

Frequently Asked Questions (FAQ)

22. 能否让蛇头变个颜色，我想知道蛇死的时候蛇头在哪。

可以在 `snake.cpp` 的第 280 行附近，即 `move` 函数的最后部分附近，`Ending(n)` 的上面，加入下列语句，这样可以在一局结束时，以数字顺序输出蛇身，蛇头即为数字 1。

```
else { printf("双方打平。"); }  
  
for (int i = 1; i <= len[t]; i++)  
{  
    GOTO(snake[t][i].x, snake[t][i].y);  
    printf("%d", i);  
}  
  
Ending(n);
```

21. 题目中说“不要提交诸如“`#include` 头文件”、`main` 函数等”具体是什么意思？

答：是指你提交的 `machine.h` 文件里，除了 `#include <snake.h>` 以外，不要再加入其它的 `#include` 头文件。

20. 我在 `check` 和 `machine_move` 函数里是否可以对函数的参数进行修改？

答：比赛时，评测程序传给你的参数都是副本，所以你可以在函数中修改参数，但这并不是一个好的编程风格，也可能会给你的程序带来潜在的风险，所以我不鼓励这样做。

19. 我一点都不会，这道题不想交了。

答：只要程序能正常编译运行，就有分，而老师给你的程序就能正常编译运行.....接下来还用我再说下去么.....

18. `panel` 数组里是否存储蛇的信息？

答：`panel` 数组里不存储蛇的位置信息。

17. `check()` 函数起什么作用？老师的评测程序里，也会调用学生编的 `check()` 函数么？

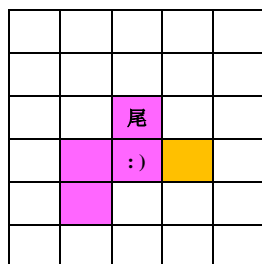
答：现在给大家的程序里，主程序调用了 `check()` 函数来判胜负。但最终老师的评测程序里，不会调用同学们编的 `check()` 函数，而是调用老师自己的 `check()` 函数来判胜负。你们编写的 `check()` 函数用于判定某方向的合法性，是你们自己的 `machine_move()` 函数的重要一环，比如最基本的，可以

FOR 方向 d=0 TO 7 DO

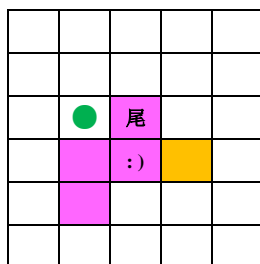
IF `check(d)`==true THEN (方向 d 是一个可行的方向.)

16. 下图的局面，蛇是否可以往左上方行进？

答：图(a)可以，图(b)不可以。



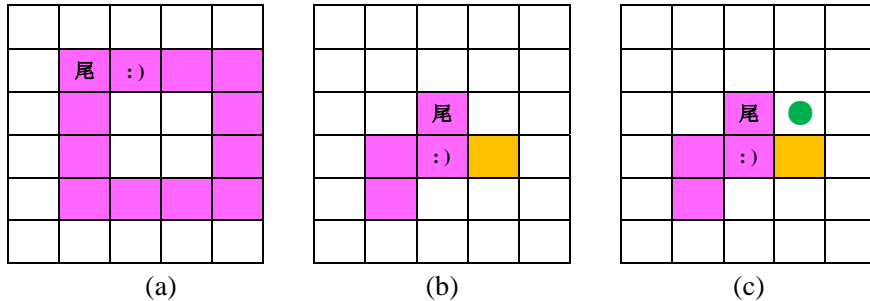
(a)



(b)

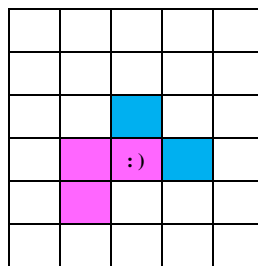
15. 蛇走到自己尾巴的位置算自杀么？

答：如下图(a)的情况，蛇可以往左走，因为蛇头往左走的同时，它的尾巴会往下移动，把蛇尾原来所占的位置让出来。同理，下图(b)的情况蛇头也可以往右上方走。但图(c)的情况，蛇不能向右上走，因为吃掉右上方的食物后，蛇身变长，蛇尾位置不变，不会让出蛇尾所占的位置。对于图(c)的情况，能不能往右上走，都能解释通，编程也都很好实现，只需要我们人为强制定一个规则即可，目前的规则是不能往右上走。



14. 像类似于上面第 15 问和第 16 问的情况判断太复杂了，可不可以简单点？

答：你可以在你的 check 函数中，使用更严厉但更简单的规则，比如下图无论蓝色的点是障碍物还是蛇神还是蛇尾，都不允许通过。亦即第 16 问的两种情况都不允许通过。这样编程简单，而且规则严格，使蛇更不容易死。



13. 头撞头怎么判定呢？

答：我们这里的规则是，主动撞的一方判负。虽然两条蛇看上去是并行走。但就像大家《操作系统》里学的进程调度一样，实际上两条蛇的移动是“并发执行”，白蛇走一步，粉蛇走一步，只不过走的很快，看起来像同时走。所以头撞头时，主动撞的一方判负。

12. 程序中如何判断一个食物被没被吃掉，比如我的蛇找到了一个食物，但是却已经被对方吃掉了？

答：可以通过 `gp.panel[gp.food[i].x][gp.food[i].y] == 1` 来判断。如果点 `x,y` 处有食物，则 `panel[x][y]==1`，当蛇把这个食物吃掉后，`panel[x][y]`就变为 0。所以可以通过下面的语句判断当前所有食物中哪些食物是没被吃的：

```
for (int i = 0; i < gp.totalfoodnum; i++)
{
    if(gp.panel[gp.food[i].x][gp.food[i].y] == 1){ food[i]即为没被吃的食物}
}
```

11. `food[]`数组、`panel[][]`数组都存了食物的信息，不重复么，它俩有什么关系？

答：给你提供两种访问食物的方式，你想用哪种都可以。`food[]`数组的内容是固定的，即便某食物被吃了，`food[]`数组里也不删除该食物。`panel`数组是动态变化的，坐标为 `x,y` 的食物被吃后，`panel[x][y]`由 1 变 0。

10. `snake[t][i]`，这里的 `t` 和 `i` 都从 1 开始吗

答：t 从 0 开始，i 从 1 开始

9. 有 2 条蛇，我控制的到底是哪条蛇？我的程序针对哪条蛇编写？

答：有 2 条蛇，第 0 条蛇（白蛇）和第 1 条蛇（粉蛇）。在对战中，你可能控制白蛇，也可能控制粉蛇，所以你的程序要有通用性，不要只针对白蛇或者只针对粉蛇。而是针对第 t 条蛇，t 为传递给 machine_move 函数的参数，可能等于 0，也可能等于 1。评测程序可能以 machine_move(0)来调用你的程序，也可能以 machine_move(1)来调用你的程序。

8. 一条蛇在行进的时候可以知道另一条蛇的位置么？

答：可以，两条蛇的位置都存储在 snake 数组里。

7. 道具用完了还有么，还可以续么？

答：倍速道具整局游戏只有 1 个，斜走道具整局游戏 2 个，吃完了就再也没有了。

6. 道具如果第 2 面没吃的话，第 3 面会把他们刷没吗？

答：不刷没，还留着。

5. 从第二面开始，障碍物的位置都不会变是吧？

答：是的。

4. 倍速道具加速原理是什么？

答：对方每走一步，你可以走两步，调用两次 machine_move 函数。这样走 50 次，每次走 2 步。

3. 一只蛇在已经吃过一个斜走道具并且还没失效的同时可以再吃一个斜走道具吗？

答：可以。这样就相当于你把这个道具抢走了，对方就没有了，也算是一种限制对方的策略。

2. panel[][]数组的列下标也是相差 2 么？

答：对。比如 panel 数组第 i 行里有意义的元素是 panel[i][0]、panel[i][2]、panel[i][4]、panel[i][6].....

1. snake.h 里的 brand()函数干什么用的，我怎么看不懂？

答：不必看懂，把它理解成 rand()用就行。