



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。

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 `ramen.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。そのプログラムは `#include` プリプロセッサ指令によって `ramen.h` を読み込むこと。

- `void Ramen(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 は町にあるラーメン屋の軒数 N である。

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。

★ `int Compare(int X, int Y)` この関数を用いて、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。

- 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 を表す。
- 戻り値は、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であり、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である。
- 引数 X, Y は0以上 $N-1$ 以下の互いに異なる整数でなければならない。これが満たされていない場合、不正解 [1] と判定される。
- 関数 `Compare` を600回超えて呼び出してはならない。600回を超えて呼び出した場合、不正解 [2] と判定される。

★ `void Answer(int X, int Y)` この関数を用いて、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回答する。

- 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 を表す。
- 引数 X, Y はそれぞれ0以上 $N-1$ 以下の整数でなければならない。これが満たされていない場合、不正解 [3] と判定される。
- ラーメン屋 X がこってり度が最も低いラーメン屋でない場合、あるいはラーメン屋 Y がこってり度が最も高いラーメン屋でない場合、不正解 [4] と判定される。
- 関数 `Ramen` の実行の終了時にこの関数が1回も呼び出されていない場合、不正解 [5] と判定される。
- 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 [6] と判定される。
- この関数が呼び出されると、プログラムは終了する。



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`, `ramen.cpp`, `ramen.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 ramen.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入力を読み込む。

```
N  
A0  
A1  
⋮  
AN-1
```

A_i ($0 \leq A_i \leq N-1$) は、ラーメン屋 i のこってり度を表す整数である。



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.

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.

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 [6] 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.

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 `Compare` 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

すべての入力データは以下の条件を満たす. A の意味については「採点プログラムのサンプルの入力」の項目を参照せよ.

- $1 \leq N \leq 400$.
- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq N - 1$).
- $A_i \neq A_j$ ($0 \leq i < j \leq N - 1$).

小課題

1. (20 点) $N \leq 30$.
2. (30 点) $N \leq 300$.
3. (50 点) 追加の制約はない.



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の呼び出しの例を以下に示す。

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