



買い物 (Shopping)

JOI 商店には N 個の商品があり、各商品には 0 から $N-1$ までの番号が付いている。商品 i ($0 \leq i \leq N-1$) の値段は P_i である。どの 2 つの商品の値段も相異なる。

Anna は JOI 商店に買い物に来ている。Anna は番号が L 以上 R 以下の商品のうち、最も安い商品を購入することに決めた。しかし、Anna は各商品の値段が分からなかったため、店員である Bruno と通信をすることで、購入する商品の番号を求めることにした。Bruno は各商品の値段を知っているが、 L と R の値は知らない。

Anna と Bruno は、 0 または 1 の文字を送受信する機械を用いて通信を行う。Anna から Bruno へは合計で 18 文字まで、Bruno から Anna へは合計で 10000 文字まで送信することができる。ただし、Bruno は送信する文字数をできるだけ少なくしたい。

Anna 側に N, L, R の値が、Bruno 側に N の値と各商品の値段が与えられたとき、通信を行うことで、Anna が購入する商品の番号を Anna 側で求める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 の行動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。

- `void InitA(int N, int L, int R)`

この関数は、最初に 1 回だけ実行される。

- 引数 N は、商品の個数 N である。
- 引数 L と R は、Anna が番号が L 以上 R 以下の商品のうち、最も安い商品を購入することを表す。

- `void ReceiveA(bool x)`

この関数は、Bruno から文字が送られる度に実行される。

- 引数 x は、Bruno から送信された文字を表す。true は文字 1 を表し、false は文字 0 を表す。

- `int Answer()`

この関数は、文字の送受信がすべて完了したときに 1 回だけ実行される。Anna が購入する商品の番号を戻り値として返さなければならない。

- 戻り値は、 L 以上 R 以下の整数でなければならない。この条件を満たさない場合、不正解 [1] と判定される。また、戻り値が Anna の購入する商品の番号と異なる場合、不正解 [2] と判定される。



このファイルでは以下の関数を呼び出すことができる。

★ void SendA(bool y)

この関数を用いて、Bruno に文字を送信する。

- 引数 y は、Bruno に送信する文字を表す。true は文字 1 を表し、false は文字 0 を表す。

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 の行動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 InitB(int N, std::vector<int> P)

この関数は、最初に 1 回だけ実行される。

- 引数 N は、商品の個数 N である。
- 引数 P は、長さ N の配列であり、 $P[i]$ は商品 i の値段 P_i を表す ($0 \leq i \leq N - 1$)。

● void ReceiveB(bool y)

この関数は、Anna から文字が送られる度に実行される。

- 引数 y は、Anna から送信された文字を表す。true は文字 1 を表し、false は文字 0 を表す。

このファイルでは以下の関数を呼び出すことができる。

★ void SendB(bool x)

この関数を用いて、Anna に文字を送信する。

- 引数 x は、Anna に送信する文字を表す。true は文字 1 を表し、false は文字 0 を表す。

プログラムは次のように実行されると考えてよい。各テストケースにおいて、Anna が送信した文字を保管するキュー Q_Y と Bruno が送信した文字を保管するキュー Q_X が用意される。最初に InitA および InitB が呼び出され、そこで送信された文字がそれぞれのキューに追加される。次に、以下の処理を繰り返す。

- Q_X と Q_Y のいずれかが空でない場合、そこから文字を 1 つ取り出し、対応する ReceiveA または ReceiveB が実行される。ただし、 Q_X と Q_Y のどちらも空でない場合、ReceiveA と ReceiveB のどちらが実行されるかは分からない。
- ReceiveA の実行中に SendA が実行された場合、送信された文字が Q_Y に追加される。
- ReceiveB の実行中に SendB が実行された場合、送信された文字が Q_X に追加される。
- どちらのキューも空である場合は、Answer が実行された後、プログラムが終了する。

Anna から Bruno へ送信する文字数の合計は 18 を超えてはならない。もし超えた場合、不正解 [3] と判定される。また、Bruno から Anna へ送信する文字数の合計は 10000 を超えてはならない。もし超えた場



合，不正解 [4] 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。

```
g++ -std=gnu++17 -O2 -fsigned-char -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，標準出力と標準エラー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 LR  
P0 P1 ⋯ PN-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入力する (引用符は実際には出力されない).

- 正解の場合, Anna から Bruno へ送信された文字数の合計 Y と, Bruno から Anna へ送信された文字数の合計 X が “Accepted: $Y X$ ” 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.

制約

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq L \leq R \leq N - 1$.
- $1 \leq P_i \leq N$ ($0 \leq i \leq N - 1$).
- $P_i \neq P_j$ ($0 \leq i < j \leq N - 1$).

小課題

1. (1 点) $N \leq 1\,000$.
 2. (9 点) $N \leq 10\,000$.
 3. (90 点) 追加の制約はない.
 - この小課題のすべてのテストケースにおける, Bruno から Anna へ送信された文字数の合計の最大値を T とする.
 - このとき, この小課題の得点は以下のように決定される.
 - $5\,000 < T \leq 10\,000$ のとき, $\left\lfloor 25 \times \frac{10\,000 - T}{5\,000} \right\rfloor$ 点.
 - $1\,000 < T \leq 5\,000$ のとき, $25 + \left\lfloor 40 \times \frac{5\,000 - T}{4\,000} \right\rfloor$ 点.
 - $300 < T \leq 1\,000$ のとき, $65 + \left\lfloor 25 \times \frac{1\,000 - T}{700} \right\rfloor$ 点.
 - $T \leq 300$ のとき, 90 点.
- ここで, $\lfloor x \rfloor$ は x を超えない最大の整数を表す.



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	呼び出し	戻り値
4 0 2 3 1 4 2	InitA(4, 0, 2)		
		SendA(true)	
		SendA(false)	
	InitB(4, {3, 1, 4, 2})		
	ReceiveB(true)		
		SendB(true)	
	ReceiveA(true)		
	ReceiveB(false)		
	Answer()		1