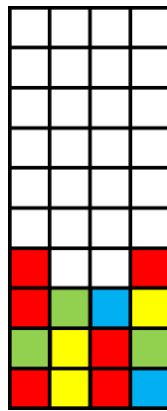


# 옥타팡!

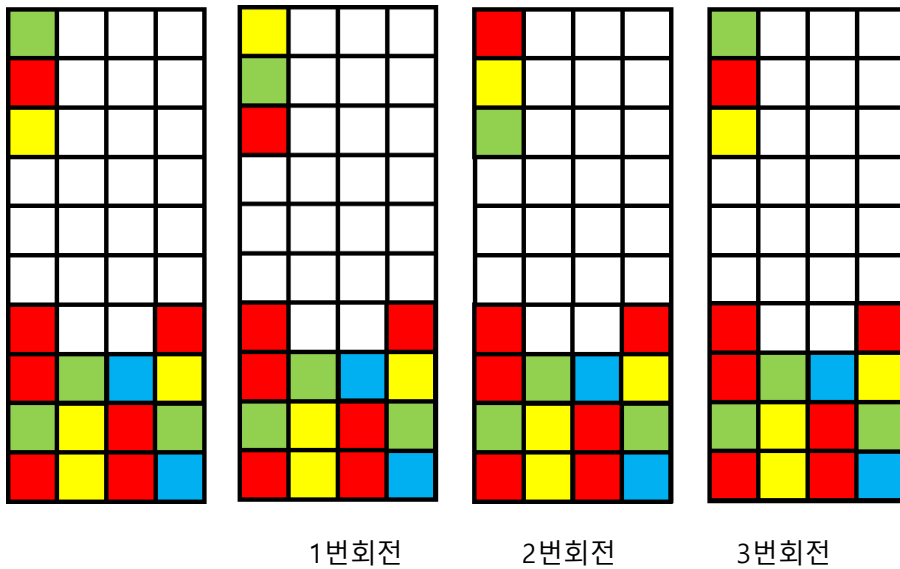
병륜이는 컴퓨터소프트웨어공학전공 학생이다. 이번 방학에 큰 마음을 먹고 게임을 한번 만들어 볼 계획이다. 하지만 창의적이기 못하기 때문에 이미 나온 게임을 살짝 바꿔서 개발할 생각을 하고있다. 게임의 규칙은 아래와 같다.

아래와 같이 맵이 있을 때 블록을 잘 놓아서 같은색의 블록을 제거 하는 게임이다.



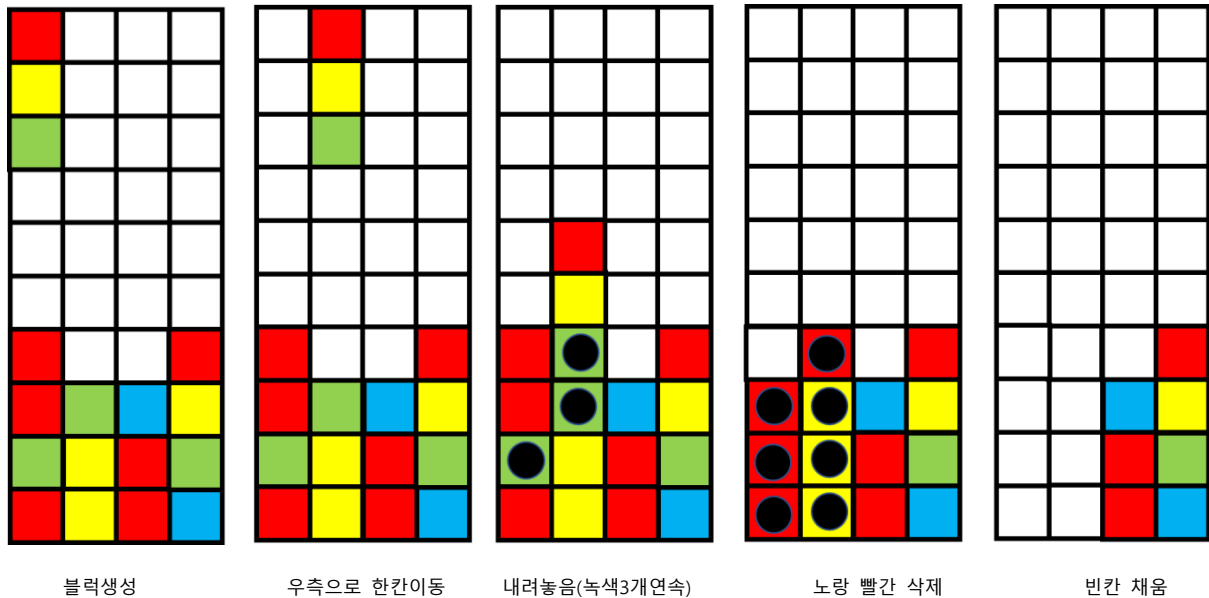
블록은 항상 세로로 3칸 주어진다. 이 블록은 좌우 이동, 블록 상하회전이 가능하다.

좌우이동은 블록을 내려놓기 전에 이동할 수 있다. 상하 회전은 아래와 같다.

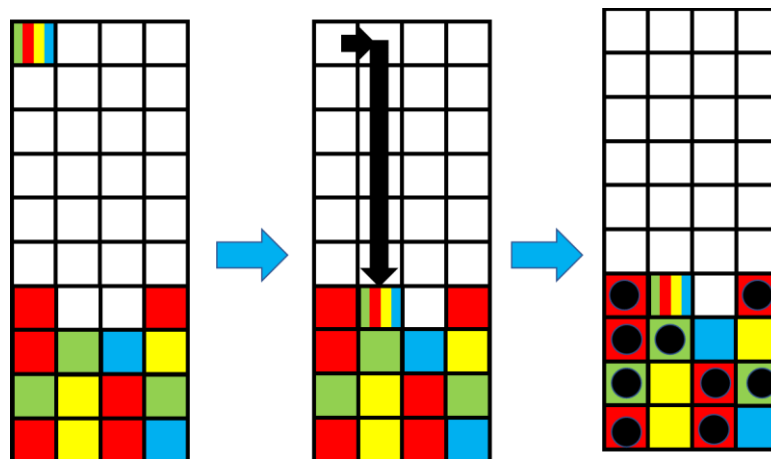


블록을 내려 놓게 되면 8방향으로 같은색이 3개이상 연속된다면 모두 터트린다. 해당하는 모든 블록을 동시에 터트린후 빈공간을 위의 블록들이 채운다. 채우고 난 뒤 또 같은색이 3개이상 연속으로 존재한다면 연쇄적으로 터트린다.

아래 예를 보자.



이 게임에 추가적으로 스페셜 블록을 넣어 게임의 재미를 추가하려고한다. 스페셜블록은 1칸이며 1칸이기때문에 상하회전은 사용할 수 없고 좌우이동만 사용할 수 있다. 스페셜블록을 내려놓게 되면 상하좌우 인접한 블록과 같은색의 블록은 맵에서 모두 사라진다.



터트린후 빈공간을 위의 블록들이 채운다. 채우고 난 뒤 또 같은색이 3개이상 연속 존재한다면 연쇄적으로 터트린다.

위의 게임을 만들기위해 병륜이는 설계를 아래와 같이 했다. 설계를 한 후 게을러진 병륜이는 개

발을 포기했다 개발을 마무리 해주자.

함수명	기능	제한
void init(int width)	게임 맵의 가로크기가 주어진다. 각 TC 초기에 한번 호출된다.	Width : 4~10
void setBlock(int* block)	블록을 생성한다 int[3] 를 매개변수로 받는다.	블록의 색상은 1~6의 숫자로 표현한다.
void setSpecialBlock()	스페셜 블록을 생성한다.	무조건 1칸이다.
void rotation(int repeat)	세로로 블록을 repeat만큼 회전한다.	스페셜 블록이 생성된상태라면 호출되지않는다.
void replace(int position)	블록을 좌우로 이동한다.	맵의 크기에 벗어난 위치일 경우 벅을 넘어가지 않는다.
int land()	블록을 내리고 남아있는 블록의 개수를 리턴한다.	

추가 사항 :

맵의 크기 height 최대 900,  $4 \leq \text{width} \leq 10$

블록이 쌓여서 맵의 크기 이상으로 쌓이지 않게 input이 주어진다.

3개 연속이 되어서 사라질 블록들은 한번에 다같이 사라진다.

(여러가지 색의 블록들이 삭제대상이 될 수 있는데 동시에 사라진다.)

```
#define INIT 1
#define SET_BLOCK 2
#define SET_SPEICAL_BLOCK 3
#define ROTATIOIN 4
#define REPLACE 5
#define LAND 6
#define EXIT 7

#include<stdio.h>

extern void init(int width);
extern void setBlock(int* block);
extern void setSpecialBlock();
extern void rotation(int repeat);
extern void replace(int position);
extern int land();

int main() {
    freopen("input.txt", "r", stdin);

    int order, n, b[3];
    while (1) {
        scanf("%d", &order);
        if (order == EXIT)
            break;
        switch (order) {
            case INIT:
                scanf("%d", &n);
```

```

        init(n);
        break;
    case SET_BLOCK:
        scanf("%d %d %d", &b[0], &b[1], &b[2]);
        setBlock(b);
        break;
    case SET_SPEICAL_BLOCK:
        setSpecialBlock();
        break;
    case ROTATIOIN:
        scanf("%d", &n);
        rotation(n);
        break;
    case REPLACE:
        scanf("%d", &n);
        replace(n);
        break;
    case LAND:
        printf("%d\n", land());
        break;
    }
}
return 0;
}

```

```

/*
You should use the standard input/output
in order to receive a score properly.

Do not use file input and output

Please be very careful.
*/

import java.util.Scanner;

/*
As the name of the class should be Solution , using Solution.java as the filename is
recommended.
In any case, you can execute your program by running 'java Solution' command.
*/
class Solution {
    static int Answer;
    static final int INIT = 1;
    static final int SET_BLOCK = 2;
    static final int SET_SPEICAL_BLOCK = 3;
    static final int ROTATIOIN = 4;
    static final int REPLACE = 5;
    static final int LAND = 6;
    static final int EXIT = 7;

    public static void init(int width){

    }
    public static void setBlock(int[] block){

    }
    public static void setSpecialBlock(){

    }
    public static void rotation(int repeat){

    }
    public static void replace(int position){

    }
    public static int land(){

        return 0;
    }
}

```

```

    }

    public static void main(String args[]) throws Exception {
        /*
            The method below means that the program will read from input.txt,
            instead of standard(keyboard) input.
            To test your program, you may save input data in input.txt file,
            and call below method to read from the file when using nextInt() method.
            You may remove the comment symbols(//) in the below statement and use
            it.
            But before submission, you must remove the freopen function or rewrite
            comment symbols(//).
        */

        /*
            Make new scanner from standard input System.in, and read data.
        */
        Scanner sc = new Scanner(System.in);
        //Scanner sc = new Scanner(new FileInputStream("input.txt"));

        int T = sc.nextInt();
        int order, n;
        int[] b = new int[3];
        while (true) {
            order = sc.nextInt();
            if (order == EXIT)
                break;
            switch (order) {
                case INIT:
                    n = sc.nextInt();
                    init(n);
                    break;
                case SET_BLOCK:
                    b[0] = sc.nextInt();
                    b[1] = sc.nextInt();
                    b[2] = sc.nextInt();
                    setBlock(b);
                    break;
                case SET_SPEICAL_BLOCK:
                    setSpecialBlock();
                    break;
                case ROTATIOIN:
                    n = sc.nextInt();
                    rotation(n);
                    break;
                case REPLACE:
                    n = sc.nextInt();
                    replace(n);
                    break;
                case LAND:
                    System.out.println(land());
                    break;
            }
        }
    }
}

```

input	output
1 4	3
2 1 2 3	6
6	6
2 2 4 3	9
5 1	3
4 2	6
6	2

2 1 1 2	5
5 1	8
6	7
2 5 4 3	10
6	13
3	16
5 -2	19
6	4
2 3 2 1	
6	
3	
5 1	
6	
2 1 1 3	
5 -1	
6	
2 3 2 5	
5 3	
6	
3	
5 -1	
6	
2 3 2 1	
6	
2 5 5 4	
5 -1	
6	
2 4 5 4	
6	
2 5 3 6	
6	
3	
6	
7	