吉林大学 2019-2020 学年第 2 学期

《数据结构》课程设计 B 题

1997 年，诺基亚公司发布了贪吃蛇游戏，并将其内置于诺基亚 6110 手机中，使这款游戏迅速风靡全球，成为一代经典。一般的观点认为，贪吃蛇是手机游戏的鼻祖。

与传统单人贪吃蛇不同的是，本题中的贪吃蛇游戏为双人对战。请你编写程序控制贪吃蛇，在动态变化的场地中与对手比拼，在最短的时间内尽可能多地吃掉食物。

游戏规则：

游戏在一张二维地图上展开，地图上有蛇、墙和障碍物、道具、食物，如图 1 所示。蛇有 2 条：白蛇（□□□）和粉蛇（■■■），初始长度为 3 格。食物由●表示。墙和障碍物由■表示。道具分为两种：第 1 种为“倍速道具▲”，全场 1 个，吃该道具后，速度加倍，50 步后恢复原速度；第二种道具为“斜走道具★”，全场 2 个，正常情况下，蛇移动方向为上、下、左、右 4 个方向，但吃该道具后，还可以斜着走，即上、下、左、右、上左、下左、上右、下右 8 个方向都可以行进，该道具吃后永久有效。蛇头碰到墙或障碍物、自己的身体、对方的身体均被判定为死亡。游戏具体实例请见附件 2。

图 1

游戏每面（也可以称为关卡）随机放置 10 个食物。第一面只有食物没有障碍物和道具，第二面开始出现随机放置的障碍物和道具。

胜负判定：

先吃到 35 个食物的一方获胜，在此期间若任何一条蛇死亡，判对方获胜。此外，若距任意一方吃到食物后 200 步内双方均未再吃到食物，则判超时，此时吃食物多的一方获胜，若两条蛇吃的食物一样多，则判为平局。

代码实现：

老师提供程序框架和图形显示，学生只需按接口要求编写相关函数，实现操控贪吃蛇的核心算法即可，无需负责图形显示。实现语言为 C/C++，不限编译器与开发环境。

本题程序由命令行界面模拟图形界面，用命令行界面的 n+2 行 m+4 列二维网格表示游戏地图，通过 printf("■")等画游戏地图中的一个点格。这里需特别注意的是，由于"■"等特殊字符在命令行中占 2 个字符的宽度，所以命令行的每 2 列对应游戏地图的 1 列。综上，游戏地图的坐标系如图 2 所示，其相邻两点的列坐标值相差 2 而不是 1。例如图 2 中粉蛇由 3 个点组成，其坐标分别为(3,6)、(3,4)、(3,2)。第 0 行、第 n+1 行、第 0 列、第 m+2 列为四周的墙。

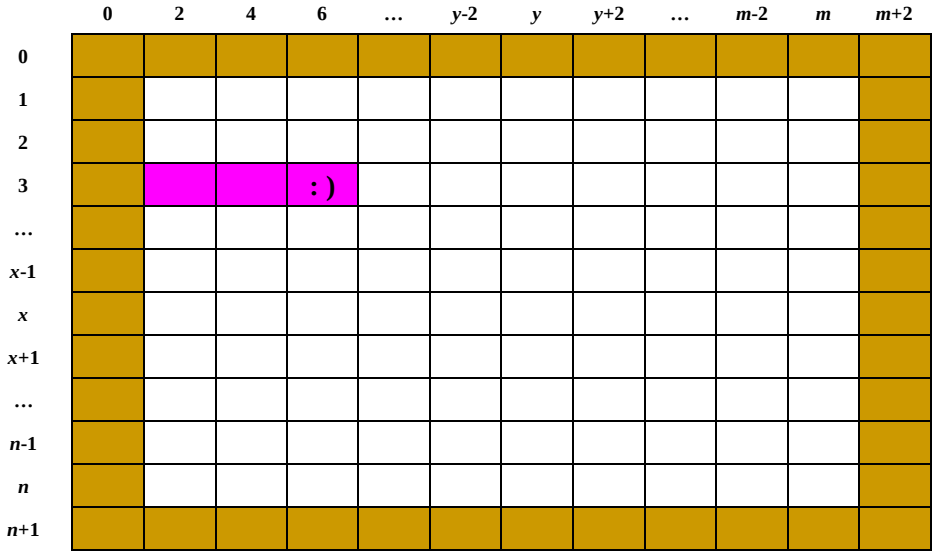

图 2

你的任务是编写程序控制贪吃蛇，即根据当前的游戏地图和蛇的位置，做出决策，给出下一步贪吃蛇行进的方向。从而吃掉食物并躲避障碍物和对方。为此你至少需要编写如下 2 个函数：

（1）machine_move 函数

int machine_move (point snake[5][100], int len[5], int direct[5], int t, GamePanel gp)

该函数的功能是：即根据当前的游戏地图、贪吃蛇当前的位置和行进方向，返回下一步应行进的方向。其中point为结构体struct point{int x; int y;}，表示游戏地图中某点的行列坐标。snake数组表示蛇的坐标，具体地，由于一条蛇由多个点/格组成，故snake[t][i]表示第t条蛇的第i个点（下标从1开始），len[t]表示蛇t的长度，即第t条蛇由地图中的snake[t][1]...snake[t][len[t]]点格组成。若图2中的粉蛇编号为t，则len[t]=3，且snake[t][1].x=3, snake[t][1].y=6; snake[t][2].x=3, snake[t][2].y=4; snake[t][3].x=3, snake[t][3].y=2。t表示你所控制的蛇的编号，由于最多是二人对战，最多有2条蛇，即t只有两种可能取值0或1。direct[t]为一个整数，表示第t条蛇当前行进的方向。函数的返回值为一个整数，表示接下来应该行进的方向。其方向整数值的含义如表1所示

（后4个方向只有吃到斜走道具后才有效）：

表 1

行进方向	下	右	左	上	下右	上右	上左	下左
数值	0	1	2	3	4	5	6	7

结构体GamePanel表示当前游戏地图。其定义如下：

```
struct GamePanel
{
    int n;                  //地图规模，含义如上所示
    int m;                  //地图规模，含义如上所述
    int success_num;        //游戏获胜所需吃的食物数,默认为35
    int totalfoodnum;       //地图中的食物总数,每进入新的一面/关置10
    int currentfoodnum;     //当前地图中所剩食物数，食物每被吃一个该值减1, 0 ≤ currentfoodnum ≤ totalfoodnum
    point food[20];         //地图中所有食物（food[0]……food[totalfoodnum-1] ）的坐标,
    int wallnum;            //地图中障碍物（不算四周的墙，只算地图内部的障碍物）总数
    point wall[20];         //地图中所有障碍物（wall[0]……wall[wallnum-1] ）的坐标
    int speednum;           //地图中倍速道具数，第1面/关卡为0，进入第2面值变为1，被蛇吃掉后值变为0
    int speedowner;         //表示倍速道具的拥有者，值为0或1；若两方均未吃该道具，则值等于-1
    int step_speed;         //玩家吃倍速道具后，走的步数，任意蛇吃该道具后置0，此变量值不超过50
    point speedprops;       //当前地图中倍速道具的坐标
    int obliquenum;         //地图中斜走道具数，第1面/关为0，进入第2面值变为2
    int obliqueowner[5];  //斜走道具的拥有者, obliqueowner[t]=1表示蛇t拥有斜走道具,等于0表示蛇t没吃斜走道具
    point obliqueprops[2];  //当前地图中斜走道具的坐标
    int panel[50][100];     //panel[i][j]的值为0,1,2,3,4分别表示地图中点[i, j]为空,食物,障碍物,倍速道具,斜走道具
};
```

（2）check 函数

```
bool check(point snake[5][100], int len[5], int t, GamePanel gp, int direction )
```

该函数的功能是：判断在贪吃蛇当前位置和游戏地图下，下一步若以 direction 方向行进，是否可行。如可行则返回 true，如不可行（即按此方向前进会导致贪吃蛇死亡）则返回 false。参数中 snake、len、t、gp 的含义同（1），direction 表示下一步拟行进的方向，含义如表 1 所示。

这里需注意的是，当贪吃蛇能斜走的时候，类似图 3 的情形，贪吃蛇是不能向右上方通过的，无论点格 1 和 2 是障碍物还是蛇自己或对方的身体。但图 4 的情形贪吃蛇则可以向右上方通过。其他方向同理。

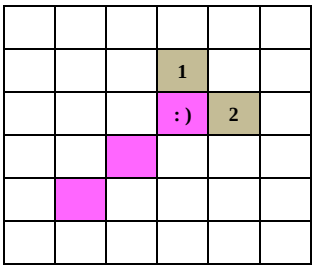

图 3

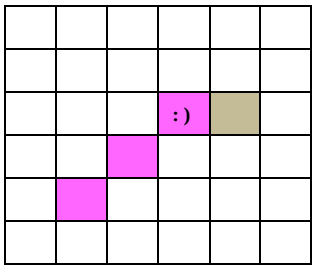

图 4

在附件 1 中，包含 2 个头文件和 1 个源文件。snake.h 包含了本程序的系统头文件、基础类型定义和一些公用函数，你的程序只允许使用这些头文件里的库函数。machine.h 文件包含操控贪吃蛇的主要逻辑，你编写的 machine_move 和 check 函数，以及其他自定义辅助函数，都须放在此头文件内。snake.cpp 是主程序，负责游戏的整体逻辑和图形显示，通过不断调用 machine_move 函数来得到贪吃蛇的前进方向，进而实现贪吃蛇的移动。程序一共包含 2 条蛇，

编号分别为 0（白蛇）和 1（粉蛇），这里默认你操控的是第 0 条蛇，当然你也可以将主程序中的变量 t 之值修改为 1，从而操控粉蛇。

提交方式：

每名同学将被分配一个编号，请大家牢记，该编号将是你在本题中的唯一标识。编号将在 QQ 群内发布。

在提交程序时，学生在自己编写的所有函数的函数名后面加一个数字，即自己的编号，无论是函数声明处还是调用处；除此之外函数名后不允许有任何其他数字。

允许定义类、结构体、全局变量，其名后也须按上述规则加入编号，但结构体、类内数据成员、函数内部定义的变量无需加编号。

例如你的编号是 5 且你编写的程序是左边形式，则需修改成右边形式：

```
void g() {}
int f() {}
int b=3;
struct C{ int d;};
bool check() { g(); }
int machine_move()
{
    int a=1;
    if (check()) f();
    return a;
}
```

```
void g5() {}
int f5() {}
int b5=3;
struct C5{ int d;};
bool check5() { g5(); }
int machine_move5()
{
    int a=1;
    if (check5()) f5();
    return a;
}
```

最后，将你的 machine.h 文件，修改为 machine+编号.h，例如你的编号是 5，则修改为 machine5.h，并在超星作业里提交该 machine5.h 头文件。

上述处理的原因是：我们会将一个班内所有同学的 machine.h 文件放入同一个项目工程里进行编译、运行和对战，所以每个同学的头文件名不能相同，且任意两个头文件中的函数名、全局变量名、类或结构体名都不能相同，否则会出现重名的编译错误。

请注意不要提交诸如“#include 头文件”、main 函数等。函数中不允许通过 printf、cout 等输出信息。machine_move 和 check 函数的参数均为值参数，对参数的修改在函数执行完毕后不会保存。

评测方法：

以比赛的方式进行评测，比赛分为两阶段，均为程序自动对战，老师提供评测程序，无需人工参与。

- 第一阶段：（1）每班内部采用双循环赛方式，即每名同学都与本班其他同学比赛 2 场，一场使用白蛇，一场使用粉蛇。如 1 个班有 31 人，则每人需进行 60 场比赛。胜一场得 3 分，平一场得 1 分，负一场得 0 分。（2）循环赛后，依据积分排出每班的名次，积分相同则所吃食物总数多者排名靠前。每班排名前 10%记 100 分，前 20%记 90 分，前 30%记 80 分，前 50%记 70 分，前 70%记 60 分，前 80%记 50 分，前 90%记 40 分，剩余能正常编译运行者记 30 分。若程序编译出错或无法正常运行，记 0 分。
- 第二阶段：诸神之战。第一阶段产生的各班冠军（计算机学院 15 个班，软件学院 10 个班）共 25 人，再进行双循环赛，决出两院总冠军。第二阶段仅供娱乐，与本课程成绩无关。
- 所有比赛的对战记录均完全公开，将以附件 2 中 rec 文件的形式发布，大家可运行、播放，看到自己每场比赛的比赛画面。

提示：

本题没有标准答案，同学们可以充分发挥想象力给出自己的解法。任何基础、任何层次的学生都有能力给出解决方案。只要程序无明显 bug，能正常编译运行，得分大于 0 分即可及格。

你的程序只需给出蛇前进的方向，至于蛇按该方向前进后身体位置如何变化，将由老师的主程序负责处理，你无需考虑，你只需重点考虑蛇头的位置。

你需要考虑如何通过从蛇头到食物的最短路径，最快吃到食物。当然如果仅考虑直线最短路径，还不够，还需考虑障碍物和蛇身，如图 5 所示的情况，如果你直接向上走，会碰到自己。再如图 6 的情况，当蛇吃掉食物后自己也就憋死了。

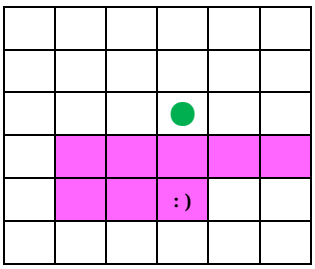

图 5

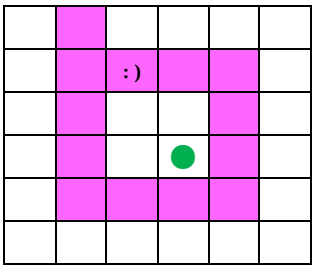

图 6

另外，当有多个食物和道具在你面前时，你如何抉择？如果你选择吃某一个，那么另一个就会被对方得到。也许你要根据对方的位置进行抉择。有的时候，道具也许是一把双刃剑，比如斜走道具，由于方向的增加，可能会增加你编程的难度。而倍速道具也只能用 50 步。当你吃的食物越多身体会越长，越难以控制。所以可能你所吃的食物数一直领先对方，但最后却由于自身太长难以控制而死亡，将胜利让给对方。甚至可以思考能否采用一些主动攻击策略，比如把对方逼入死角等。

诚信要求：

允许查阅资料、文献，同学之间可以交流思路、算法。但不允许照抄代码，老师已经收集了网络上与本题相关的所有代码作为查重对比母板，与网上代码或其他同学代码雷同者，均被视为抄袭。修改变量名或函数名、变换语句结构或函数位置等均无效。抄袭者与被抄袭者双方同论，不做区分。任何形式的破解、攻击、在提交的代码中植入恶意代码等不正当竞争行为均视为作弊。抄袭者、作弊者取消本题成绩并按情节轻重倒扣分。