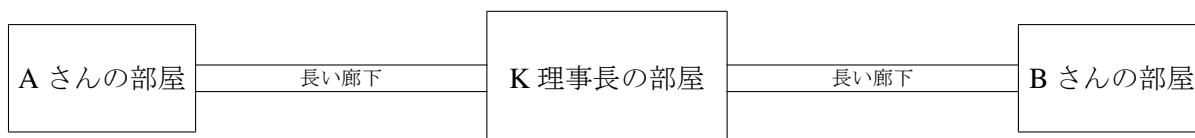




メッセンジャー (Messenger)

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 理事長と「メッセンジャー・ゲーム」を行うことになった。以下にこのゲームのルールを説明する。

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。



ゲームの開始時に、K 理事長は電話で A さんにある正の整数 X を伝える。ゲームの目的は、A さんと B さんが直接の連絡をとらずに、A さんが B さんに X の値を正しく伝えることである。

K 理事長の部屋には 4×4 のマス目がある。 i 行 j 列のマスを (i, j) と表す ($1 \leq i \leq 4, 1 \leq j \leq 4$)。マス $(1, 1)$ が左上の隅、マス $(1, 4)$ が右上の隅、マス $(4, 1)$ が左下の隅、マス $(4, 4)$ が右下の隅である。ゲームの開始時、1 つの駒がいずれかのマスに置かれる。以後、K 理事長は駒が置かれているマスを変えることはない。

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繰り返される。A さんまたは B さんは、K 理事長の部屋に来るたびに、駒を上下左右のいずれかのマスに動かさなければならない。駒を動かしたら、A さんまたは B さんは自分の部屋に戻る。B さんは、K 理事長の部屋に来たときにも X の値がわかったならば、駒を動かさず代わりにそれを K 理事長に答えることができる。正しい X の値を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。ただし、K 理事長は 2 人を部屋に呼ぶ順序をゲームの開始時に決めており、また、K 理事長が A さんまたは B さんの一方を 100 回より多く連続して部屋に呼ぶことはないことが保証されている。K 理事長が A さんと B さんを合計 10000 回部屋に呼んだ後に A さんまたは B さんが自分の部屋に戻った段階で B さんが X の値を答えていない場合は、不正解としてゲームは終了される。

課題

A さんと B さんの戦略を実装し、上で説明された「メッセンジャー・ゲーム」で正解できるプログラムを作成せよ。

実装の詳細

あなたは同じプログラミング言語で 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.c` または `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、以下のルーチンを実装していなければならない。

- `void InitA(int T, int X)`

このルーチンは、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。引数 X は、A さんが B さんに伝えるべき正の整数 X である。

- `int GameA(int I, int J)`

このルーチンは、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 I, J は、A さんが K 理事長の部屋に来たときに駒があるマスを表す整数の組 (I, J) である。 $1 \leq I \leq 4$ および $1 \leq J \leq 4$ を満たす。

このルーチンは整数 $-1, -2, -3, -4$ のいずれかを返さなければならない。それ以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。これらは A さんが以下の行動を行うことを表す。

- -1 : 駒を上マス $(I-1, J)$ に動かす。
- -2 : 駒を下マス $(I+1, J)$ に動かす。
- -3 : 駒を左マス $(I, J-1)$ に動かす。
- -4 : 駒を右マス $(I, J+1)$ に動かす。

駒が盤の外に出るような動きを指定した場合は、**不正解 [2]** と判定され、プログラムの実行は終了される。

2つ目のファイルは `playerB.c` または `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、以下のルーチンを実装していなければならない。

- `void InitB(int T)`

このルーチンは、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 I, int J)`

このルーチンは、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 I, J は、B さんが K 理事長の部屋に来たときに駒があるマスを表す整数の組 (I, J) である。 $1 \leq I \leq 4$ および $1 \leq J \leq 4$ を満たす。

このルーチンは整数 $-1, -2, -3, -4$ のいずれか、または正の整数 Y を返さなければならない。それ以外の値を返した場合は、**不正解 [3]** と判定され、プログラムの実行は終了される。これらは B さんが以下の行動を行うことを表す。

- -1 : 駒を上マス $(I-1, J)$ に動かす。
- -2 : 駒を下マス $(I+1, J)$ に動かす。
- -3 : 駒を左マス $(I, J-1)$ に動かす。



- -4 : 駒を右のマス $(I, J + 1)$ に動かす.

駒が盤の外に出るような動きを指定した場合は、**不正解 [4]** と判定され、プログラムの実行は終了される。

- 正の整数 $Y : K$ 理事長に、 X の値は Y である、と答える。このとき、 $X = Y$ であれば**正解**、そうでなければ**不正解 [5]** と判定され、プログラムの実行は終了される。

ルーチン **GameA** および **GameB** は、一方が連続して 100 回より多く呼び出されることはない。また、**GameA** と **GameB** が合計 10 000 回呼び出された段階で **GameB** が正の整数を返していない場合は、**不正解 [6]** と判定され、プログラムの実行は終了される。

内部での使用のために他のルーチンを実装したり、グローバル変数を宣言したり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ルーチンを **static** 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.c** または **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。

- C の場合

```
gcc -O2 -lm grader.c playerA.c playerB.c -o grader
```

- C++ の場合

```
g++ -O2 -lm grader.cpp playerA.cpp playerB.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。

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 T, X, I_0, J_0 が空白を区切りとして書かれており、小課題の番号が T 、AさんがBさんに伝えるべき整数が X 、ゲームの開始時の駒の位置が (I_0, J_0) であることを表す。
- 2行目には2種類の文字 **A, B** からなる、長さがちょうど10000の文字列が書かれており、 k 文字目 ($1 \leq k \leq 10000$) が **A** または **B** である場合、ルーチン `InitA` と `InitB` が呼び出された後に k 番目に呼び出されるルーチンがそれぞれ `GameA` または `GameB` であることを表す。

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[5]については、正しい X の値およびルーチン `GameB` が返した正の整数 Y の値が“Wrong Answer [5] : $X = 2, Y = 3$ ”のように出力される。

制限

すべての入力データは以下の条件を満たす。

- $1 \leq X \leq 1000000000$.

小課題

小課題1 [20点]

- $T = 1$ を満たす。
- 最初に **K** 理事長の部屋に呼ばれるのは **A** さんであり、**A** さんと **B** さんは交互に **K** 理事長の部屋に呼ばれる。すなわち、ルーチン `InitA` および `InitB` が呼び出された後に、`GameA, GameB, GameA, GameB, ...` という順でルーチンが呼び出される。

小課題2 [20点]

- $T = 2$ を満たす。
- ゲームの開始時の駒の位置は $(I_0, J_0) = (1, 1)$ である。すなわち、ルーチン `InitA` および `InitB` が呼び出された後に、まずルーチン `GameA` または `GameB` が、 $I = 1, J = 1$ を引数として呼び出される。



小課題 3 [60 点]

- $T = 3$ を満たす.

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、それに対応するルーチンの呼び出しの例を以下に示す.

入力
2 6 1 1
ABBAABAAABBBBBBABBAABBBABBBBBAAAAABABAABAABBBBBABAAAA (以下省略)

A さん側		B さん側	
呼び出し	返り値	呼び出し	返り値
InitA(2, 6)	なし		
		InitB(2)	なし
GameA(1, 1)	-2		
		GameB(2, 1)	-4
		GameB(2, 2)	-2
GameA(3, 2)	-3		
GameA(3, 1)	-1		
		GameB(2, 1)	6

この例は、必ずしも意味のあるやりとりを記述しているとは限らないことに注意せよ.



プレゼント (Presents)

JOI 学園では毎年、ホワイトデーの時期に合わせて、お菓子のプレゼント交換会が行われる。今年のプレゼント交換会には、1 から N までの学生番号が付いた N 人の生徒が参加する。それぞれの生徒は自分以外の誰か 1 人の生徒のためにクッキーまたはケーキのいずれかを作る。番号 i の生徒は番号 A_i の生徒に、作ったお菓子を B_i 個プレゼントする。

生徒によっては、自分が作ったものと同じ種類のお菓子をプレゼントされて味の研究をしたいという人もいれば、自分が作ったものと違う種類のお菓子(クッキーを作ったならケーキ、ケーキを作ったならクッキー)をプレゼントされて楽しみたいという人もいる。番号 i の生徒は、自分が作ったものと同じ種類のお菓子を 1 個もらうごとに「嬉しさ」が C_i ポイント加算され、自分が作ったものと違う種類のお菓子を 1 個もらうごとに「嬉しさ」が D_i ポイント加算される。 N 人の生徒がクッキーまたはケーキのいずれを作るかを上手く選んだとき、 N 人の生徒の「嬉しさ」の合計は最大でいくつになりうるだろうか。

課題

生徒がお菓子をプレゼントする相手と個数および「嬉しさ」の情報が与えられたとき、「嬉しさ」の合計の最大値を求めるプログラムを作成せよ。

入力

標準入力から以下の入力を読み込め。

- 1 行目には整数 N が書かれており、JOI 学園の生徒の人数を表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には整数 A_i, B_i, C_i, D_i が空白を区切りとして書かれており、番号 i の生徒は、番号 A_i ($1 \leq A_i \leq N, A_i \neq i$) の生徒に B_i 個のお菓子をプレゼントすること、自分が作ったお菓子と同じ種類のお菓子をもらったときに得る「嬉しさ」が C_i ポイント、違う種類のお菓子をもらったときに得る「嬉しさ」が D_i ポイントであることを表す。

出力

標準出力に、 N 人の生徒の「嬉しさ」の合計の最大値を 1 行で出力せよ。

制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 100\,000$.



- $1 \leq B_i \leq 1\,000\,000$ ($1 \leq i \leq N$).
- $1 \leq C_i \leq 1\,000\,000$ ($1 \leq i \leq N$).
- $1 \leq D_i \leq 1\,000\,000$ ($1 \leq i \leq N$).

小課題

小課題 1 [10 点]

- $N \leq 16$ を満たす.

小課題 2 [20 点]

- $N \leq 5\,000$ を満たす.

小課題 3 [70 点]

追加の制限はない.

入出力例

入力例 1	出力例 1
7	257
3 3 6 5	
7 2 8 8	
4 5 3 9	
1 8 7 2	
1 8 8 4	
3 7 4 5	
2 5 1 2	

この例では、たとえば、番号 1, 2, 5, 6 の生徒がクッキー、番号 3, 4, 7 の生徒がケーキを作ることで、

- 番号 1 の生徒はクッキー 8 個とケーキ 8 個をもらうので、「嬉しさ」は 88,
- 番号 2 の生徒はケーキ 5 個をもらうので、「嬉しさ」は 40,
- 番号 3 の生徒はクッキー 10 個をもらうので、「嬉しさ」は 90,
- 番号 4 の生徒はケーキ 5 個をもらうので、「嬉しさ」は 35,
- 番号 5 の生徒はお菓子をもらえないので、「嬉しさ」は 0,



- 番号 6 の生徒はお菓子をもらえないので、「嬉しさ」は 0,
 - 番号 7 の生徒はクッキー 2 個をもらうので、「嬉しさ」は 4,
- となり、「嬉しさ」の合計が 257 となる.



宇宙船 (Spaceships)

宇宙の遙か彼方、とある銀河には、文明が発達した N 個の星がある。星には 1 から N までの番号がついている。それぞれの星は 1 つの宇宙船を管理している。宇宙船は、ある他の星へ行くために使われている状態、または、使われていない状態のいずれかにある。星 a が管理する宇宙船が星 b へ行くために使われている状態にあるとすると、宇宙船は星 a と星 b の間を何回も往復している。宇宙船が星 a から星 b へ行くとき、一般旅客は宇宙船に乗って星 a から星 b へ行くことができるが、宇宙船が星 b から星 a へ戻るときは、燃料の問題や荷物をのせるなどの都合により、一般旅客は乗船できない。また、星 a が管理する宇宙船が使われていない状態にあるとき、その宇宙船は星 a で待機している。

今、すべての宇宙船が使われていない状態にある。今後の宇宙船の状態の変更のスケジュールが決まっている。状態の変更は、次のいずれかの種類である。

- 使われていない状態にある星 a が管理する宇宙船を、星 b へ行くために使われている状態にする。ただし、これは一般旅客が宇宙船に何回か乗って星 b から星 a へ行くことができないときにのみ行われる。
- 使われている状態にある星 a が管理する宇宙船を、使われていない状態にする。

この銀河を旅行する計画を立てているある 2 人は、待ち合わせの予定を考えるため、次の形式の質問を何個か用意した。

- スケジュールのある時点で、1 人が星 a に、もう 1 人が星 b にいるとしたとき、2 人が一般旅客として宇宙船を使って合流することができるか、さらに、合流することができるならば、どの星で合流するのが宇宙船の使用回数が最も少なくなるか。すなわち、星 c であって、一般旅客が宇宙船に何回か乗って星 a から星 c へも星 b から星 c へも行くことができるようなものは存在するか、さらに、存在するならば、星 a から星 c へ行くための宇宙船の使用回数と星 b から星 c へ行くための宇宙船の使用回数の合計を最小にするような星 c はどれか。

優秀なプログラマであるあなたは、2 人の質問すべてに対する答えを求めることを求められた。

課題

今後の宇宙船の状態の変更のスケジュールと質問が時系列順に与えられたとき、質問に答えるプログラムを作成せよ。

入力

標準入力から以下の入力を読み込め。

- 1 行目には整数 N, Q が空白を区切りとして書かれており、星の個数が N 、状態の変更の回数と質問の個数の合計が Q であることを表す。



- 続く Q 行は状態の変更と質問を時系列順に表している。これらのうちの i 行目 ($1 \leq i \leq Q$) には2個または3個の整数が空白を区切りとして書かれている。1個目の整数を T_i とすると、以下のいずれかである。

(i) $T_i = 1$ のとき。

この行には整数 T_i, A_i, B_i が書かれており、次の状態の変更を表す：星 A_i が管理する宇宙船を、星 B_i へ行くために使われている状態にする。

$1 \leq A_i \leq N, 1 \leq B_i \leq N, A_i \neq B_i$ であること、この時点で星 A_i が管理する宇宙船は使われていない状態にあること、この時点で一般旅客が宇宙船に何回か乗って星 B_i から星 A_i へ行くことはできないことが保証される。

(ii) $T_i = 2$ のとき。

この行には整数 T_i, A_i が書かれており、次の状態の変更を表す：星 A_i が管理する宇宙船を、使われていない状態にする。

$1 \leq A_i \leq N$ であること、この時点で星 A_i が管理する宇宙船は使われている状態にあることが保証される。

(iii) $T_i = 3$ のとき。

この行には整数 T_i, A_i, B_i が書かれており、次の質問を表す：この時点で、1人が星 A_i に、もう1人が星 B_i にいるとしたとき、2人が一般旅客として宇宙船を使って合流することができるか、さらに、合流することができるならば、どの星で合流するのが宇宙船の使用回数が最も少なくなるか。

$1 \leq A_i \leq N, 1 \leq B_i \leq N, A_i \neq B_i$ であることが保証される。

出力

標準出力に、それぞれの質問ごとに、

- 合流することができるならば、宇宙船の使用回数を最も少なくするための合流する星の番号、
- 合流することができないならば、整数 -1

を順に1行ずつ出力せよ。

制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 1\,000\,000$.
- $1 \leq Q \leq 1\,000\,000$.



小課題

小課題 1 [10 点]

以下の条件を満たす.

- $N \leq 5000$.
- $Q \leq 5000$.

小課題 2 [30 点]

- $T_i \neq 2$ ($1 \leq i \leq Q$) を満たす.

小課題 3 [60 点]

追加の制限はない.

入出力例

入力例 1	出力例 1
6 5	-1
1 2 4	4
3 2 6	
1 4 3	
1 6 4	
3 2 6	

この例では、状態の変更と質問は順に以下のようにになっている.

- 星 2 が管理する宇宙船が、星 4 へ行くために使われる状態になる.
- この時点で 2 人が星 2 と星 6 にいた場合、合流することはできないので、-1 を出力する.
- 星 4 が管理する宇宙船が、星 3 へ行くために使われる状態になる.
- 星 6 が管理する宇宙船が、星 4 へ行くために使われる状態になる.
- この時点で 2 人が星 2 と星 6 にいた場合、星 3 または星 4 で合流することができる. 使用する宇宙船の回数を最小にするためには星 4 で合流するべきであるため、4 を出力する.



入力例 2	出力例 2
8 36	5
1 1 2	2
1 6 5	-1
1 7 8	1
3 5 6	1
1 5 4	2
1 8 1	-1
3 7 2	5
3 3 8	4
3 1 8	-1
1 3 2	5
1 4 1	2
3 8 5	5
3 4 3	3
2 4	5
3 6 8	4
1 2 5	3
3 6 8	5
2 8	6
3 1 4	
3 6 8	
3 6 3	
2 3	
3 1 2	
1 4 3	
3 2 6	
1 8 3	
3 1 7	
3 1 6	
3 5 4	
2 2	
2 5	
1 3 6	
1 2 7	
3 1 4	
3 1 5	
3 6 7	