



最後の戦い (The Last Battle)

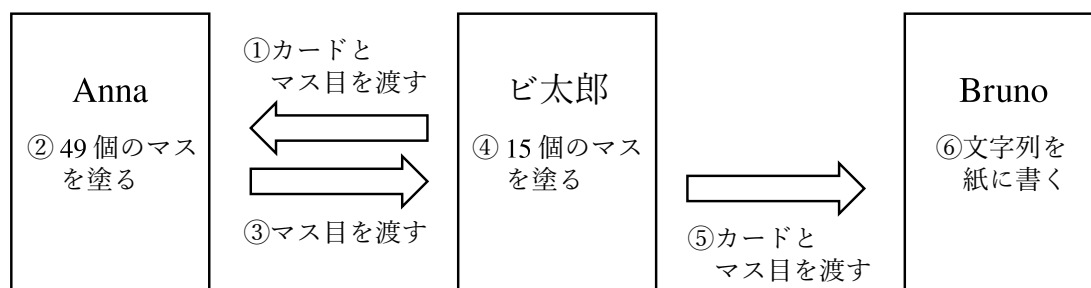
JOI 放送局が開催する人気バラエティ番組 JOICup は、いよいよ最終ラウンドを迎えることになった。最終ラウンドでは「メッセージゲーム」が行われ、予選を勝ち抜いた Anna と Bruno の 2 人からなるチームのみが出場する。

メッセージゲームは、8 行 8 列のマスを介して情報を伝えるゲームである。マス目の各行には 0 から 7 までの行番号が、各列には 0 から 7 までの列番号が付けられている。

メッセージゲームでは、Anna と Bruno が別の部屋に隔離された状態で Q 回のチャレンジが行われる。 i 回目 ($1 \leq i \leq Q$) のチャレンジは次のように進行する。

1. Anna は司会のビ太郎から、3 つの整数 X_i, Y_i, N_i ($0 \leq X_i \leq 7, 0 \leq Y_i \leq 7, 1 \leq N_i \leq 43$) および ‘A’ と ‘B’ からなる N_i 文字の文字列 S_i が書かれたカードと、すべてのマスが白で塗られた 8 行 8 列のマスを受け取る。
2. Anna は、行番号が X_i でなく列番号が Y_i でない 49 個のマスについて、各マスを青か赤で塗る。
3. Anna はマスを司会のビ太郎に渡す。
4. ビ太郎は、行番号が X_i または列番号が Y_i である 15 個のマスについて、各マスを青か赤で塗る。この操作は Anna や Bruno から見えない場所で行われる。
5. Bruno は司会のビ太郎から、整数 N_i のみが書かれたカードとマスを受け取る。
6. Bruno はある文字列を紙に書く。書いた文字列が S_i と一致していれば、Anna と Bruno の勝利となる。

チャレンジの進行については、以下の図も参照せよ。



Anna と Bruno が「メッセージゲーム」で勝利するための戦略を実装せよ。なお、この課題の採点方法については、採点基準の項を参照すること。



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 X, int Y, int N, std::string S)`

この関数は合計 Q 回呼び出される。 i 回目 ($1 \leq i \leq Q$) の呼び出しは、ゲームにおける i 回目のチャレンジの手順 1., 手順 2., 手順 3. に相当する。

- 引数 X は、 i 回目のチャレンジの手順 1. で `Anna` が受け取るカードに書かれた整数 X_i である。
- 引数 Y は、 i 回目のチャレンジの手順 1. で `Anna` が受け取るカードに書かれた整数 Y_i である。
- 引数 N は、 i 回目のチャレンジの手順 1. で `Anna` が受け取るカードに書かれた整数 N_i である。
- 引数 S は、 i 回目のチャレンジの手順 1. で `Anna` が受け取るカードに書かれた文字列 S_i である。

関数 `Anna` の1回の呼び出しについて、以下の関数を、行番号が X_i でなく列番号が Y_i でない49個のマスのそれぞれについて1回ずつ、合計49回呼び出さなければならない。

- ★ `void Paint(int a, int b, int c)`

- ◇ 引数 a, b は、行番号が a で列番号が b のマスに色を塗ることを表す。ここで、 $0 \leq a \leq 7, 0 \leq b \leq 7, a \neq X, b \neq Y$ を満たさなければならない。これが満たされない場合、不正解 [1] と判定される。
- ◇ 引数 c は、 $c = 0$ のとき青、 $c = 1$ のとき赤で塗ることを表す。ここで、 $0 \leq c \leq 1$ を満たさなければならない。これが満たされない場合、不正解 [2] と判定される。
- ◇ 関数 `Paint` を同じ (a, b) 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 `Paint` の呼び出し回数が49回でなかった場合、不正解 [4] と判定される。



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::string Bruno(int N, std::vector<std::vector<int>> T)`

この関数は Anna がマス目を塗り終わるたびに 1 回、合計で Q 回呼び出される。 i 回目 ($1 \leq i \leq Q$) の呼び出しは、ゲームにおける i 回目のチャレンジの手順 5., 手順 6. に相当する。

- 引数 N は、 i 回目のチャレンジの手順 5. で Bruno が受け取るカードに書かれた整数 N_i である。
- 引数 T は、 i 回目のチャレンジの手順 5. で Bruno が受け取るマス目の各マスの色を表す 8×8 の 2 次元配列である。行番号が a ($0 \leq a \leq 7$) で列番号が b ($0 \leq b \leq 7$) のマスの色は、 $T[a][b] = 0$ であれば青、 $T[a][b] = 1$ であれば赤である。
- 戻り値は、Bruno が紙に書く文字列である。
- 戻り値が 44 文字以上の文字列である場合、不正解 [5] と判定される。
- 戻り値に 'A' と 'B' 以外の文字が含まれている場合、不正解 [6] と判定される。

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。



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。なお、アーカイブの中に含まれている `compile.sh` というファイルを代わりに実行してもよい。

```
g++ -std=gnu++17 -O2 -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。特に、ビ太郎が各マスに塗る色が無作為に選ばれるとは限らない。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。

```
Q
X1 Y1 N1 S1
X2 Y2 N2 S2
⋮
XQ YQ NQ SQ
```

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、 L^* の値が “Accepted: 28” のように出力される。 L^* の値については採点基準の項を参照せよ。
- 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。



採点プログラムのサンプルにおいては、それぞれのチャレンジでビ太郎が各マスを塗る色が、実行ごとに結果が変わらない疑似乱数を用いて無作為に選ばれる。シードの値を変更したい場合は、

```
./grader 2023
```

のように 1 個目の引数に整数値を与えて実行せよ。

制約

- $1 \leq Q \leq 20\,000$.
- $0 \leq X_i \leq 7$ ($1 \leq i \leq Q$).
- $0 \leq Y_i \leq 7$ ($1 \leq i \leq Q$).
- $1 \leq N_i \leq 43$ ($1 \leq i \leq Q$).
- Q, X_i, Y_i, N_i は整数である。
- S_i ($1 \leq i \leq Q$) は 'A' と 'B' からなる N_i 文字の文字列である。

採点基準

この課題のテストケースの中で、1 つでも不正解 [1] ~ [6] (実装の詳細を参照) と判定されたものや、実行時エラー (実行時間制限超過, メモリ制限違反, 異常終了など) と判定されたものがあつた場合、他のテストケースでどのチャレンジに勝利したかにかかわらず無条件で 0 点となる。

そうでない場合、この課題のすべてのテストケースに対する以下の値の最小値を L^* とするとき、下表のように得点が与えられる。

- $N_i \leq L$ を満たすすべてのチャレンジについて勝利したような最大の整数 L 。ただし、テストケース内のすべてのチャレンジに勝利した場合は $L = 43$ とする。

L^*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
得点	0	5	8	10	11	13	14	16	18	19	21	22	24	26	27

L^*	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
得点	29	30	32	34	35	37	38	40	42	43	45	46	48	50	51

L^*	30	31	32	33	34	35	36	37	38	39	40	41	42	43
得点	53	54	56	57	59	60	62	65	68	71	74	77	84	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、関数 Bruno の呼び出しにおける引数 T は省略されている。

入力例 1	関数の呼び出しの例		
	Anna の呼び出し	Bruno の呼び出し	Bruno の戻り値
2 0 0 1 B 5 7 8 AAAABBBB	Anna(0, 0, 1, "B")		
	Paint(1, 1, 0)		
	Paint(1, 2, 1)		
	⋮		
	Paint(7, 7, 1)		
		Bruno(1, ...)	
			"B"
	Anna(5, 7, 8, "AAAABBBB")		
	Paint(0, 0, 1)		
	Paint(0, 1, 1)		
	⋮		
	Paint(7, 6, 0)		
		Bruno(8, ...)	
		"AAAABBBB"	

この入力例は $Q (= 2)$ 回のチャレンジからなる。

- 1 回目のチャレンジでは $X_1 = 0, Y_1 = 0, N_1 = 1, S_1 = \text{"B"}$ であるため、Anna は行番号が 0 でなく列番号が 0 でない 49 個のマスを色を塗らなければならない。
- 2 回目のチャレンジでは $X_2 = 5, Y_2 = 7, N_2 = 8, S_2 = \text{"AAAABBBB"}$ であるため、Anna は行番号が 5 でなく列番号が 7 でない 49 個のマスを色を塗らなければならない。

たとえば、1 回目のチャレンジで `Paint(0, 2, 1)` を呼び出した場合、行番号が 0 のマスを指定しているため不正解 [1] となることに注意せよ。



また、採点プログラムのサンプルが読み込む入力のもう 1 つの例を以下に示す。

入力例 2

```
30
3 1 1 A
1 4 1 A
6 6 2 AA
1 1 2 BB
3 1 3 BAB
7 4 3 AAB
6 4 4 BAAB
6 7 4 BABA
3 3 5 BABBA
1 5 5 ABBBA
4 3 6 ABBBBB
2 1 6 ABAAAA
6 0 7 AAABABA
6 6 7 BBABBAA
0 4 8 AABAABAB
2 1 8 ABBBBBBA
2 0 9 BABABAAA
1 5 9 BBAAABABB
6 7 10 BAAABAAABB
1 7 10 BBBBBBBABA
2 6 12 AABAABABABAB
3 4 15 BBAABAAAABABAAB
5 6 18 BAAAABBABABBABBAB
7 0 22 BABBAABAAABBABBBBBBABA
2 0 26 AAAABBABBAAAAABABABBAABAAA
0 7 30 AAABBBAAABAABBBBAABBAABBBABBB
2 7 34 BABAABBAABABBABAABBABBABAABBBBABB
2 5 38 BBBBAABAABAABABABBBBBBAAAABBABAAAABAAABB
5 2 41 AABABBAABBBAAAAABBABBBAAAAABBABBABBABA
1 0 43 AABBBABBBABABBABBBAABAAAAABABAABBBBAABBAAB
```