

0909Pro 뿌요뿌요

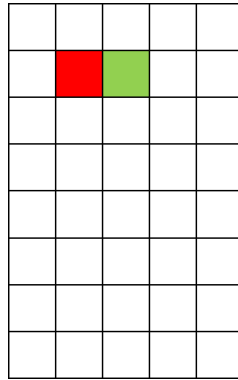
뿌요뿌요 게임을 만들어본다.

생성되는 블록은 무조건 2개로 제한되어 있고 두 블록의 색은 같거나 다를 수 있다.

1.기능명세

A. 블록 생성

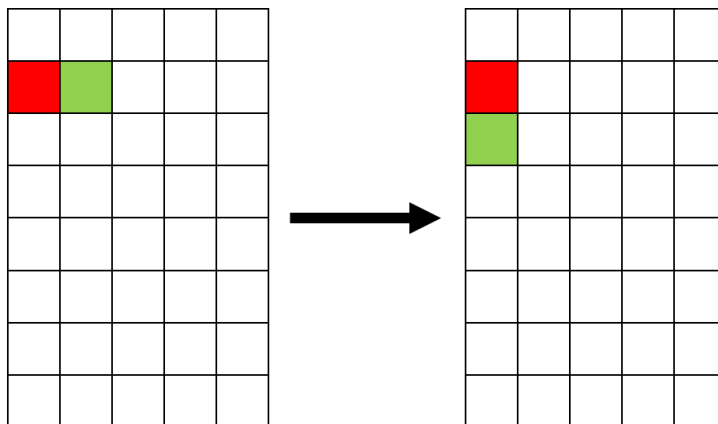
블록이 생성되면 왼쪽 끝에서 1칸 떨어진 자리에서 생성된다.



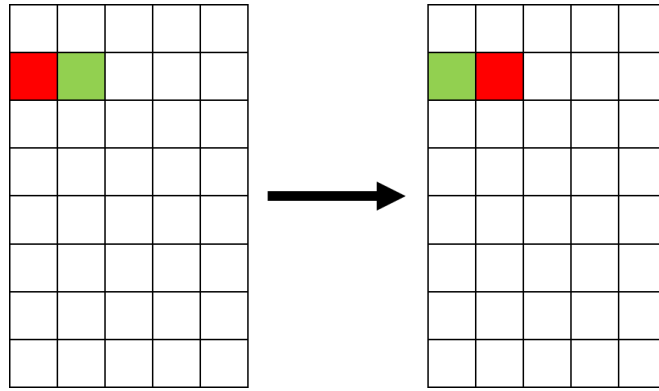
B. 회전

90 180 270 360 회전할 수 있다. 이전의 상태에서 4가지의 회전을 할 수 있다. 회전의 기준은 생성될 때의 왼쪽 블록이 기준블록이 된다. 기준블록은 변하지 않는다. 회전 할 때 블록의 위치가 좌측 또는 우측에 접촉되어 있는 경우, 블록의 위치가 이동 될 수 있다.

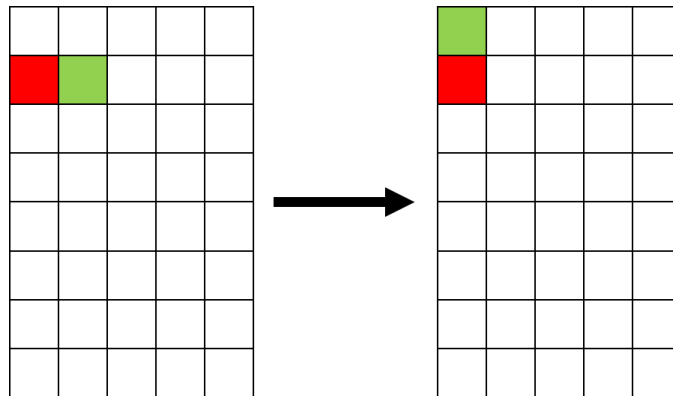
➤ 빨간색 블록이 기준일때 90도 회전



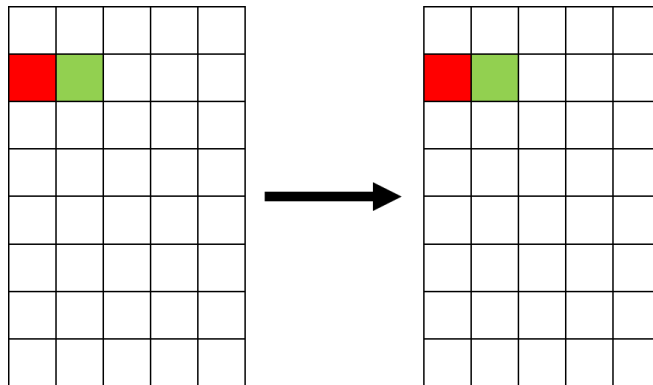
➤ 빨간색 블록이 기준일때 180도 회전(아래와 같이 벽을 넘어가기 때문에 우측으로 밀려나게 된다.)



➤ 빨간색 블록이 기준일때 270도 회전



➤ 빨간색 블록이 기준일때 360도 회전



C. 이동

좌 우로 이동 할 수 있으며 벽을 넘어 가지 못한다.

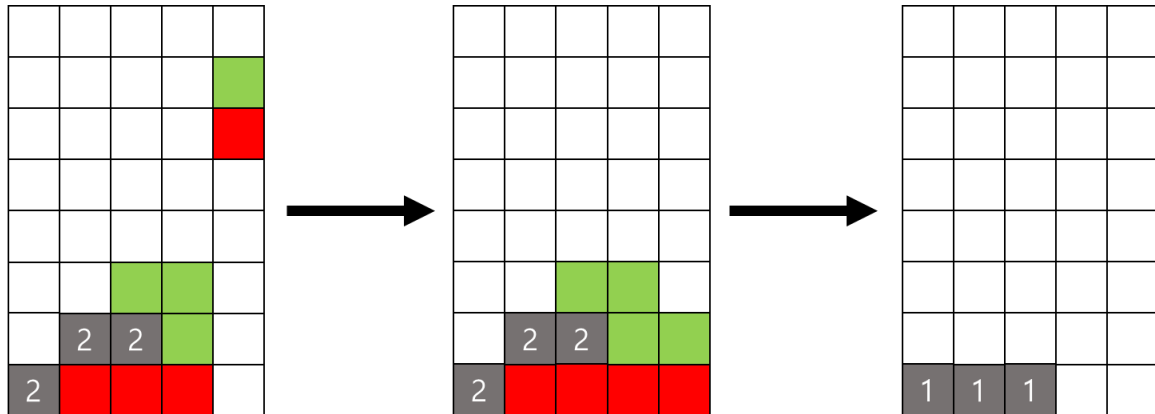
D. 내리기

블록을 내려서 반영한다. 반영하였을 때 인접블록이 4개이상인경우 사라지게 되고 사라진 빈자리는 위에서부터 내려와서 모두 채우게 된다. 채운 후 또 인접블록이 4개이상인경우가 있다면 연쇄적으로 사라지게 된다. 한번에 여러 종류의 블록이 동시에 사라질 수도 있다.

E. 방해블럭

방해 블록은 생명이 존재하는 색이 없는 블록이다. 4개 이상이 인접하여도 사라지지 않으며, 방해블록과 인접한 일반 블록이 4개이상인접하여 사라질 때 생명이 1씩 감소한다. 0이되는경우 방해블록은 사라진다.

아래와 같이 블록을 내려서 두가지의 블록이 사라지는 경우에도 방해블록의 생명은 1만 감소한다.



2. 함수설계

void init(int width)	맵과 데이터를 초기화한다.
void newBlock(int* block)	새로운 블록을 생성한다. 매개변수로 [0]과 [1]에 두 블록의 색상 숫자를 전달한다. 가장 좌측에서 한 칸 떨어진 자리에서 생성된다.
void newHinderBlock(int count, int life)	새로운 방해블록을 생성한다. 좌측0번째부터 count개수만큼 방해블록이 생성된다. 방해블록은 이동하지 못한다.
void moveBlock(int distance)	생성된 일반블록을 좌우로 이동한다. 매개변수값만큼 이동한다. 매개변수값이 0보다작으면 좌로이동하고 0보다 크면 우로이동한다. 0이면 이동하지않는다.
void rotateBlock(int angle)	블록을 회전한다. angle 값에 따라 회전한다. 1 : 90 2 : 180 3 : 270 4 : 360
int landBlock()	생성된 일반 블록 또는 방해블록을 내린다. 내린 후 제거할 것이 있으면 제거 하고 연쇄적인 제거도 모두 이루어진다. 더 이상 제거할 것이 없는 상태로 만들고 난 뒤 최대 높이를 리턴한다.

3.제약사항

- Height 는 600 이다.
- Width 는 4~10 이다.
- 생성된 블록에 대해 move 또는 rotate는 최대 3번까지만 요청한다.
- Move는 -10 ~ 10 사이로 요청한다.
- 방해블록의 생명은 1~5이다.
- 생성 횟수는 10 ~ 300번이다.
- 블록 색상숫자는 1~5이다.

Main code

```
#define HINDERBLOCK 9
#define MOVE 200
#define ROTATE 300

#include <stdio.h>

extern void init(int width);
extern void newBlock(int* block);
extern void newHinderBlock(int count, int life);
extern void moveBlock(int distance);
extern void rotateBlock(int angle);
extern int landBlock();

int main() {
    int tc;
    int totalscore = 0;
    //freopen("input.txt", "r", stdin);
    scanf("%d", &tc);
    for (int testcase = 1; testcase <= tc; testcase++) {
        //init
        int width;
        int blockCount;
        int score = 100 / tc;
        scanf("%d", &width);
        init(width);
        scanf("%d", &blockCount);

        for (int i = 0; i < blockCount; i++) {
            int block[2];
            scanf("%d %d", &block[0], &block[1]);
            if (block[0] == HINDERBLOCK) {
```

```

        int life;
        scanf("%d", &life);
        newHinderBlock(block[1], life);
    }
    else {
        newBlock(block);
        int cmdCount;
        scanf("%d", &cmdCount);
        for (int j = 0; j < cmdCount; j++) {
            int cmd, value;
            scanf("%d %d", &cmd, &value);
            if (cmd == MOVE) {
                moveBlock(value);
            }
            else if (cmd == ROTATE) {
                rotateBlock(value);
            }
        }
    }
    int result = landBlock();
    int answer;
    scanf("%d", &answer);
    if (answer != result) {
        score = 0;
    }
}
printf("#%d : %d\n", testcase , score);
totalscore += score;

}

printf("totalScore : %d\n", totalscore);

return 0;

}

```

Usercode(구현)

```

#define ROTATE_90 1
#define ROTATE_180 2
#define ROTATE_270 3
#define ROTATE_360 4
#include <iostream>
#include <string>
using namespace std;
void init(int width) {

}
void newBlock(int* block) {

}
void newHinderBlock(int count, int life) {

```

```
}  
void moveBlock(int distance) {  
  
}  
void rotateBlock(int angle) {  
  
}  
int landBlock() {  
    return -1;  
}
```