



## ラーメンの食べ比べ (Ramen)

JOI 君と IOI ちゃんはラーメンが好きである。JOI 君はあっさりしたラーメンが好きなのに対して、IOI ちゃんはこってりしたラーメンが好きである。JOI 君と IOI ちゃんが住んでいる町には、ラーメン屋 0 からラーメン屋  $N-1$  までの  $N$  軒のラーメン屋がある。

どのラーメン屋がこってりしたラーメンを提供しているか、またどのラーメン屋があっさりしたラーメンを提供しているかはわからない。そこで、JOI 君と IOI ちゃんは、近所のラーメン屋を巡って、最もあっさりしたラーメンを提供しているラーメン屋と最もこってりしたラーメンを提供しているラーメン屋がどこであるかを決めることにした。

JOI 君と IOI ちゃんが住んでいる町にあるラーメン屋には、そのラーメン屋で食べることができるラーメンのこってり度がそれぞれ定まっている。こってり度は 0 以上  $N-1$  以下の整数であり、それぞれのラーメン屋のこってり度は互いに異なる。JOI 君と IOI ちゃんは 1 日に 2 軒のラーメン屋を巡り、味を比べることで、2 軒のラーメン屋のうちこってり度が高いのがどちらのラーメン屋であるかを判定することができる。

健康のため、JOI 君と IOI ちゃんはラーメンの食べ比べを行う日数を 600 日までに抑えたい。

### 課題

町にあるラーメン屋の軒数  $N$  が与えられたとき、高々 600 日の食べ比べによって、 $N$  軒のラーメン屋の中でこってり度が最も低いラーメン屋と、こってり度が最も高いラーメン屋を決定するプログラムを作成せよ。



## 実装の詳細

あなたは、こってり度が最も低いラーメン屋と、こってり度が最も高いラーメン屋を、高々 600 日の食べ比べによって決定する方法を実装した 1 個のプログラムを書かねばならない。

プログラムは、以下のルーチンを実装しなければならない。

- **void Ramen(int N)**

このルーチンは、各テストケースに対し 1 回だけ呼び出される。引数 **N** は町にあるラーメン屋の軒数である。

このルーチンは、**Compare** を呼び出すことによって、こってり度が最も低いラーメン屋とこってり度が最も高いラーメン屋を決定し、**Answer** を呼び出すことで終了しなければならない。

プログラム中では以下の関数を呼び出すことができる。

- **int Compare(int X, int Y)**

この関数は、ラーメンの食べ比べを行う際に呼び出す。引数 **X, Y** はそれぞれ食べ比べを行うラーメン屋の番号 **X, Y** である。**X** と **Y** は 0 以上  $N-1$  以下の互いに異なる整数でなくてはならない。これを満たさない引数とともに **Compare** を呼び出した場合は **不正解 [1]** となり、プログラムは終了する。

この関数は、ラーメン屋 **X** のこってり度がラーメン屋 **Y** のこってり度より大きいときは 1 を返し、ラーメン屋 **X** のこってり度がラーメン屋 **Y** のこってり度より小さいときは -1 を返す。

600 回より多く **Compare** を呼び出した場合は **不正解 [2]** となり、プログラムは終了する。

**Ramen** は、次のルーチン呼び出しによって終了しなければならない。**Ramen** が **Answer** を呼び出さなかった場合は **不正解 [3]** となり、プログラムは終了する。

- **void Answer(int X, int Y)**

このルーチンは、ラーメンの食べ比べによって、こってり度が最も低いラーメン屋とこってり度が最も高いラーメン屋が定まっている状態で呼び出す。引数 **X, Y** は、それぞれ、こってり度が最も低いラーメン屋の番号 **X** と、こってり度が最も高いラーメン屋の番号 **Y** である。**X, Y** はそれぞれ 0 以上  $N-1$  以下の整数でなくてはならない。これを満たさない引数とともに **Answer** を呼び出した場合は **不正解 [4]** となる。

**Compare** の呼び出しの結果と矛盾しない答えが唯一であり、その答えを引数にして **Answer** を呼び出した場合は **正解** となる。そうでない場合は **不正解 [5]** となる。

このルーチンが呼ばれると、プログラムは終了する。

## 採点上の注意

採点の際には、**Compare** の呼び出しの結果に矛盾しない答えが一意に定まっていない場合は、**Answer** の引数によらず、**不正解 [5]** と判定される。

採点用のデータの中には、**Compare** の返り値が以前の **Compare** の呼び出しの内容によって変化するものがある。この場合も **Compare** の返り値は以前の **Compare** の呼び出しの結果と矛盾することはない。



## 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.c` または `grader.cpp` である。例えば、作成したプログラムを `Ramen.c` または `Ramen.cpp` とするとき、作成したプログラムをテストするには、次のようにコマンドを実行する。

- C の場合

```
gcc -O2 -lm grader.c Ramen.c -o grader
```

- C++ の場合

```
g++ -O2 grader.cpp Ramen.cpp -o grader
```

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入力を読み込む。

- 1行目には整数  $N, T$  が空白を区切りとして書かれている。 $N$  はラーメン屋の軒数である。採点プログラムのサンプルは、 $T = 1$  であるデータを扱う。
- 続く  $N$  行のうちの  $i + 1$  行目 ( $0 \leq i \leq N - 1$ ) には整数  $A_i$  ( $0 \leq A_i \leq N - 1$ ) が書かれている。これは、ラーメン屋  $i$  のこってり度を表す。

## 採点プログラムのサンプルの出力

プログラムの実行が正常に終了した場合、採点プログラムのサンプルは標準出力へ以下の情報を1行で出力する (引用符は実際には出力されない)。

- 正解の場合、“Accepted” と出力される。
- 不正解の場合、不正解の種類が “Wrong Answer [2]” のように出力される。

なお、採点プログラムのサンプルは、 $A_X = 0, A_Y = N - 1$  なる  $X, Y$  について `Answer(X, Y)` を呼び出した場合は、不正解 [5] に該当する場合でも正解と判定する。これは実際の採点プログラムの挙動とは異なるので注意せよ。



## 制限

すべての入力データは以下の条件を満たす.

- $1 \leq N \leq 400$ .

## 小課題

### 小課題 1 [20 点]

- $N \leq 30$  を満たす.

### 小課題 2 [30 点]

- $N \leq 300$  を満たす.

### 小課題 3 [50 点]

追加の制限はない.

## 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とルーチンの呼び出しの例を以下に示す.

入力例	関数とルーチンの呼び出しの例	
3 1 1 2 0	呼び出し	戻り値
	<code>Compare(0, 1)</code>	-1
	<code>Compare(0, 2)</code>	1
	<code>Answer(2, 1)</code>	プログラムは終了する