
`void afficherLabyrinthe(Labyrinthe labyrinthe, Joueur joueur);`
et en modifiant son code pour qu'elle représente par la lettre "J" le joueur dans le labyrinthe, à la position donnée par les coordonnées contenues dans la structure Joueur. Cette fonction ne se souciera pas du fait que ces coordonnées pourraient être dans un mur. Elle ne se souciera pas non plus de si le joueur est contenu dans la grille de jeu : on considère ici que les coordonnées données sont correctes.

3 - Toujours dans Dans **Q1.cpp**, Modifiez la fonction `main` pour qu'elle place `joueur1` la case notée `E`, et pour qu'elle fasse un appel correct à la fonction `afficherLabyrinthe`. On ne demande

pas de faire un algorithme particulier, juste de mettre, en dur, les bonnes coordonnées dans la variable `joueur1`.

I.4 - Se déplacer dans le labyrinthe : version 1

1 - Copiez/collez **Q1.cpp** dans un nouveau fichier **Q2.cpp**. Dans ce nouveau fichier, créez une fonction :

```
int DeplacementPasseMuraille(Labyrinthe* labyrinthe, Joueur* J, int deltaX, int deltaY);
```

qui permet au joueur de se déplacer sur le labyrinthe de `deltaX` cases selon x et `deltaY` cases selon y. Ces valeurs seront parmi $\{-1, 0, 1\}$, le signe indiquant la direction du déplacement. Les valeurs plus grandes que 1 ne sont pas à considérer (on ne se souciera pas de ce cas cas dans le code). Dans cette version, le joueur peut traverser les murs : on ne se soucie donc pas du fait qu'il y a des murs. En revanche, il ne doit pas pouvoir sortir de l'espace du labyrinthe. Si jamais les deltaX et deltaY donnés sont tels que le joueur sortirait du labyrinthe, alors la fonction ne doit pas modifier les coordonnées de `J` et renvoyer 0. Dans le cas contraire, la variable `J` doit bien être mise à jour pour prendre en compte le déplacement, et renvoyer 1.

Conseil algorithmique : Vous n'êtez pas obligé de faire ainsi mais nous vous conseillons d'utiliser 2 variables x et y et d'y recopier les coordonnées du joueur, puis d'y appliquer les delta. Ensuite, regarder si les coordonnées obtenues sont bien contenues dans le labyrinthe.

2 - Toujours dans **Q2.cpp**, modifiez le programme pour que le joueur se déplace 3 fois d'une case à droite (seule la premiere devrait être réalisée compte tenu de la position initiale du joueur), puis 2 fois d'une case vers le bas (ce qui ne devrait pas se faire), puis trois fois d'une case vers le haut, et enfin deux fois d'une case à gauche.

I.5 - Se déplacer dans le labyrinthe : version 2

1 - Copiez/collez **Q2.cpp** dans un nouveau fichier **Q3.cpp**. Dans ce nouveau fichier, créez une fonction :

```
int DeplacementJoueur(Labyrinthe* labyrinthe, Joueur* J, int deltaX, int deltaY);
```

qui fait exactement la même chose que précédemment mais qui en plus fait en sorte que le joueur ne puisse pas passer à travers les murs. Elle renvoie 1 si le déplacement du joueur est correct (joueur contenu dans le labyrinthe ET joueur qui ne passe pas à travers les murs), 0 sinon.

2 - Toujours dans **Q3.cpp**, dans le `main`, remplacez les appels à `DeplacementPasseMuraille`, par des appels à `DeplacementJoueur`.

I.6 - Jeu du labyrinthe

1 - Copiez/collez **Q3.cpp** dans un nouveau fichier **Q4.cpp**. Dans ce nouveau fichier, vous remplacerez le `main` pour qu'il permette de jouer au jeu de labyrinthe. On utilisera cin `pour` les commandes de déplacement :

- Z pour se déplacer vers le haut,
- S pour se déplacer vers le bas,
- Q pour se déplacer à gauche,
- D pour se déplacer à droite.

Le jeu doit réafficher le labyrinthe après chaque déplacement. Il doit aussi afficher "Aie, Ca fait mal!" si le déplacement demandé est incorrect.

Important : Dans cette question, on ne modifiera que le `main` et aucune autre fonction.