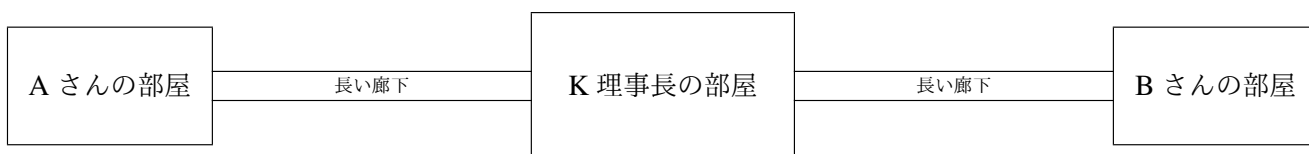




## 一緒に足し算 (Add together)

国際情報オリンピックの日本代表に選ばれた A さんと B さんは、情報処理技術を高めるため、情報オリンピック日本委員会の K 理事長に以下のような課題を提示された。

A さんと B さんは、図のように部屋に隔離されている。A さん・B さん・K 理事長の各部屋には内線電話が設置されているが、電話を発信することが可能なのは K 理事長のみである。その他の連絡手段は断たれており、A さんと B さんは指示がない限り部屋から出ることはできない。



課題の目的は、K 理事長が A さんと B さんにいくつかの整数を伝えていったとき、A さんと B さんが直接の連絡をとらずに、「A さんに伝えられた偶数すべての和」と「B さんに伝えられた奇数すべての和」の合計を B さんが正しく求めることである。

K 理事長の部屋には赤と青の 2 枚の電光掲示板がある。これらは 0 以上 999 999 999 以下の整数を 1 個表示できる。課題の開始時、両方の電光掲示板は 0 を表示している。

これから、K 理事長が A さんまたは B さんを電話で部屋に呼ぶ、ということが行われていく。A さんまたは B さんは、K 理事長の部屋に来るたびに、整数を 1 個伝えられる。その後、A さんならば赤の電光掲示板、B さんならば青の電光掲示板に表示する整数を 1 個指定する。K 理事長は A さんまたは B さんの指定通りに電光掲示板の表示する整数を変更する。

最後に、K 理事長は B さんを電話で部屋に呼ぶ。このとき B さんは、「A さんに伝えられた偶数すべての和」と「B さんに伝えられた奇数すべての和」の合計を答える。正しく答えられれば正解であり、そうでなければ不正解である。

K 理事長が A さんまたは B さんを部屋に呼ぶタイミングは不規則であり、A さんと B さんは K 理事長がどういう順序で 2 人を部屋に呼んでいるかはわからない。

A さんと B さんの戦略を実装し、上で説明された課題で正解でき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 `playerA.cpp` という名前である。このファイルは A さん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。

- `void InitA(int T)` この関数は最初に 1 回だけ呼び出される。引数 `T` は、小課題の番号 `T` である。



- `int GameA(int Q, int X)` この関数は、AさんがK理事長の部屋に来ることに対応して呼び出される。引数  $Q$  は、AさんがK理事長の部屋に来たときに青の電光掲示板が表示している整数  $Q$  である。  $0 \leq Q \leq 999\,999\,999$  を満たす。引数  $X$  は、AさんがK理事長に伝えられる整数  $X$  である。  $1 \leq X \leq 100$  を満たす。この関数は  $0 \leq P \leq 999\,999\,999$  を満たす整数  $P$  を返さなければならない。これは、Aさんが赤い電光掲示板に表示する整数として  $P$  を指定することを表す。範囲外の値を返した場合は、不正解 [1] と判定され、プログラムの実行は終了される。

2つ目のファイルは `playerB.cpp` という名前である。このファイルはBさん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。

- `void InitB(int T)` この関数は最初に1回だけ呼び出される。引数  $T$  は小課題の番号  $T$  である。
- `int GameB(int P, int X)` この関数は、BさんがK理事長の部屋に来ることに対応して呼び出される。引数  $P$  は、BさんがK理事長の部屋に来たときに赤の電光掲示板が表示している整数  $P$  である。  $0 \leq P \leq 999\,999\,999$  を満たす。引数  $X$  は、BさんがK理事長に伝えられる整数  $X$  である。  $1 \leq X \leq 100$  を満たす。この関数は  $0 \leq Q \leq 999\,999\,999$  を満たす整数  $Q$  を返さなければならない。これは、Bさんが青い電光掲示板に表示する整数として  $Q$  を指定することを表す。範囲外の値を返した場合は、不正解 [2] と判定され、プログラムの実行は終了される。
- `int AnswerB(int P)` この関数は、課題の最後にBさんがK理事長の部屋に来ることに対応して呼び出される。引数  $P$  は、BさんがK理事長の部屋に来たときに赤の電光掲示板が表示している整数  $P$  である。  $0 \leq P \leq 999\,999\,999$  を満たす。この関数は整数  $Y$  を返さなければならない。これは、Bさんが「Aさんに伝えられた偶数すべての和」と「Bさんに伝えられた奇数すべての和」の合計として  $Y$  を答えることを表す。このとき、 $Y$  が正しい答えであれば正解、そうでなければ不正解 [3] と判定され、プログラムの実行は終了される。

関数 `GameA` および `GameB` は合計で高々100回呼び出される。

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Aさん側、Bさん側として2個のプロセスとして実行されるので、Aさん側とBさん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。

## 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、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 playerA.cpp playerB.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。

## 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。

```
T N  
C1 X1  
:  
CN XN
```

$N$  は  $K$  理事長が  $A$  さんまたは  $B$  さんに伝える整数の個数を表す。

文字  $C_i$  ( $1 \leq i \leq N$ ) は  $A$  または  $B$  である。  $C_i$  が  $A$  または  $B$  である場合、関数 `InitA` と `InitB` が呼び出された後に  $i$  番目に呼び出される関数がそれぞれ `GameA` または `GameB` であること、そのとき引数  $X$  の値として  $X_i$  が与えられることを表す。

## 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。

- 正解の場合、“Accepted” と出力される (引用符は実際には出力されない。以下同様である)。
- 不正解の場合、不正解の種類が「実装の詳細」の節に書かれた番号によって “Wrong Answer [1]” のように出力される。さらに、不正解 [3] については、関数 `AnswerB` が返した整数  $Y$  の値が “Wrong Answer [3] : Y = 4” のように出力される。

## 制約

- $1 \leq N \leq 100$ .
- $1 \leq X_i \leq 100$  ( $1 \leq i \leq N$ ).



## 小課題

- (30 点)  $T = 1$ . さらに  $N = 2$  であり,  $C_1$  は文字 B,  $C_2$  は文字 A である. すなわち, 関数 `InitA` および `InitB` が呼び出された後に, `GameB`, `GameA`, `AnswerB` という順で関数が呼び出される.
- (70 点)  $T = 2$ .

## 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示す.

| 入力  |
|-----|
| 2 6 |
| A 6 |
| B 5 |
| B 4 |
| A 3 |
| A 2 |
| B 1 |

| A さん側                     |     | B さん側                      |     |
|---------------------------|-----|----------------------------|-----|
| 呼び出し                      | 戻り値 | 呼び出し                       | 戻り値 |
| <code>InitA(2)</code>     | なし  |                            |     |
|                           |     | <code>InitB(2)</code>      | なし  |
| <code>GameA(0, 6)</code>  | 6   |                            |     |
|                           |     | <code>GameB(6, 5)</code>   | 5   |
|                           |     | <code>GameB(6, 4)</code>   | 60  |
| <code>GameA(60, 3)</code> | 53  |                            |     |
| <code>GameA(60, 2)</code> | 530 |                            |     |
|                           |     | <code>GameB(530, 1)</code> | 601 |
|                           |     | <code>AnswerB(530)</code>  | 14  |

この例では, 「A さんに伝えられた偶数すべての和」は  $6 + 2 = 8$  であり, 「B さんに伝えられた奇数すべての和」は  $5 + 1 = 6$  であるから,  $8 + 6 = 14$  が正解となる.