



道案内 2 (Navigation 2)

JOI 王国は周囲をすべて海に囲まれた島であり、縦 N 行、横 N 列のマスマ目状に区切られた正方形の形をしている。JOI 王国の縦方向は南北方向に平行であり、横方向は東西方向に平行である。北から $r + 1$ 行目 ($0 \leq r \leq N - 1$)、西から $c + 1$ 列目 ($0 \leq c \leq N - 1$) にあるマスを取画 (r, c) と呼ぶ。

JOI 王国の女王である Anna は、Bruno をパーティーに招待することになった。Anna はパーティーの会場を決めることになり、会場の候補として $K (= 7)$ 個の区画が選ばれた。候補となった区画には 0 から $K - 1$ までの候補地番号が割り当てられており、候補 i ($0 \leq i \leq K - 1$) は区画 (R_i, C_i) である。ただし、どの候補の区画も海に接することはない。

パーティー会場となる区画は、パーティーの当日に決定される。

パーティーの前日、Anna は、Bruno が道に迷わずにパーティーの会場にたどり着けるようにするために、すべての区画に 1 本ずつ整数が書き込まれた旗を立てることにした。Anna は、それぞれの旗に、 1 以上 $1\,000\,000\,000$ 以下の整数を 1 個ずつ書き込む。

パーティーの当日、Bruno にはパーティー会場の候補地番号 t ($0 \leq t \leq K - 1$) のみが知らされる。その後、Bruno は海に接しない区画のいずれかにヘリコプターで到着し、そこからパーティー会場まで移動する。

Bruno は自分が現在いる区画の場所は分からない。しかし、Bruno は東西南北の方位は分かる。また、自分が現在いる区画と、その周囲 8 マスの区画に立てられた旗のみを見ることができる。すなわち、Bruno が区画 (a, b) ($1 \leq a \leq N - 2, 1 \leq b \leq N - 2$) にいるとき、Bruno が見ることができる旗は 9 個の区画

$$(a - 1, b - 1), (a - 1, b), (a - 1, b + 1), (a, b - 1), (a, b), (a, b + 1), (a + 1, b - 1), (a + 1, b), (a + 1, b + 1)$$

の旗のみである。Bruno は以下の 5 種類の行動のいずれかを取ることができる。

- 行動 0 : 東方向に 1 区画分移動する。すなわち、区画 (a, b) から区画 $(a, b + 1)$ に移動する。
- 行動 1 : 西方向に 1 区画分移動する。すなわち、区画 (a, b) から区画 $(a, b - 1)$ に移動する。
- 行動 2 : 南方向に 1 区画分移動する。すなわち、区画 (a, b) から区画 $(a + 1, b)$ に移動する。
- 行動 3 : 北方向に 1 区画分移動する。すなわち、区画 (a, b) から区画 $(a - 1, b)$ に移動する。
- 行動 4 : 現在いる区画でパーティーが行われると判断し、移動を終える。

パーティーに遅刻することは許されないため、Bruno はパーティー会場まで最小の行動回数で移動する。したがって、Bruno は海に接した区画に立ち入ることはないことがこの問題の条件下で保証される。

旗に大きな整数を書き込むのは大変なので、Anna は旗に書き込む整数の最大値をできるだけ小さくしたい。

旗に整数を書き込む Anna の戦略を実装したプログラムおよび、最小の行動回数でパーティー会場にたどり着くための Bruno の戦略を実装したプログラムを作成せよ。



実装の詳細

あなたは2つのファイルを提出しなければならない。

1つ目のファイルは `Anna.cpp` という名前である。このファイルは `Anna` の戦略を実装したファイルであり、以下の関数を実装していなければならない。また、`#include` プリプロセッサ指令によって `Anna.h` を読み込むこと。

- `void Anna(int N, int K, std::vector<int> R, std::vector<int> C)`

これは、旗に整数を書き込む `Anna` の戦略を実装した関数である。この関数は、各シナリオ (詳しくは採点の方法の項を参照) について最初に1回だけ呼び出される。

- 引数 `N` は、JOI 王国の縦方向および横方向のマス目の個数を表す。
- 引数 `K` は、パーティー会場の候補の数 $K (= 7)$ である。
- 引数 `R`, `C` は長さ K の配列であり、`R[i]`, `C[i]` は候補 i の区画 (R_i, C_i) を表す $(0 \leq i \leq K - 1)$ 。
- 引数 `N`, `K`, `R[i]`, `C[i]` $(0 \leq i \leq K - 1)$ の値の範囲については、制約の項を参照せよ。

関数 `Anna` の1回の呼び出しにおいて、以下の関数を、それぞれの区画に対して1回ずつ、合計 N^2 回呼び出さなければならない。

- `void SetFlag(int r, int c, int value)`

- 引数 `r`, `c` は、`Anna` が区画 (r, c) の旗に整数を書き込むことを表す。ここで、 $0 \leq r \leq N - 1$, $0 \leq c \leq N - 1$ でなければならない。この範囲外の値で関数を呼び出した場合、不正解 [1] と判定される。
- 引数 `value` は、`Anna` が指定した旗に書き込む整数である。ここで、 $1 \leq \text{value} \leq 1\,000\,000\,000$ でなければならない。この範囲外の値で関数を呼び出した場合、不正解 [2] と判定される。
- 関数 `SetFlag` を同じ (r, c) で2回呼び出した場合、不正解 [3] と判定される。
- 関数 `Anna` の実行の終了時に関数 `SetFlag` の呼び出し回数が N^2 回でなかった場合、不正解 [4] と判定される。

関数 `SetFlag` の呼び出しが不正解と判定された場合、その時点でプログラムは終了する。



2つ目のファイルは `Bruno.cpp` という名前である。このファイルは Bruno の戦略を実装したファイルであり、以下の関数を実装していなければならない。また、`#include` プリプロセッサ指令によって `Bruno.h` を読み込むこと。

• `std::vector<int> Bruno(int K, std::vector<int> value)`

この関数では、引数に対して、Bruno が取る行動を提示しなければならない。この関数は、各シナリオ (詳しくは 採点の方法 の項を参照) について、関数 Anna が呼び出された後に 1 回だけ呼び出される。

- 引数 K は、パーティー会場の候補の数 $K (= 7)$ である。
- 引数 `value` は長さ 9 の配列であり、Bruno が現在いる区画と、その周囲 8 マスの区画に立てられた旗に書かれた整数を表す。具体的には、Bruno が現在いる区画を (a, b) ($1 \leq a \leq N - 2$, $1 \leq b \leq N - 2$) とすると、`value[0]`, `value[1]`, ..., `value[8]` はそれぞれ区画

$(a-1, b-1)$, $(a-1, b)$, $(a-1, b+1)$, $(a, b-1)$, (a, b) , $(a, b+1)$, $(a+1, b-1)$, $(a+1, b)$, $(a+1, b+1)$

の旗に書かれた整数を表す。

- 関数 Bruno は、パーティー会場が候補 $t = 0, 1, 2, \dots, K - 1$ になった場合それぞれについて、Bruno が取る行動を決定しなければならない。戻り値は長さ K の配列で、その $i + 1$ 番目 ($0 \leq i \leq K - 1$) の要素は $t = i$ である場合に Bruno が取る行動の番号となっている必要がある。
- 戻り値が長さ K の配列ではない場合、不正解 [5] と判定される。
- 戻り値の配列のすべての要素は 0, 1, 2, 3, 4 のいずれかでなければならず、ひとつでもそれ以外の値となっていた場合、不正解 [6] と判定される。
- どの t に対しても、関数 Bruno で指定した行動が、最小の行動回数でパーティー会場に移動するものでなければならない。特に、行動 4 を取る場合は、Bruno が現在いる区画がパーティー会場の区画と同じでなければならない。ひとつでもこれを満たさない場合は、不正解 [7] と判定される。また、最小の行動回数で会場に移動する方法が複数ある場合は、それらのうちどれを指定してもよい。

この問題では、ひとつのテストケースは Q 個のシナリオからなり、各シナリオにつき関数 Anna と関数 Bruno が 1 回ずつ呼び出される。すなわち、全体では関数 Anna と関数 Bruno が交互に、合計 Q 回ずつ呼び出されることになる。詳しくは採点の方法の項を参照せよ。



重要な注意

- 内部での使用のために他の関数を実装したり、グローバル変数を宣言するのは自由である。ただし、提出された2つのプログラムは、採点プログラムとまとめてリンクされて1つの実行ファイルになるので、各ファイル内のすべてのグローバル変数と内部関数を無名名前空間内で宣言して、他のファイルとの干渉を避ける必要がある。採点時には、このプログラムは Anna 側, Bruno 側として2個のプロセスとして実行されるので、Anna 側と Bruno 側でプログラム中のグローバル変数を共有することはできない。
- あなたの提出したプログラムは、標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。ただし、標準エラー出力にデバッグ情報等を出力することは許される。

採点の方法

1つのテストケースは Q 個のシナリオからなり、シナリオには 0 から $Q-1$ までの番号が付けられている。各シナリオに対して、以下の値が定められている。ただし、以下に記された値の範囲については、制約の項を参照せよ。

- JOI 王国の縦方向および横方向のマス目の個数 N 。
- パーティー会場の候補の数 $K (= 7)$ 。
- パーティー会場の候補の区画 $(R_0, C_0), (R_1, C_1), \dots, (R_{K-1}, C_{K-1})$ 。
- Bruno が現在いる区画 (a, b) 。

関数 Anna はそれぞれのシナリオに対して実行され、与えられた引数に対して旗に整数を書き込まなければならない。関数 Bruno もまた、それぞれのシナリオに対して実行され、与えられた引数に対して Bruno が取る行動を決定しなければならない。ここで、以下のような手順で関数 Anna と関数 Bruno が呼び出される。

1. $k = 0, 1, 2, \dots, Q-1$ の順に、以下の処理 2 と処理 3 をこの順に行う。
2. 関数 Anna が呼び出される。引数は、シナリオ k について実装の詳細の項に書かれた通りに設定される。
3. 関数 Bruno が呼び出される。引数は、シナリオ k について実装の詳細の項に書かれた通りに設定される。

ただし、これらの処理の途中で不正解と判定された場合、その時点で採点は終了し、そのテストケースは不正解となる。



コンパイル・実行の方法

作成したプログラムをテストするための、採点プログラムのサンプルが、コンテストサイトからダウンロードできるアーカイブの中に含まれている。このアーカイブには、提出しなければならないファイルのサンプルも含まれている。

採点プログラムのサンプルは1つのファイルからなる。そのファイルは `grader.cpp` である。作成したプログラムをテストするには、`grader.cpp`, `Anna.cpp`, `Bruno.cpp`, `Anna.h`, `Bruno.h` を同じディレクトリに置き、次のようにコマンドを実行する。

```
g++ -std=gnu++17 -O2 -fsigned-char -o grader grader.cpp Anna.cpp Bruno.cpp
```

コンパイルが成功すれば、`grader` という実行ファイルが生成される。

実際の採点プログラムは、採点プログラムのサンプルとは異なることに注意すること。採点プログラムのサンプルは単一のプロセスとして起動する。このプログラムは、標準入力から入力を読み込み、標準出力に結果を出力する。

採点プログラムのサンプルの入力

採点プログラムのサンプルは標準入力から以下の形式で入力を読み込む。ただし、入力はすべて整数でなければならない。

```
Q  
(シナリオ 0 に対する入力)  
:  
(シナリオ  $Q-1$  に対する入力)
```

また、それぞれのシナリオに対する入力は、以下の形式で与えられる。

```
N K  
R0 C0  
:  
RK-1 CK-1  
a b
```

ただし、採点プログラムのサンプルの入力には、 N は $3 \leq N \leq 100$ の範囲で、 K は $1 \leq K \leq 7$ の範囲で指定できる。これは実際の制約と異なることに注意せよ。



採点プログラムのサンプルの出力

採点プログラムのサンプルは標準出力へ以下の情報を出力する (引用符は実際には出力されない).

- 正解の場合, 関数 `Anna` で旗に書き込まれた整数の最大値が “Accepted : Maximum value = 12” のように出力される.
- 不正解の場合, 不正解の種類が “Wrong Answer [1]” のように出力される.

実行するプログラムが複数の不正解の条件を満たした場合, 表示される不正解の種類はそれらのうち 1 つのみである.

制約

- $1 \leq Q \leq 300$.
- $5 \leq N \leq 100$.
- $K = 7$.
- $1 \leq R_i \leq N - 2$ ($0 \leq i \leq K - 1$).
- $1 \leq C_i \leq N - 2$ ($0 \leq i \leq K - 1$).
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq K - 1$).
- $1 \leq a \leq N - 2$.
- $1 \leq b \leq N - 2$.



採点基準

この課題におけるテストケースに対して、1つでも不正解があった場合、この課題の得点は0点となる。

また、この課題におけるすべてのテストケースに正解した場合、この課題のすべてのテストケースに対する「旗に書かれた整数の最大値」を L として、この課題の得点は以下のように与えられる。

- $70\,001 \leq L \leq 1\,000\,000\,000$ のとき、7点。
- $10\,001 \leq L \leq 70\,000$ のとき、13点。
- $2\,001 \leq L \leq 10\,000$ のとき、19点。
- $21 \leq L \leq 2\,000$ のとき、 $50 - 12.5 \times \log_{10} \left(\frac{L}{20} \right)$ 点を整数に切り捨てた得点。

また、 $L \leq 20$ の場合は、以下の表の通りに得点が与えられる。

L	20	19	18	17	16	15	14	13	12 以下
得点	50	53	56	60	64	69	75	85	100

やりとりの例

採点プログラムのサンプルが読み込む入力の例と、それに対応する関数の呼び出しの例を以下に示す。ただし以下の例は、Anna が関数 `SetFlag` で、25本の旗に以下に示すような値を書き込んだ場合である。

入力例 1	Anna が旗に書き込む整数の例
1	47 15 63 56 71
5 7	10 46 52 18 67
1 1	63 56 71 19 48
1 2	52 18 67 99 26
2 1	71 19 48 60 89
2 2	
2 3	
3 2	
3 3	
1 1	



Anna の呼び出し	Bruno の呼び出し	Bruno の戻り値
Anna(5, 7, [1, 1, 2, ..., 3], [1, 2, 1, ..., 3])		
SetFlag(0, 0, 47)		
SetFlag(0, 1, 15)		
SetFlag(0, 2, 63)		
⋮		
SetFlag(4, 4, 89)		
	Bruno(7, [47, 15, 63, ..., 71])	[4, 0, 2, 2, 2, 0, 0]

この例では、 $(a, b) = (1, 1)$ が指定されている。ここで、例えば候補 0, 1, 2, 3 がパーティー会場として選ばれた場合、それぞれ次のような行動を取らなければならない、関数 Bruno の戻り値はその通りにしなければならない。

- 候補 0 が選ばれた場合、パーティー会場は区画 (1, 1) となり、行動 4 を取らなければならない。
- 候補 1 が選ばれた場合、パーティー会場は区画 (1, 2) となり、行動 0 を取らなければならない。
- 候補 2 が選ばれた場合、パーティー会場は区画 (2, 1) となり、行動 2 を取らなければならない。
- 候補 3 が選ばれた場合、パーティー会場は区画 (2, 2) となり、行動 0 または 2 を取らなければならない。

この例では $[4, 0, 2, 2, 2, 0, 0]$ を返しているが、最小の行動回数でパーティー会場に移動する方法が複数存在することもある。例えば $[4, 0, 2, 0, 2, 0, 2]$ を返した場合でも正解とみなされる。

入力例 2
1
100 7
3 21
16 9
44 36
44 78
45 78
67 59
90 22
84 59

この例では、関数 Bruno が $[3, 1, 1, 0, 0, 3, 2]$ などを返した場合、正解とみなされる。