사각형 많이 놓기

# 목표

판 위에 최대한 많은 사각형을 놓는다!

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 3 | 3 | 0 | 0 |  |  |  |
| 1 | 1 | 3 | 3 | 0 | 0 |  |  |  |
| 2 | 2 | 0 | 0 | 0 | 0 | 0 |  |  |
| 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 0 |  |  |  | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 7 | 7 | 0 | 0 | 0 | 5 | 5 | 0 |
| 4 | 17 | 13 | 0 | 0 | 8 | 13 | 14 | 9 |
| 4 | 10 | 6 | 0 | 0 | 8 | 8 | 9 | 9 |

표와 같이 9 x 9 의 크기를 갖는 판이 있다.

표에서 처럼 2 x 2 의 크기를 갖는 사각형을 놓는다.

(표에서 검정색으로 칠해진 칸은 **‘죽은칸’**이다.)

# 규칙

## 사각형 규칙

사각형의 크기는 모두 2 x 2로 동일하다.

사각형의 내부는 동일한 숫자로 채워져 있다.

사각형의 숫자는 1~9의 범위를 가진다.

사각형들은 겹쳐놓을 수 있다.

겹쳐놓을 경우 표에서 주황색으로 표시된 부분처럼 안의 숫자가 합쳐진다.

**‘죽은칸’**에는 사각형을 겹쳐놓을 수 없으며, **‘죽은칸’**에 겹쳐놓을 경우 해당 사각형이 놓인 칸이 모두 **‘죽은칸’**이 된다.

인접한 칸이 모두 **‘죽은칸’**인 곳에는 사각형을 놓을 수 없다.

(예시 표에서의 파란색 부분 – 검정칸은 모두 죽은칸)

**더 이상 사각형을 놓을 수 없게되면 프로그램을 종료한다.**

## 위험과 죽음 규칙

이웃한 숫자와의 차가 10 이상이면서 2배 이상이 되면 숫자가 **큰** 칸은 *‘위험’* 상태에 놓이게 된다.

[if m-n>=10 && m>=2\*n then n.danger = true; end if]

*‘위험’* 상태에서 작은 쪽에 사각형을 겹쳐서 차이가 줄거나 2배가 안되게 되면 *‘위험’* 상태는 해제된다.

인접한 4면이 모두 *‘위험’* 또는 **‘죽은칸’** 상태가 되면 해당 칸은 **‘죽은칸’** 이 된다.

**‘죽은칸’**에는 더 이상 사각형을 겹쳐놓을 수 없으며, 죽은셀과 인접한 면은 모두 *‘위험’* 상태가 된다.

판을 벗어난 영역은 모두 **‘죽은칸’** 으로 간주한다.

※주의 – 판을 벗어난 영역을 모두 죽은칸으로 할 경우, 인접한 부분이 ‘위험’상태가 되어 판의 모서리 부분은 죽게됨!!

표의 노란 부분처럼 위아래 혹은 좌우의 칸이 모두 **‘죽은칸’**이면 이 칸도 **‘죽은칸’**이 된다.

직전 16번 동안 놓았던 좌표에 사각형을 놓을 경우 해당 사각형이 놓인 칸은 모두 **‘죽은칸’**이 된다.

# 데이터

## 입력

입력되는 숫자는 1에서 9까지 증가와 감소를 반복한다. (따로 파일이나 스트림 형태로 입력되지 않는다.)

**1** 2 3 4 5 6 7 8 **9** 8 7 6 5 4 3 2 **1** 2 3 4 5 6 7 …

주어지는 사각형은 5개까지 가지고 있을 수 있고, 들고있는 5개 사각형 중 필요한 것을 골라서 놓을 수 있다. (buffer size = 5)

처음 1~5를 가지고 시작하며, 6에서부터 입력된다.

*(※주의 - 부정한 값이 입력될 경우 해당 값은 무시됩니다)*

## 출력

사각형을 놓는 좌표를 [숫자n x좌표 y좌표] 순서로 아래와 같이 txt 파일로 출력한다.

표의 가로를 x축 세로를 y축이라 한다.

표의 좌상단은 x=0, y=0 이다.

표의 우하단으로 갈수록 x, y 값이 증가한다.

|  |
| --- |
| n x y  1 0 0  2 2 0  3 5 7  5 7 7 |

# 실행 및 컴파일 환경

실행시간은 최대 10분을 넘겨서는 안된다. 10분이 넘을 경우 프로그램은 중지하고, 그 때까지의 결과물로 채점한다. (문제의 내용상 10분 제한은 지나치게 오래 실행하는 것을 막기 위한 방편일 뿐, 크게 의미를 갖는 수치는 아님)

C, C++, Objective-C, C#, Java, Python 사용이 가능하며, 외부 라이브러리 사용에 제한은 없다.

Windows 7 또는 OS X 10.9.4 환경에서 소스코드를 빌드하고 실행할 수 있어야 한다.

컴파일러 또는 인터프리터, VM 등의 실행환경은 Visual Studio, Xcode 를 제외하고 회사에서 사용시 라이선스 문제가 없는 제품이어야 한다.

제출하는 소스 코드에 사용한 모든 외부 라이브러리를 포함해야 한다.

즉시 빌드가 가능하도록 프로젝트 파일이나 Makefile 등을 포함해야 한다.

**점수 부여 & 평가**

사용된 블록의 숫자 총합을 기준으로 ‘상대평가’ 한다.

빌드 및 실행 여부에 따른 기본 점수가 존재한다!

출력파일 검증 중에 잘못된 사각형(가지고 있지 않은 숫자, 놓을 수 없는 위치 등)의 경우 해당 사각형을 놓는 자리(최대 4칸)를 모두 **‘죽은칸’**으로 처리하며 해당 숫자는 합산에서 제외된다.

출력파일에 심각한 문제가 있을 경우 해당 시점까지 검증된 결과를 기준으로 한다.

합산되는 사각형이 1만개 이상일 경우 무한 반복되는 것으로 판단하여 만점으로 처리한다.