作者：董佳昕

一、背景

实现一款名为**Adventure**的CLI游戏。玩家必须探索一个城堡，其中包含多个关卡和多个房间。 玩家的任务是找到囚禁公主的房间，并带她离开!

房间有很多种类型，每种类型都有不同的出口。 注意，在其中一个房间里有一个怪物，它的确切位置是未知的。 但一旦玩家遇到怪物，游戏就结束了。

程序总是显示玩家进入的房间的信息:房间的名称，有多少个出口，以及所有出口的名称。 例如，当游戏开始时，玩家在城堡的大厅中，程序输出:

```
Welcome to the lobby. There are 3 exits: east, west and up.
Enter your command:
```

然后玩家可以输入go命令，然后输入他/她想要通过的出口名称，例如:

```
go east
```

因此，玩家将进入东边的房间。 该程序提供了该房间的信息，就像上面显示的大厅。 这个过程重复。

在这个过程中，如果玩家进入一个有怪物的房间，程序就会显示一条信息，游戏就结束了。 同样地，当玩家进入公主所在的密室时，程序便会显示角色与公主之间的对话。 在那之后，她准备带着这个角色离开。 然后玩家必须找到自己的出路。 离开城堡的唯一途径是通过大厅。

为了简化实现，所有打印的消息和用户输入都以英文显示。

二、实现思路

2.1 地图的存储

通过一个二维**int**数组**map**，记录当前游戏的地图

每一个位置有5种不同的取值：

```
//房间的种类
const int cannot_go = 0;    //该房间不能进入
const int lobby = 1;        //出生点(大厅)
const int princessRoom = 2; //房间有公主
const int monsterRoom = 3;  //房间有怪兽
const int emptyRoom = 4;    //空房间
```

2.2 地图的创建

根据输入的地图大小**n*m**，依次创建出：

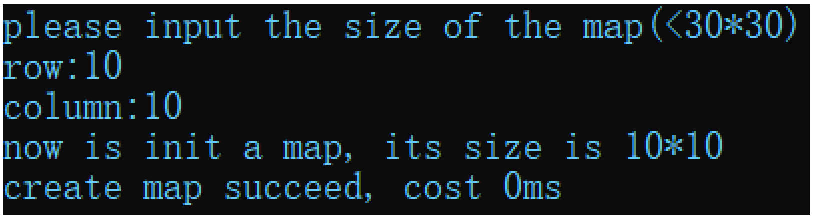
(1)玩家的初始位置

(2)公主的初始位置

(3)玩家==>公主之间的路径（为了保证存在答案）

(4)怪兽的位置

(5)其它房间的状态（不能进入 or 空房间）



```
please input the size of the map(<30*30)
row:10
column:10
now is init a map, its size is 10*10
create map succeed, cost 0ms
```

2.3 对玩家的控制

(1)首先，通过调用**GetNextStep()**函数，获取当前位置的出口

(2)为了便于玩家游玩，在每一步，都会输出当前已经经过的房间的类型

(a)白色的**@**表示玩家的当前位置

(b)红色的*****表示出生点/大厅的位置

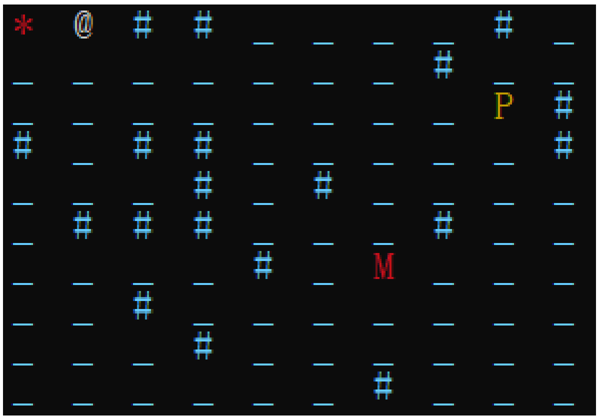
(c)蓝色的_表示空房间

(d)蓝色的**?**表示没有去过该房间

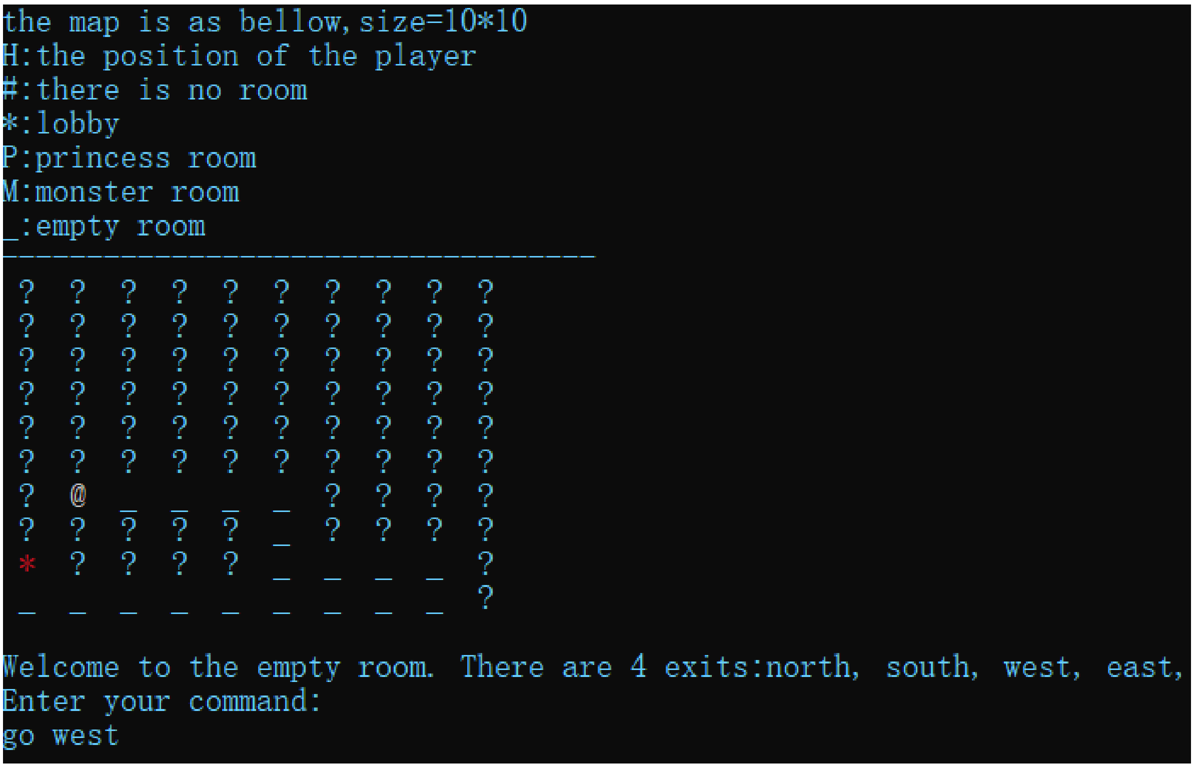
(e)蓝色的**#**表示该房间不能进入

(f)黄色的**P**表示公主位置

(g)红色的**M**表示怪兽的位置



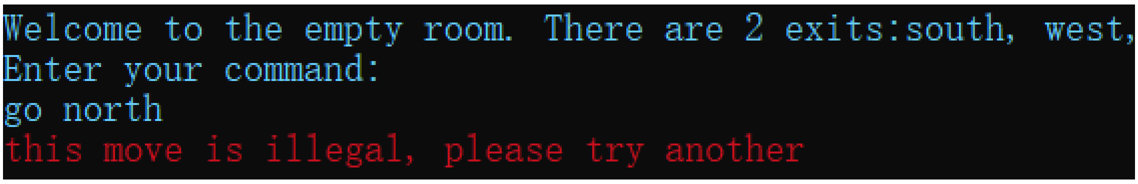
```
*  @  #  #  _  _  _  _  #  _
_  _  _  _  _  _  _  #  _  _
_  _  _  _  _  _  _  _  P  #
#  _  #  #  _  _  _  _  _  #
_  _  _  #  _  #  _  _  _  _
_  #  #  #  _  _  _  #  _  _
_  _  _  _  #  _  M  _  _  _
_  _  #  _  _  _  _  _  _  _
_  _  _  #  _  _  _  _  _  _
_  _  _  _  _  _  #  _  _  _
```



```
the map is as bellow,size=10*10
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
 ?  @  _  _  _  _  ?  ?  ?  ?
 ?  ?  ?  ?  ?  _  ?  ?  ?  ?
 *  ?  ?  ?  ?  _  _  _  _  ?
 _  _  _  _  _  _  _  _  _  ?

Welcome to the empty room. There are 4 exits:north, south, west, east,
Enter your command:
go west
```

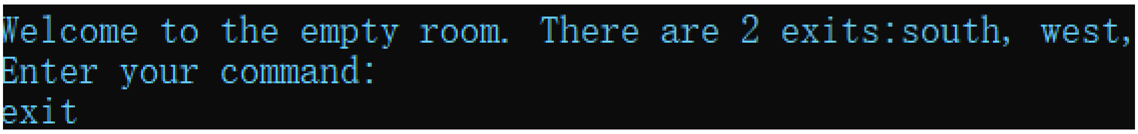
(3)当输入正确的指令时，程序会调用**MovePlayer()**函数，修改玩家的位置

(4)当输入错误的指令时，程序会输出，并且不修改玩家的位置



```
Welcome to the empty room. There are 2 exits:south, west,
Enter your command:
go north
this move is illegal, please try another
```

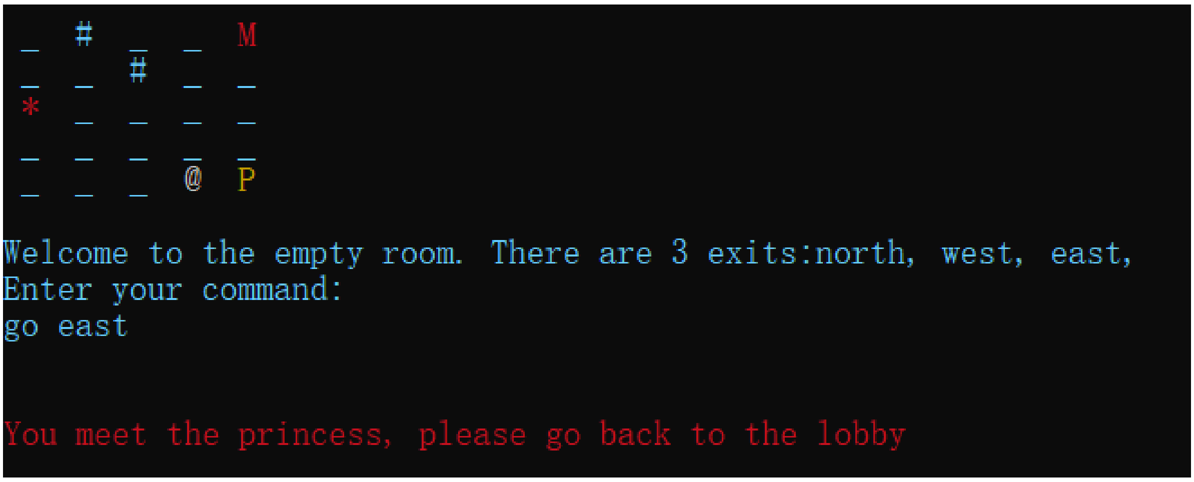
(5)如果要退出程序，需要输入**exit**



```
Welcome to the empty room. There are 2 exits:south, west,
Enter your command:
exit
```

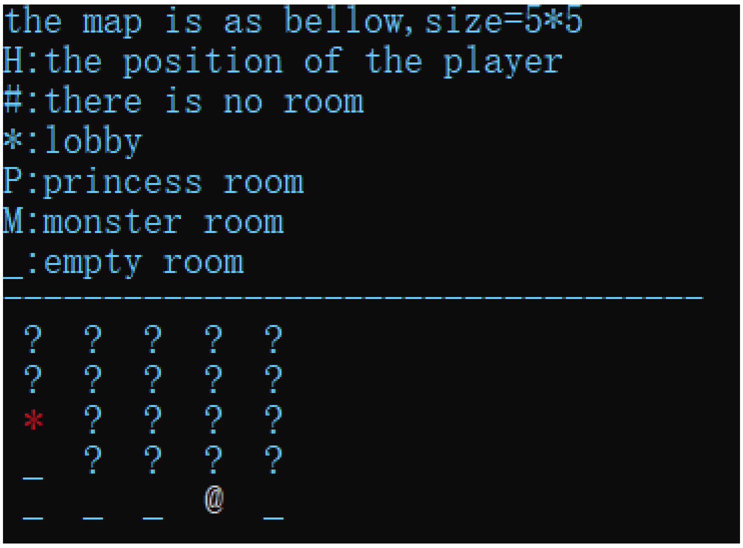
(5)当玩家遇到怪兽时，游戏结束

(6)当玩家遇到公主时，程序会输出**You meet the princess, please go back to the lobby**



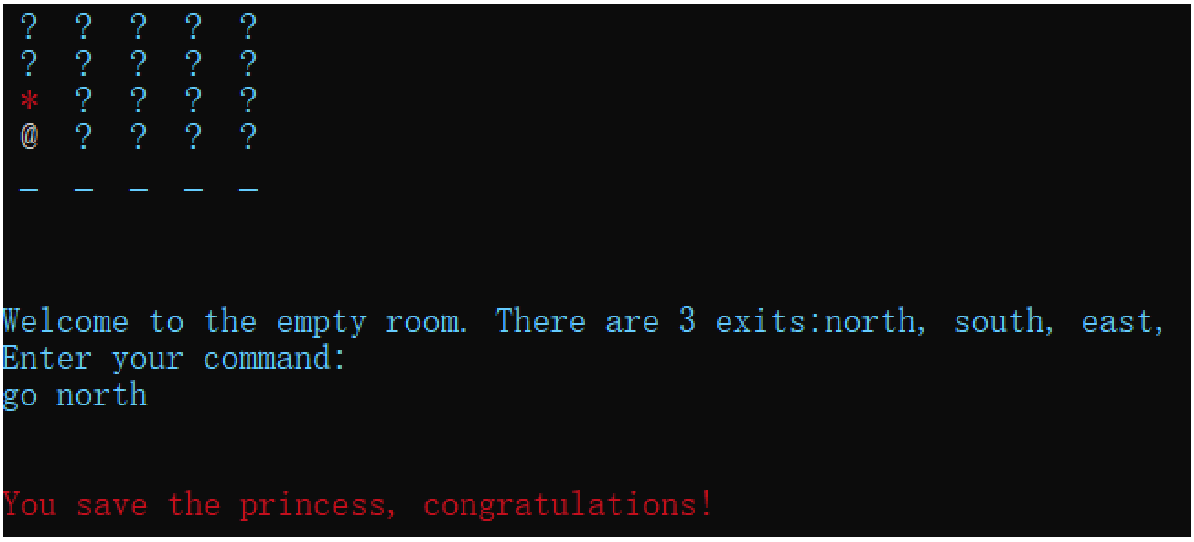
```
 _  #  _  _  M
 _  _  #  _  _
 *  _  _  _  _
 _  _  _  _  _
 _  _  _  @  P

Welcome to the empty room. There are 3 exits:north, west, east,
Enter your command:
go east


You meet the princess, please go back to the lobby
```

并且在后续的地图中，可以看到，公主原来所在的房间里面已经没有公主了



```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?
 *  ?  ?  ?  ?
 _  ?  ?  ?  ?
 _  _  _  @  _
```

(7)在找到公主的情况下，返回大厅，游戏结束



```
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?
 *  ?  ?  ?  ?
 @  ?  ?  ?  ?

 _  _  _  _  _



Welcome to the empty room. There are 3 exits:north, south, east,
Enter your command:
go north


You save the princess, congratulations!
```

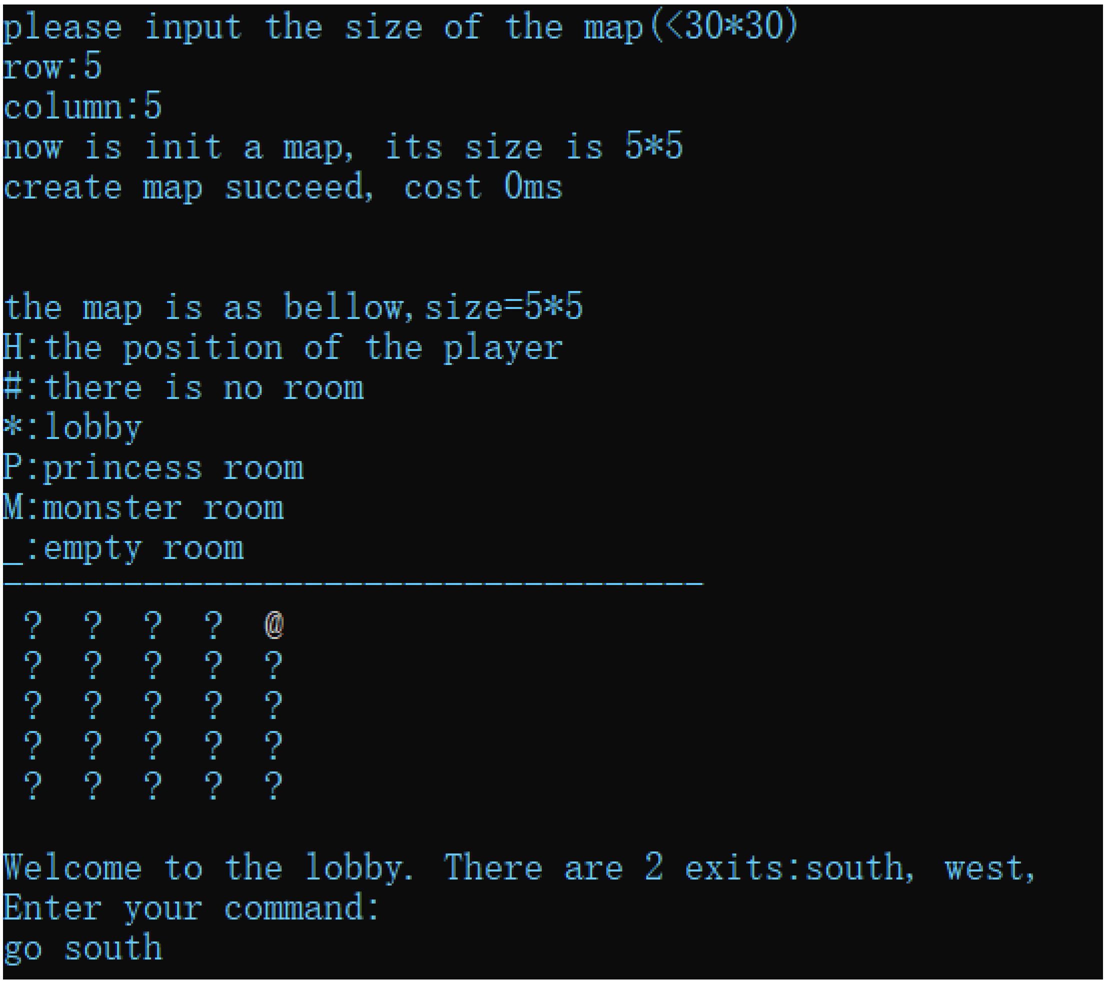
2.4 类的封装

游戏的控制程序封装为一个类**Adventure**，通过调用**adventure.StartAdventure()**函数，进行游戏

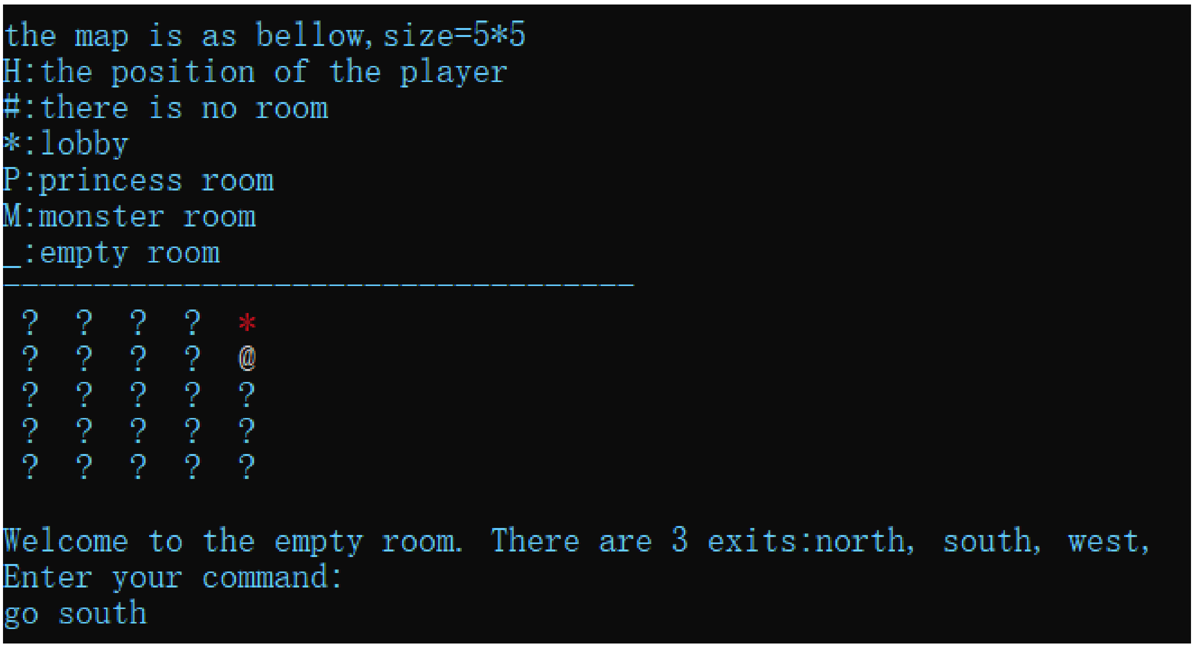
三、测试样例

由于每一次的地图为系统随机生成，本程序并没有固定的测试样例

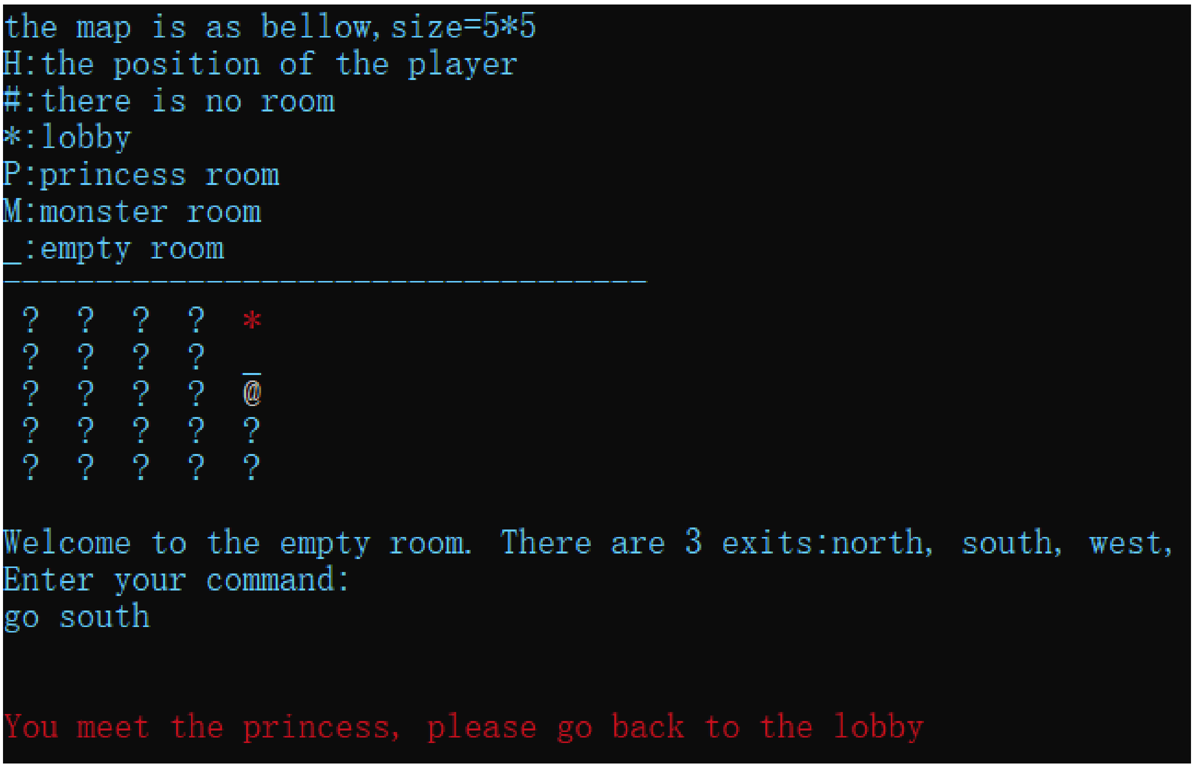
所以，通过一次运行，作为本程序的测试样例



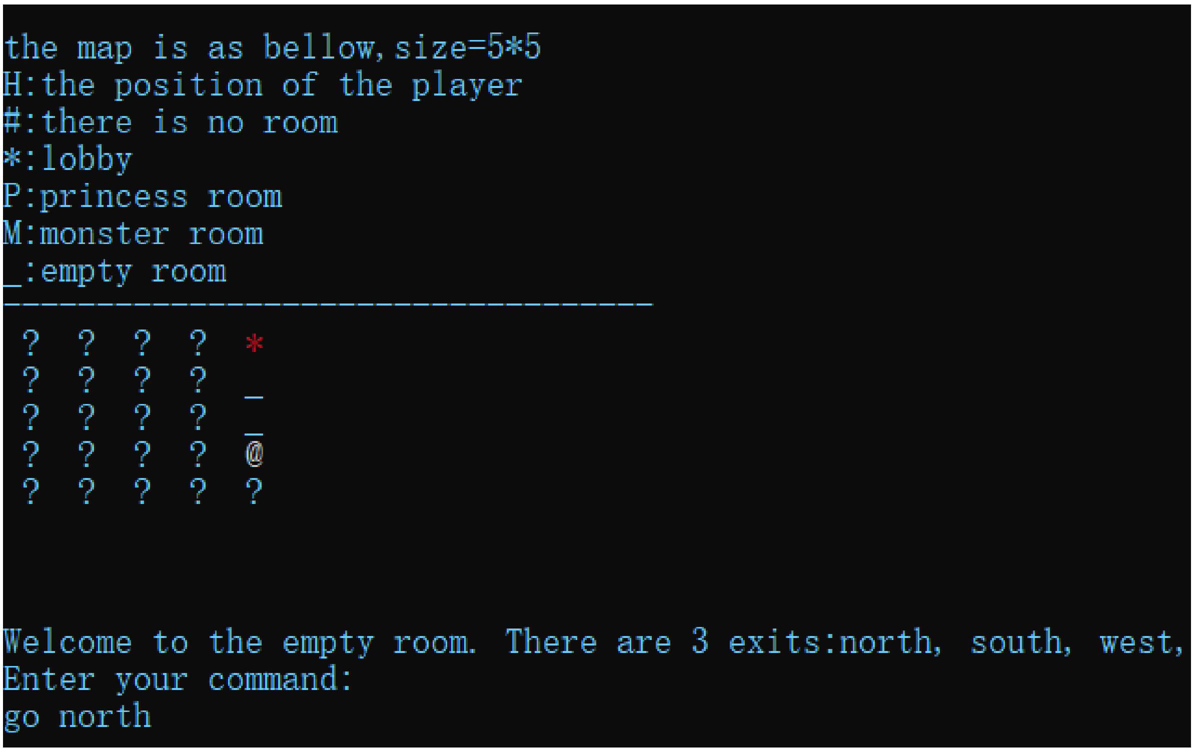
```
please input the size of the map(<30*30)
row:5
column:5
now is init a map, its size is 5*5
create map succeed, cost 0ms


the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  @
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?

Welcome to the lobby. There are 2 exits:south, west,
Enter your command:
go south
```



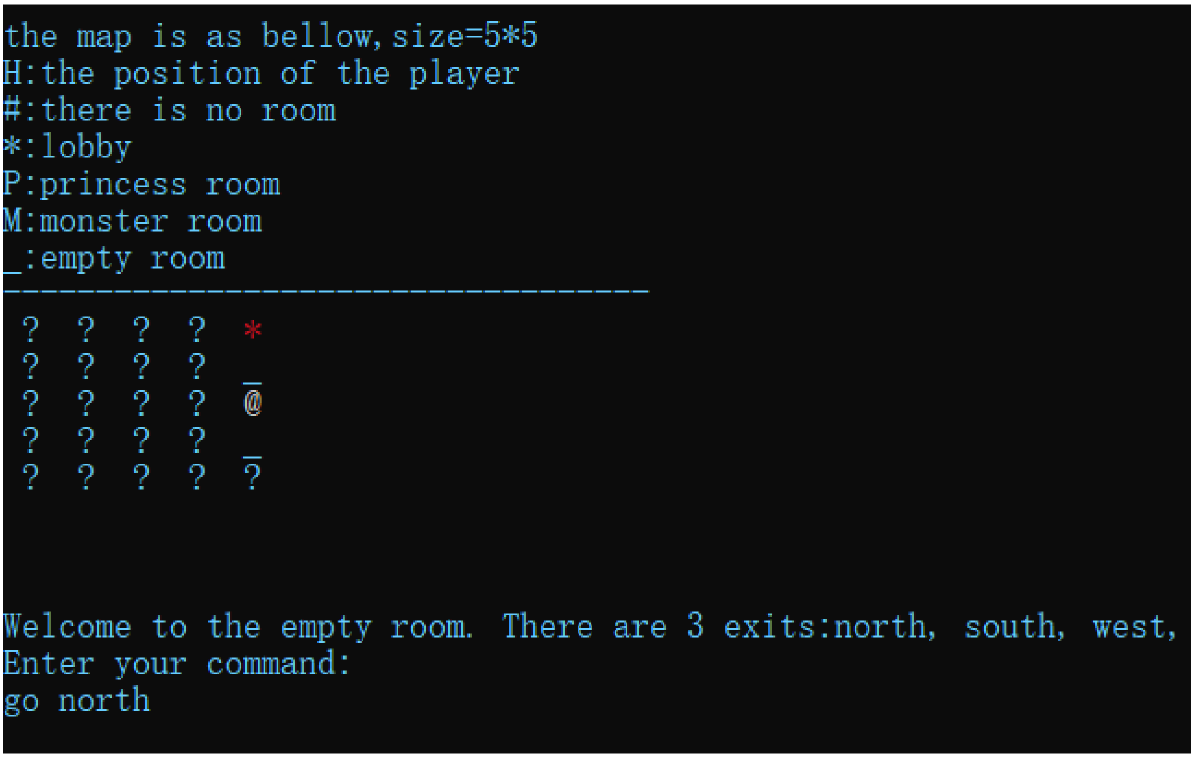
```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  *
 ?  ?  ?  ?  @
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?

Welcome to the empty room. There are 3 exits:north, south, west,
Enter your command:
go south
```



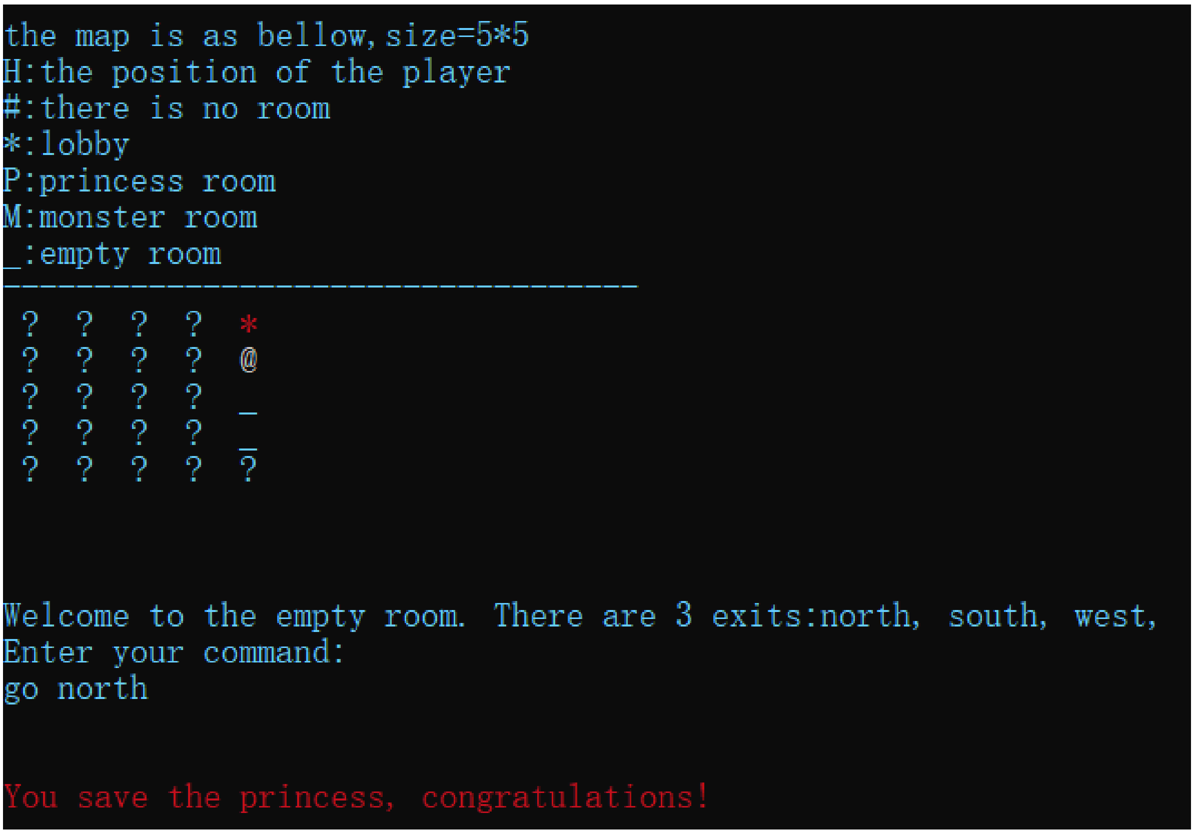
```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  *
 ?  ?  ?  ?  _
 ?  ?  ?  ?  @
 ?  ?  ?  ?  ?
 ?  ?  ?  ?  ?

Welcome to the empty room. There are 3 exits:north, south, west,
Enter your command:
go south


You meet the princess, please go back to the lobby
```



```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  *
 ?  ?  ?  ?  _
 ?  ?  ?  ?  _
 ?  ?  ?  ?  @
 ?  ?  ?  ?  ?



Welcome to the empty room. There are 3 exits:north, south, west,
Enter your command:
go north
```



```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  *
 ?  ?  ?  ?  _
 ?  ?  ?  ?  @
 ?  ?  ?  ?  _
 ?  ?  ?  ?  ?



Welcome to the empty room. There are 3 exits:north, south, west,
Enter your command:
go north
```



```
the map is as bellow,size=5*5
H:the position of the player
#:there is no room
*:lobby
P:princess room
M:monster room
_:empty room
-----------------------------------
 ?  ?  ?  ?  *
 ?  ?  ?  ?  @
 ?  ?  ?  ?  _
 ?  ?  ?  ?  _
 ?  ?  ?  ?  ?



Welcome to the empty room. There are 3 exits:north, south, west,
Enter your command:
go north


You save the princess, congratulations!
```

附录：源代码

```cpp
#include <iostream>
#include <ctime>
#include <algorithm>
#include <string>
using namespace std;
const string rootName[] = {"connot_go","lobby","princess room","monster room","empty room"};
class Adventure {
    static const int maxn = 30+5;

    //地图是一个n行m列的矩阵
    int n, m;
    int map[maxn][maxn];
    bool vis[maxn][maxn];
    bool NextStep[4];

    //房间的种类
    const int cannot_go = 0;    //该房间不能进入
    const int lobby = 1;        //出生点(大厅)
    const int princessRoom = 2; //房间有公主
    const int monsterRoom = 3;  //房间有怪兽
    const int emptyRoom = 4;    //空房间

    //位置信息
    struct Position {
        int init_x, init_y;         //初始的位置==大厅的位置
        int monster_x, monster_y;   //怪兽的位置
        int princess_x, princess_y; //公主的位置
        int now_x, now_y;           //勇者现在的位置
        void Print() {
            printf("初始位置 = (%d,%d)\n", init_x, init_y);
            printf("怪物位置 = (%d,%d)\n", monster_x, monster_y);
            printf("公主位置 = (%d,%d)\n", princess_x, princess_y);
            printf("勇者位置 = (%d,%d)\n", now_x, now_y);
        }
    }position;

    //获得随机数
    int random(int L, int R) {
        int len = R - L + 1;
        return rand() * rand() % len + 1;
    }

    //高亮输出,0-黑,1-红,2-绿,3-黄,4-蓝,5-紫,6-深绿,7-白
    void HighLightPrint(const char* s, int color) {
        switch (color) {
        case 0://黑
            printf("\033[30m"); break;
        case 1://红
            printf("\033[31m"); break;
        case 2://绿
            printf("\033[32m"); break;
        case 3://黄
            printf("\033[33m"); break;
        case 4://蓝
            printf("\033[34m"); break;
        case 5://紫
            printf("\033[35m"); break;
        case 6://深绿
            printf("\033[36m"); break;
        case 7://白
            printf("\033[37m"); break;
        }
        printf("%s", s);
        printf("\033[0m");
    }

    //得到房间(x,y)的类型
    int GetRoom(int x, int y) {
        if (x <= 0 || x > n)return 0;
        if (y <= 0 || y > m)return 0;
        return map[x][y];
    }

    //得到当前房间的类型
    int GetNowRoom() {
        vis[position.now_x][position.now_y] = true;
        return GetRoom(position.now_x, position.now_y);
    }

    //创建出生点->公主的路径
    void CreateSucceedPath() {
        int x1 = position.init_x;
        int y1 = position.init_y;
        int x2 = position.princess_x;
        int y2 = position.princess_y;
        if (x1 <= x2 && y1 <= y2) {//出生点在公主的左下角
            for (int i = x1 + 1; i < x2; i++)
                map[i][y1] = emptyRoom;
            for (int i = y1 + 1; i < y2; i++)
                map[x2][i] = emptyRoom;
            if (y1 != y2)map[x2][y1] = emptyRoom;

        }
        else if (x1 <= x2 && y1 >= y2) {//出生点在公主的左上角
            for (int i = x1 + 1; i < x2; i++)
                map[i][y1] = emptyRoom;
            for (int i = y1 -1; i > y2; i--)
                map[x2][i] = emptyRoom;
            if (y1 != y2)map[x2][y1] = emptyRoom;
        }
        else if (x1 >= x2 && y1 <= y2) {//出生点在公主的右下角
            for (int i = x1 - 1; i > x2; i--)
                map[i][y1] = emptyRoom;
            for (int i = y1 + 1; i < y2; i++)
                map[x2][i] = emptyRoom;
            if (y1 != y2)map[x2][y1] = emptyRoom;
        }
        else if (x1 >= x2 && y1 >= y2) {//出生点在公主的右上角
            for (int i = x1 - 1; i > x2; i--)
                map[i][y1] = emptyRoom;
            for (int i = y1 - 1; i > y2; i--)
                map[x2][i] = emptyRoom;
            if (y1 != y2)map[x2][y1] = emptyRoom;
        }
    }

    //创建大厅、公主、怪物的位置
    void CreateSpecialPosition() {
        //大厅位置
        position.init_x = random(1, n);
        position.init_y = random(1, m);
        map[position.init_x][position.init_y] = lobby;
        //出生点位置==大厅位置
        position.now_x = position.init_x;
```

```cpp
        position.now_y = position.init_y;
        //公主位置
        do {
            position.princess_x = random(1, n);
            position.princess_y = random(1, m);
        } while (map[position.princess_x][position.princess_y] != -1);
        map[position.princess_x][position.princess_y] = princessRoom;
        CreateSucceedPath();
        //怪物位置
        do {
            position.monster_x = random(1, n);
            position.monster_y = random(1, m);
        } while (map[position.monster_x][position.monster_y] != -1);
        map[position.monster_x][position.monster_y] = monsterRoom;
    }

    //创建其它房间,每个房间可以走的概率为x%
    void CreateOtherRoom(int x) {
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= m; j++)
                if (map[i][j] == -1) {
                    int now = random(1, 100);
                    if (now < x)map[i][j] = emptyRoom;
                    else map[i][j] = cannot_go;
                }
    }

    //将地图初始化
    void CreateMap() {
        srand((int)time(0));
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= m; j++)
                map[i][j] = -1, vis[i][j] = false;
        CreateSpecialPosition();
        CreateOtherRoom(80);
        vis[position.init_x][position.init_y] = true;
    }

    //打印地图,0-全部,1-部分
    void PrintMap(int type) {
        cout << "the map is as bellow,size=" << n << "*" << m << endl;
        cout << "H:the position of the player\n";
        cout << "#:there is no room\n";
        cout << "*:lobby\n";
        cout << "P:princess room\n";
        cout << "M:monster room\n";
        cout << "_:empty room\n";
        cout << "-----------------------------------\n";
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= m; j++) {
                if (i == position.now_x && j == position.now_y) {//玩家位置
                    HighLightPrint(" @ ", 7);
                    continue;
                }
                if (!vis[i][j] && type == 1) {
                    cout << " ? ";
                    continue;
                }
                switch (map[i][j]) {
                case 0:cout << " # "; break;//不能走
                case 1:HighLightPrint(" * ", 1); break;//出生点(大厅)
                case 2:HighLightPrint(" P ", 3); break;//公主
                case 3:HighLightPrint(" M ", 1); break;//怪兽
                case 4:cout << " _ "; break;//空房间
                default:cout << " | "; break;
                }
            }
            printf("\n");
        }
        printf("\n");
    }

    //北-0,南-1,西-2,东-3
    void MovePlayer(int head) {
        int nextRoom;
        if (head == 0) {//向上(北)
            nextRoom = GetRoom(position.now_x - 1, position.now_y);
            if (nextRoom == 0)
                HighLightPrint("this move is illegal, please try another\n", 1);
            else
                position.now_x--;
        }
        else if (head == 1) {//向下(南)
            nextRoom = GetRoom(position.now_x + 1, position.now_y);
            if (nextRoom == 0)
                HighLightPrint("this move is illegal, please try another\n", 1);
            else
                position.now_x++;
        }
        else if (head == 2) {//向左(西)
            nextRoom = GetRoom(position.now_x, position.now_y - 1);
            if (nextRoom == 0)
                HighLightPrint("this move is illegal, please try another\n", 1);
            else
                position.now_y--;
        }
        else {//向右(东)
            nextRoom = GetRoom(position.now_x, position.now_y + 1);
            if (nextRoom == 0)
                HighLightPrint("this move is illegal, please try another\n", 1);
            else
                position.now_y++;
        }
    }

    //获取当前房间的出口
    int GetNextStep() {
        int x = position.now_x, y = position.now_y;
        int ans = 0;
        //北-0,南-1,西-2,东-3
        if (GetRoom(x - 1, y) != 0)NextStep[0] = true, ans++;
        else NextStep[0] = false;
```

```cpp
        if (GetRoom(x + 1, y) != 0)NextStep[1] = true, ans++;
        else NextStep[1] = false;
        if (GetRoom(x, y - 1) != 0)NextStep[2] = true, ans++;
        else NextStep[2] = false;
        if (GetRoom(x, y + 1) != 0)NextStep[3] = true, ans++;
        else NextStep[3] = false;
        return ans;
    }

    //将公主从房间带出
    void TakeOutPrincess() {
        map[position.princess_x][position.princess_y] = emptyRoom;
    }
public:
    //进行游戏的初始化
    void reset() {
        cout << "please input the size of the map(<30*30)\n";
        cout << "row:";
        cin >> n;
        //n = 10;
        cout << "column:";
        cin >> m;
        //m = 10;
        cout << "now is init a map, its size is " << n << "*" << m << "\n";
        int nowTime = clock();
        CreateMap();
        cout << "create map succeed, cost " << (clock() - nowTime) << "ms\n\n\n";
    }

    //开始游戏
    void StartAdventure() {
        reset();
        string command;
        //找公主
        while (true) {
            if (GetNowRoom() == monsterRoom) {
                HighLightPrint("You meet the monster, game end\n", 1);
                HighLightPrint("Please try another time\n", 1);
                cout << "\n\n";
                return;
            }
            else if (GetNowRoom() == princessRoom) {
                HighLightPrint("You meet the princess, please go back to the lobby\n", 1);
                TakeOutPrincess();
                cout << "\n\n";
                break;
            }

            int room = GetNowRoom();
            PrintMap(1);
            cout << "Welcome to the " << rootName[room] << ". ";
            cout << "There are " << GetNextStep() << " exits:";
            //北-0,南-1,西-2,东-3
            if (NextStep[0])cout << "north, ";
            if (NextStep[1])cout << "south, ";
            if (NextStep[2])cout << "west, ";
            if (NextStep[3])cout << "east, ";
            cout << endl;
            cout << "Enter your command:" << endl;

            cin >> command;
            if (command == "exit")return;
            if (command != "go") {
                HighLightPrint("the command is illegal\n\n\n", 1);
                continue;
            }
            cin >> command;

            if (command == "north")MovePlayer(0);
            else if (command == "south")MovePlayer(1);
            else if (command == "west")MovePlayer(2);
            else if (command == "east")MovePlayer(3);
            else HighLightPrint("the command is illegal\n", 1);
            cout << "\n\n";
        }
        //回大厅
        while (true) {
            if (GetNowRoom() == monsterRoom) {
                HighLightPrint("You meet the monster, game end\n", 1);
                HighLightPrint("Please try another time\n", 1);
                return;
            }
            else if (GetNowRoom() == lobby) {
                HighLightPrint("You save the princess, congratulations!\n", 1);
                return;
            }

            int room = GetNowRoom();
            PrintMap(1);
            cout << "\n\n";
            cout << "Welcome to the " << rootName[room] << ". ";
            cout << "There are " << GetNextStep() << " exits:";
            //北-0,南-1,西-2,东-3
            if (NextStep[0])cout << "north, ";
            if (NextStep[1])cout << "south, ";
            if (NextStep[2])cout << "west, ";
            if (NextStep[3])cout << "east, ";
            cout << endl;
            cout << "Enter your command:" << endl;

            cin >> command;
            if (command == "exit")return;
            if (command != "go") {
                HighLightPrint("the command is illegal\n\n\n", 1);
                continue;
            }
            cin >> command;

            if (command == "north")MovePlayer(0);
            else if (command == "south")MovePlayer(1);
            else if (command == "west")MovePlayer(2);
            else if (command == "east")MovePlayer(3);
            else HighLightPrint("the command is illegal\n", 1);
            cout << "\n\n";
        }
    }
};
Adventure now;
int main(){
    now.StartAdventure();
    return 0;
}
```