モンスターゲーム (Monster Game)

新しいゲームが発売された．このゲームの世界には，N 匹のモンスターがおり，0 から $N-1$ までの番号が付けられている．また，それぞれのモンスターには**強さ**という整数の値が定まっている．モンスター i $(0 \leqq i \leqq N-1)$ の強さは S_i である．強さについて，以下のことが分かっている．

- それぞれのモンスターの強さは，0 以上 $N-1$ 以下の整数である．
- どの異なる 2 匹のモンスターの強さも同じであることはない．

あなたは異なる 2 匹のモンスターを選んで対戦をさせることができる．モンスター a とモンスター b $(0 \leqq a \leqq N-1,\ 0 \leqq b \leqq N-1,\ a \neq b)$ に対戦をさせると，次の規則に従って勝敗が決まる．

- $|S_a - S_b| = 1$ のとき，強さが小さい方のモンスターが勝利する．
- $|S_a - S_b| > 1$ のとき，強さが大きい方のモンスターが勝利する．

勝敗にかかわらず，あなたは同じモンスターに何度でも対戦をさせることができる．

あなたは最初，それぞれのモンスターの強さを知らない．そこで，対戦をさせて勝敗を知ることを 25 000 回以下繰り返すことによって，それぞれのモンスターの強さを求めたい．さらに，対戦の回数をなるべく少なくしたい．

モンスターの数が与えられたとき，モンスターに何度か対戦をさせることで，モンスターの強さを求めるプログラムを作成せよ．

実装の詳細

あなたは 1 つのファイルを提出しなければならない．

あなたの提出するファイルは `monster.cpp` という名前である．このファイルは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 `monster.h` を読み込むこと．

- `std::vector<int> Solve(int N)`
  この関数は各テストケースにおいて 1 回だけ呼び出される．
  - 引数 `N` はモンスターの数 N を表す．
  - 戻り値は配列であり，あなたのプログラムが求めたモンスターの強さを表す．以下，戻り値を `T` として説明する．
  - `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．
  - `T` の各要素は 0 以上 $N-1$ 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 [2]** と判定される．

- すべての i $(0 \leqq i \leqq N-1)$ に対し，T[i] $= S_i$ 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．

あなたのプログラムは以下の関数を呼び出すことができる．

★ `bool Query(int a, int b)`

この関数を用いて，モンスターに対戦をさせる．

- 引数 a と b は，対戦をさせるモンスターの番号 a, b である．
- 戻り値は対戦の結果を表し，モンスター a が勝利した場合は true，モンスター b が勝利した場合は false を返す．
- $0 \leqq a \leqq N-1$ かつ $0 \leqq b \leqq N-1$ 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．
- $a \neq b$ でなければならない．これが満たされない場合，不正解 **[5]** と判定される．
- 関数 Query を 25 000 回を超えて呼び出してはならない．25 000 回を超えて呼び出した場合，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．
- 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，これらのファイル grader.cpp, monster.cpp, monster.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 monster.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
$S_0 \cdots S_{N-1}$

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 `Query` の呼び出し回数が “`Accepted: 100`” 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

いくつかのテストケースについて，実際の採点プログラムは適応的 (adaptive) である．これは，採点プログラムは初めから固定された答えを持たず，それ以前の `Query` 関数の呼び出しに応じて採点プログラムが応答するということである．ただし，すべての応答に矛盾しないような答えが少なくとも 1 つ存在するということが保証される．

制約

- $4 \leqq N \leqq 1\,000$.
- $0 \leqq S_i \leqq N-1$ $(0 \leqq i \leqq N-1)$.
- $S_i \neq S_j$ $(0 \leqq i < j \leqq N-1)$.

小課題

1. (10 点) $N \leqq 200$.
2. (15 点) テストケースは適応的ではない．

3. (75 点) 追加の制約はない．すべてのテストケースに正解した場合，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，関数 `Query` が呼び出された回数の最大値を X 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．
     - $10\,000 < X \leqq 25\,000$ のとき，$\left\lfloor 75 \times \dfrac{25\,000 - X}{15\,000} \right\rfloor$ 点．
     - $X \leqq 10\,000$ のとき，75 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	呼び出し	戻り値
5 3 1 4 2 0	`Solve(5)`			
			`Query(1, 0)`	`false`
			`Query(4, 0)`	`false`
			`Query(1, 3)`	`true`
		`[3, 1, 4, 2, 0]`		