



壊れた機器 2 (Broken Device 2)

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、ある壊れた機器を介してのみ意思疎通ができる。この機器を介して、Anna がディ太郎から伝えられた整数を Bruno に伝えることが、このゲームにおける 2 人の目標である。

ゲームが始まったら、はじめに Anna は 1 以上 2000 以下の整数 m を宣言する。その後、 Q 回のラウンドを行う。 i 回目 ($1 \leq i \leq Q$) のラウンドは、次の通りに進行する。

1. ディ太郎は、整数 A_i を Anna に伝える。
2. Anna は、すべての要素が 0 または 1 である数列 s_i, t_i を機器に入力する。数列 s_i と t_i の長さは等しく、かつその長さは 1 以上 m 以下でなければならない。
3. 機器は、数列 s_i と t_i のリフルシャッフル（後述）で得られる数列のうちの一つである u_i を Bruno に伝える。
4. Bruno は、1 つの整数をディ太郎に伝える。この整数が、ラウンド冒頭でディ太郎が Anna に伝えた整数 A_i と一致すれば、このラウンドで Anna と Bruno は勝利する。

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。

リフルシャッフルについて

数列 Z が数列 X と Y のリフルシャッフルで得られるとは、 Z の各要素を 2 つのグループに上手く分けることで、以下の条件をすべて満たすことが可能であることを言う。

- 1 つ目のグループの要素をそのままの順番で抜き出すと、数列 X と一致する。
- 2 つ目のグループの要素をそのままの順番で抜き出すと、数列 Y と一致する。

たとえば、 $X = [1, 1, 0]$, $Y = [1, 0, 0]$, $Z = [1, 1, 1, 0, 0, 0]$ のとき、 Z は X と Y のリフルシャッフルで得られる。なぜなら、 Z の 1, 2, 4 番目の要素をそのままの順番で抜き出すと $[1, 1, 0]$ になり、 Z の 3, 5, 6 番目の要素をそのままの順番で抜き出すと $[1, 0, 0]$ になるためである。

一方、 $X = [1, 1, 0]$, $Y = [1, 0, 0]$, $Z = [0, 0, 0, 1, 1, 1]$ のとき、 Z は X と Y のリフルシャッフルで得られない。



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。

- `int Declare()`

この関数は、最初に1回だけ呼び出される。

- 戻り値は、`Anna` が宣言する整数 m である。
- m は1以上2000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 [1] と判定される。

- `std::pair<std::vector<int>, std::vector<int>> Anna(long long A)`

この関数は、関数 `Declare` が呼び出された後に、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 回目のラウンドの手順 1. および手順 2. に相当する。

- 引数 A は、 i 回目のラウンド冒頭でディ太郎が `Anna` に伝える整数 A_i である。
- 戻り値は、`Anna` が機器に入力する2つの数列 s_i, t_i を表す配列のペアである。
- 数列 s_i と t_i の各要素は0または1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。
- 数列 s_i と t_i の長さはともに1以上 m 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 [3] と判定される。
- 数列 s_i と t_i の長さは等しくなければならない。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。

- `long long Bruno(std::vector<int> u)`

この関数は、`Anna` が機器への入力を完了するたびに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 回目のラウンドの手順 3. および手順 4. に相当する。

- 引数 u は、 i 回目のラウンドで機器が `Bruno` に伝える数列 u_i を表す配列である。
- 戻り値は、`Bruno` がディ太郎に伝える整数である。
- 戻り値は、ディ太郎が `Anna` に伝えた整数 A_i 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。

```
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。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
 A_1
 A_2
 \vdots
 A_Q

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 `Declare` の戻り値 m が “Accepted: 2000” 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。

採点プログラムのサンプルにおいては、各ラウンドにおける機器のリフレッシュの結果が、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 2022
```

のように 1 個目の引数に整数値を与えて実行せよ。



制約

- $1 \leq Q \leq 1\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000\,000\,000\,000\,000\,000 (= 10^{18})$ ($1 \leq i \leq Q$).

小課題

1. (5 点) $A_i \leq 2\,000$ ($1 \leq i \leq Q$).
2. (5 点) $A_i \leq 4\,000\,000$ ($1 \leq i \leq Q$).
3. (3 点) $A_i \leq 10\,000\,000 (= 10^7)$ ($1 \leq i \leq Q$).
4. (12 点) $A_i \leq 100\,000\,000 (= 10^8)$ ($1 \leq i \leq Q$).
5. (15 点) $A_i \leq 100\,000\,000\,000 (= 10^{11})$ ($1 \leq i \leq Q$).
6. (60 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まる。
 - この小課題のすべてのテストケースにおける、Anna が宣言した整数 m の最大値を m^* とする。
 - このとき、小課題の得点は以下のように決定される。

m^* の値	得点
$201 \leq m^* \leq 2\,000$	$40 - 25 \times \log_{10} \left(\frac{m^*}{200} \right)$ 点を整数に切り捨てた得点
$161 \leq m^* \leq 200$	40 点
$156 \leq m^* \leq 160$	44 点
$151 \leq m^* \leq 155$	48 点
$146 \leq m^* \leq 150$	52 点
$141 \leq m^* \leq 145$	56 点
$m^* \leq 140$	60 点



やり取りの例

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
2	Declare()	4
42	Anna(42)	([0, 0, 1, 0], [1, 1, 0, 1])
2000	Bruno([1, 0, 0, 1, 0, 1, 0, 1])	42
	Anna(2000)	([0, 1], [0, 0])
	Bruno([0, 0, 1, 0])	2000

この入力例は $Q (= 2)$ 回のラウンドからなる。1 回目のラウンドでは整数 $A_1 (= 42)$ がディ太郎から Anna に伝えられ、2 回目のラウンドでは整数 $A_2 (= 2000)$ がディ太郎から Anna に伝えられる。

この入力例はすべての小課題の制約を満たす。