



## 三色灯 (Tricolor Lights)

賭け事の達人である Anna と Bruno の 2 人組は、ディーラーのディ太郎を相手にゲームで勝負することとなった。

ゲーム中、Anna と Bruno は別々の部屋に隔離され、ディ太郎を介してのみやりとりができる。

このゲームでは横一列に並んだ  $N$  個のライトを用いる。これらのライトには左から順に 1 から  $N$  までの番号が付けられている。これらのライトは赤、緑、青のうち、いずれかの色に点灯させることができる。

Anna は、ゲーム開始時にこれらのライトそれぞれについて赤、緑、青のいずれかの色に点灯させる。また、ディ太郎により、それぞれのライトについて 1 つの禁止色が定められており、これは長さ  $N$  の文字列  $S$  によって表される。 $S$  の  $i$  文字目 ( $1 \leq i \leq N$ ) を  $S_i$  とすると、ライト  $i$  の禁止色は、 $S_i$  が 'R' のとき赤、'G' のとき緑、'B' のとき青である。Anna はライト  $i$  をそのライトの禁止色に点灯させることができない。例えば、 $S_1$  が 'R' であるならば、Anna はライト 1 について、緑か青のいずれかの色に点灯させなければならない。ディ太郎がそれぞれのライトについて定めた禁止色の情報は、Anna には伝えられるが、Bruno には伝えられない。

Anna はそれぞれのライトを点灯させたのち、 $1 \leq l \leq \min(N, 130)$  を満たす整数  $l$  を選び、ディ太郎に伝える。ディ太郎は、Bruno にライトの個数  $N$  と、Anna が選んだ整数  $l$  を伝える。その後、 $Q$  回のラウンドを行う。 $j$  回目 ( $1 \leq j \leq Q$ ) のラウンドは、以下のように進行する。

- ディ太郎は 1 以上  $N - l + 1$  以下の整数  $a_j$  を選び、ライト  $a_j, a_j + 1, \dots, a_j + l - 1$  について点灯している色の並びを Bruno に見せる。
- Bruno は、見せられた色の並びをもとに、1 つの整数をディ太郎に伝える。この整数が  $a_j$  と一致すれば、このラウンドで Anna と Bruno は勝利する。

ただし、ディ太郎は、Anna が点灯させたライトの色の並びや整数  $l$  に応じて、 $a_1, a_2, \dots, a_Q$  の選び方を変える可能性がある。

Anna と Bruno が  $Q$  回すべてのラウンドで勝利するため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。

- `std::pair<std::string, int> anna(int N, std::string S)`

この関数は、最初に1回だけ呼び出される。

- 引数  $N$  は、ライトの個数を表す整数  $N$  である。
- 引数  $S$  は、ディ太郎が定めた禁止色を表す長さ  $N$  の文字列  $S$  である。
- 戻り値は、`Anna` がそれぞれのライトに点灯させる色を表す長さ  $N$  の文字列  $t$ 、整数  $l$  のペアである。 $t$  の  $i$  文字目 ( $1 \leq i \leq N$ ) を  $t_i$  とすると、`Anna` が  $i$  番目のランプを色  $t_i$  に点灯させることを意味する。 $t_i$  は、'R' のとき赤、'G' のとき緑、'B' のとき青を意味する。
- 文字列  $t$  の長さは  $N$  でなければならない。これが満たされない場合、不正解 [1] と判定される。
- 文字列  $t$  の各文字は 'R' または 'G' または 'B' でなければならない。これが満たされない場合、不正解 [2] と判定される。
- 文字列  $t$  の  $i$  文字目 ( $1 \leq i \leq N$ ) は  $S$  の  $i$  文字目と異ならなければならない。これが満たされない場合、不正解 [3] と判定される。
- $l$  は1以上  $\min(N, 130)$  以下でなければならない。これが満たされない場合、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。

- `void init(int N, int l)`

この関数は、最初に1回だけ呼び出される。

- 引数  $N$  は、ライトの個数を表す整数  $N$  である。
- 引数  $l$  は、`Anna` が指定した整数  $l$  である。

- `int bruno(std::string u)`

この関数は、関数 `init` が呼び出された後に、合計で  $Q$  回呼び出される。 $j$  回目 ( $1 \leq j \leq Q$ ) の呼び出しは、ゲームにおける  $j$  回目のラウンドの手順 1. および手順 2. に相当する。

- 引数  $u$  は 'R', 'G', 'B' からなる長さ  $l$  の文字列で、ライト  $a_j, a_j + 1, \dots, a_j + l - 1$  に点灯している色の並びを表す。 $u$  の  $k$  文字目 ( $1 \leq k \leq l$ ) を  $u_k$  とすると、`Anna` が  $a_j + k - 1$  番目のランプに点



灯させた色は、 $u_k$  が 'R' のとき赤、'G' のとき緑、'B' のとき青である。

- 戻り値は、Bruno が答える整数である。
- 戻り値は、 $a_j$  と一致していなければならない。これが満たされない場合、不正解 [5] 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つ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++20 -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。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。

```
N
S
Q
a1 a2 ⋯ aQ
```

採点プログラムのサンプルは、実際の採点プログラムと異なり、初めから固定された答えを持つ必要がある。  $a_j$  ( $1 \leq j \leq Q$ ) は、ディ太郎が  $j$  回目のラウンドで選ぶ整数  $a_j$  を表す。あなたが書いたプログラムにおいて Anna が選んだ整数  $l$  に対して、 $1 \leq a_j \leq N - l + 1$  を満たさなければならない。

## 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、Anna が選んだ整数  $l$  が “Accepted: 130” 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。

## 採点に関する注意

実際の採点プログラムは、Anna が点灯させたライトの色の並びや選んだ整数  $l$  に応じて  $a_1, a_2, \dots, a_Q$  を定める。特に、Bruno が答えた整数によって  $a_1, a_2, \dots, a_Q$  の値を変えることはない。

## 制約

- $1 \leq N \leq 500\,000$ .
- $1 \leq Q \leq 10\,000$ .
- $S$  は ‘R’, ‘G’, ‘B’ からなる長さ  $N$  の文字列である。
- $N, Q$  は整数である。



## 小課題

1. (5 点)  $N \leq 131$ .
2. (5 点)  $N \leq 250$ .
3. (5 点)  $N \leq 380$ .
4. (15 点)  $N \leq 7000$ .
5. (70 点) 追加の制約はない。この小課題では、以下に従い得点が決定される。
  - この小課題のすべてのテストケースにおける、Anna が選んだ整数  $l$  の最大値を  $l^*$  とする。
  - この小課題のテストケースの中で、1 つでも不正解 [1] ~ [5] (実装の詳細を参照) と判定されたものや、実行時間制限超過、メモリ制限超過、実行時エラーと判定されたものがあつた場合、小課題の得点は 0 点となる。
  - この小課題におけるすべてのテストケースに正解した場合、小課題の得点は以下のように決定される。

$l^*$ の値	得点
$61 < l^* \leq 130$	10 点
$41 < l^* \leq 61$	20 点
$34 < l^* \leq 41$	$25 + 3 \times (41 - l^*)$ 点
$28 < l^* \leq 34$	$46 + 4 \times (34 - l^*)$ 点
$l^* \leq 28$	70 点



## 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。

入力例 1	関数の呼び出しの例	
	呼び出し	戻り値
8	<code>anna(8, "RGGBRBBG")</code>	<code>("BBRGBGRR", 5)</code>
RGGBRBBG	<code>init(8, 5)</code>	
2	<code>bruno("RGBGR")</code>	3
3 1	<code>bruno("BBRGB")</code>	1

このやりとりの例において、Anna はライトの個数  $N = 8$ 、禁止色を表す文字列  $S = \text{"RGGBRBBG"}$  を受け取り、それぞれのライトに点灯させる色を表す文字列  $t$  として `"BBRGBGRR"`、整数  $l$  として 5 を指定し、ディ太郎に伝える。その後、ディ太郎は Bruno に整数  $N = 8$ 、 $l = 5$  を伝える。

1 回目のラウンドでは、ディ太郎は  $a_1 = 3$  としている。Bruno はライト 3, 4, 5, 6, 7 に点灯している色の並びを表す文字列  $u = \text{"RGBGR"}$  を受け取り、 $a_1$  である 3 を答える。

入力例 1 はすべての小課題の制約を満たす。コンテストサイトからダウンロードできるファイルのうち、`sample-01-in.txt` は入力例 1 に対応する。