



切符の手配 (Arranging Tickets)

JOI 国には N 個の駅 $1, 2, \dots, N$ がある。これらは、円形の線路に沿って時計回りに順に並んでいる。

鉄道に乗るための切符は N 種類あり、それぞれ $1, 2, \dots, N$ の番号が付けられている。切符 i ($1 \leq i \leq N-1$) を 1 枚用いると、駅 i から駅 $i+1$ 、または駅 $i+1$ から駅 i へ 1 人が移動できる。また、切符 N を 1 枚用いると、駅 1 から駅 N 、または駅 N から駅 1 へ 1 人が移動できる。これらの切符は、 N 種類の切符をちょうど 1 枚ずつ含む N 枚セットの形でのみ販売されている。

あなたは、JOI 国の旅行会社で切符の手配をする業務を行っている。

本日、あなたは M 件の依頼を受けた。 i 番目 ($1 \leq i \leq M$) の依頼は、駅 A_i から駅 B_i に C_i 人が移動したい、というものである。ただし、移動する C_i 人は、全員が同じ経路で移動する必要はない。

これらの依頼をすべて処理するために、切符を最小で何セット購入する必要があるかを求めたい。

課題

駅の数および依頼の情報が与えられたとき、購入すべき切符のセット数の最小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データを読み込め。

- 1 行目には、2 個の整数 N, M が空白を区切りとして書かれている。これは、JOI 国に N 個の駅があり、あなたが M 件の依頼を受けたことを表す。
- 続く M 行のうちの i 行目 ($1 \leq i \leq M$) には、3 個の整数 A_i, B_i, C_i が空白を区切りとして書かれている。これは、 i 番目の依頼が駅 A_i から駅 B_i に C_i 人が移動したい、というものであることを表す。

出力

標準出力に購入すべき切符のセット数の最小値を 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。

- $3 \leq N \leq 200\,000$.
- $1 \leq M \leq 100\,000$.



- $1 \leq A_i \leq N$ ($1 \leq i \leq M$).
- $1 \leq B_i \leq N$ ($1 \leq i \leq M$).
- $1 \leq C_i \leq 1\,000\,000\,000$ ($1 \leq i \leq M$).
- $A_i \neq B_i$ ($1 \leq i \leq M$).

小課題

この課題では小課題は全部で 5 個ある。各小課題の配点および追加の制限は以下の通りである。

小課題 1 [10 点]

- $N \leq 20$.
- $M \leq 20$.
- $C_i = 1$ ($1 \leq i \leq M$).

小課題 2 [35 点]

- $N \leq 300$.
- $M \leq 300$.
- $C_i = 1$ ($1 \leq i \leq M$).

小課題 3 [20 点]

- $N \leq 3\,000$.
- $M \leq 3\,000$.
- $C_i = 1$ ($1 \leq i \leq M$).

小課題 4 [20 点]

- $C_i = 1$ ($1 \leq i \leq M$).

小課題 5 [15 点]

追加の制限はない。



入出力例

入力例 1	出力例 1
3 3 1 2 1 2 3 1 3 1 1	1

全員が時計回りに移動すると、それぞれの切符は 1 枚ずつ必要となるので、1 セット購入すればよい。

入力例 2	出力例 2
3 2 1 2 4 1 2 2	3

次のように移動することで、それぞれの切符は 3 枚ずつ必要となる。

- 最初の依頼において、3 人が時計回りに、1 人が反時計回りに移動する。
- 2 番目の依頼において、2 人が反時計回りに移動する。

したがって、3 セット購入すればよい。

2 セットで移動することはできないので、3 を出力する。

入力例 3	出力例 3
6 3 1 4 1 2 5 1 3 6 1	2

例えば、切符を 2 セット購入して、以下のように配ればよい。

- 駅 1 から駅 4 に移動したい人に、切符 1, 2, 3 を配る。
- 駅 2 から駅 5 に移動したい人に、切符 1, 6, 5 を配る。
- 駅 3 から駅 6 に移動したい人に、切符 3, 4, 5 を配る。

1 セットで移動することはできないので、2 を出力する。



壊れた機器 (Broken Device)

考古学者の Anna と Bruno はイランで遺跡調査を行っている。

二人の役割分担は、Anna が遺跡に向いて発掘し、Bruno はベースキャンプで発掘結果の解析を行うというものである。

発掘は $Q (= 1000)$ 日間に渡って行われる。毎日、Anna は通信機器を用いて Bruno に発掘結果を伝える。1 日の発掘結果は、1 つの整数 X で表される。

Anna は通信機器を 1 日に 1 回だけ使うことができる。通信機器は 0 または 1 からなる長さ $N (= 150)$ の数列を一度に送信することができる。

ところが、通信機器が壊れてしまい、送信する長さ N の数列のうち、機能しない箇所が発生するようになった。機能しない箇所については、どのように設定しても値が 0 として送信されてしまう。Anna は通信機器のどの箇所が機能しないかを送信前に確認できるが、Bruno は確認できない。また、機能しない箇所やその個数は、毎日、異なるかもしれない。

このままでは遺跡調査に遅れが出てしまうので、Anna と Bruno はイランで開催される国際的なプログラミングコンテストの選手候補であるあなたに、発掘結果を伝えるためのプログラムを書くことを依頼した。

課題

Anna と Bruno の通信を実現するために、次の 2 つのプログラムを作成せよ。

- 1 つ目のプログラムは、送信する数列の長さ N 、伝えるべき整数 X 、機能しない箇所の個数 K とその位置の情報 P が与えられたとき、Anna が送信する数列 S の情報を設定する。
- 2 つ目のプログラムは、Bruno が受信する数列 A の情報が与えられたとき、整数 X を復元する。

通信機器が機能している箇所においては、数列 S と数列 A は同じ値である。通信機器が機能しない箇所においては、数列 S の値に関わらず、数列 A の値は 0 である。

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 `Anna.c` または `Anna.cpp` という名前である。このファイルは Anna が送信する数列を設定するファイルであり、以下のルーチンを実装していなければならない。プログラムは `Annalib.h` をインクルードすること。

- `void Anna(int N, long long X, int K, int P[])`

この関数は各テストケースにおいて $Q = 1000$ 回だけ呼び出される。

- 引数 N は送信する数列の長さを表す。



- 引数 X は伝えるべき整数を表す.
- 引数 K は機能しない箇所の個数を表す.
- 引数 $P[]$ は長さ K の数列であり, 機能しない箇所を表す.

関数 `Anna` 中では以下の関数を呼び出さなければならない.

★ `void Set(int pos, int bit)`

この関数は, 通信機器に数列 S の一箇所を設定する操作を表す.

- ◇ 引数 `pos` は設定する数列の場所を表す. `pos` は 0 以上 $N-1$ 以下の整数値でなければならない. 場所を 0 から数えることに注意せよ. この範囲外の値を指定して呼び出した場合, 不正解 [1] と判定される. また, 同じ引数 `pos` で 2 回以上呼び出すことはできない. 同じ引数で 2 回呼び出した場合, 不正解 [2] と判定される.
- ◇ 引数 `bit` は数列の `pos` 番目の箇所に設定する値を表す. `bit` は 0 または 1 でなければならない. これら以外の値を指定して呼び出した場合, 不正解 [3] と判定される.

関数 `Anna` 中では, 関数 `Set` をちょうど N 回呼び出さなければならない. 関数 `Anna` の実行の終了時に関数 `Set` の呼び出し回数が N でなかった場合, 不正解 [4] と判定される.

`Anna` の呼び出しが不正解と判定された場合, その時点でプログラムは終了する.

2 目目のファイルは `Bruno.c` または `Bruno.cpp` という名前である. このファイルは発掘結果を表す整数の復元方法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. プログラムは `Brunolib.h` をインクルードすること.

• `long long Bruno(int N, int A[])`

この関数は各テストケースにおいて $Q = 1000$ 回だけ呼び出される.

- 引数 N は受信した数列の長さを表す.
- 引数 A は長さ N の数列であり, 受信した数列を表す.
- 関数 `Bruno` は復元した X の値を戻り値として返さなければならない.

採点の手順

採点は以下の手順で行われる. 不正解と判定された場合はその時点でプログラムは終了される.

- (1) `cnt = 0` とする.
- (2) 関数 `Anna` を 1 回呼び出す.
- (3) 関数 `Anna` 中で設定された数列を S とする. S のうち, P に含まれる箇所を 0 に変更したものを引数 A の値とし, 関数 `Bruno` を 1 回呼び出す.
- (4) `cnt = cnt + 1` とする. `cnt < Q` ならば (2) へ戻る. `cnt = Q` ならば (5) へ進む.
- (5) 採点される.



重要な注意

- 実行時間計測・使用メモリ計測の対象となるのは、「採点の手順」における手順 (1), (2), (3), (4) である。
- 手順 (2) における Anna の呼び出し、および手順 (3) における Bruno の呼び出しにおいて、不正解と判定されたり、実行時エラーとなったりしてはならない。
- 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 Anna 側、Bruno 側として 2 個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。
- 関数 Anna と Bruno はプロセスの実行中に $Q = 1000$ 回ずつ呼びだされるので、適切に変数を初期化しなければならない。
- 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 `Anna.c` および `Bruno.c`、または `Anna.cpp` および `Bruno.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 -std=c11 -O2 -o grader grader.c Anna.c Bruno.c -lm
```

- C++ の場合

```
g++ -std=c++14 -O2 -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1つの整数 Q が書かれている。
- 続いて、 Q 個のクエリの情報が与えられる。
- 各クエリの情報は2行からなる。各クエリの情報は以下の通り。
 - 1行目には3個の整数 N, X, K が空白を区切りとして書かれている。これは送信する数列の長さが N 、Anna が伝えるべき整数が X 、機能しない箇所の個数が K であることを表す。
 - 2行目には K 個の整数 P_0, P_1, \dots, P_{K-1} が空白を区切りとして書かれている。これは、各 i ($0 \leq i \leq K-1$) について、送信する数列の P_i 番目の箇所が機能しないことを表す。

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出力する（引用符は実際には出力されない）。

- プログラムの実行中にいずれかの不正解と判定された場合、不正解の種類が “Wrong Answer [1]” のように出力され、実行が終了される。
- 各 Anna の呼び出しが不正解と判定されなかった場合、“Accepted” と出力される。また、 L^* の値が出力される。 L^* の値については、採点基準の項を参照せよ。

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。

制限

すべての入力データは以下の条件を満たす。

- $Q = 1000$.
- $N = 150$.
- $0 \leq X \leq 1\,000\,000\,000\,000\,000\,000$.
- $1 \leq K \leq 40$.
- $0 \leq P_i \leq N - 1$ ($0 \leq i \leq K - 1$).
- $P_i < P_{i+1}$ ($0 \leq i \leq K - 2$).



採点基準

- この課題の全てのテストケースにおける以下の値の最小値を L^* とする。
 - $K \leq L$ を満たす全てのクエリについて Bruno が X と等しい整数を返すような最大の整数 $L \leq 40$
- このとき、この課題の得点は以下のように与えられる。
 - $L^* = 0$ のとき、0 点
 - $1 \leq L^* \leq 14$ のとき、8 点
 - $15 \leq L^* \leq 37$ のとき、 $(L^* - 15) \times 2 + 41$ 点
 - $38 \leq L^* \leq 40$ のとき、 $(L^* - 38) \times 5 + 90$ 点

やり取りの例

grader 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。この例では $Q = 2, N = 3$ なので、課題の制約は満たしていないことに注意せよ。

入力例	ルーチンの呼び出しの例			
	呼び出し	戻り値	呼び出し	戻り値
2	Anna(...)			
3 14 1			Set(0,0)	
2				(なし)
3 9 2			Set(1,0)	
0 1				(なし)
			Set(2,1)	
				(なし)
		(なし)		
	Bruno(...)			
		14		
	Anna(...)			
			Set(0,0)	
				(なし)
			Set(1,1)	
				(なし)
			Set(2,1)	
				(なし)
	Bruno(...)			
		9		



このとき、 $\text{Anna}(\dots)$, $\text{Bruno}(\dots)$, $\text{Anna}(\dots)$, $\text{Bruno}(\dots)$ に渡される引数はそれぞれ次の通りである。

引数	$\text{Anna}(\dots)$	$\text{Bruno}(\dots)$	$\text{Anna}(\dots)$	$\text{Bruno}(\dots)$
N	3	3	3	3
X	14		9	
K	1		2	
P	{2}		{0, 1}	
A		{0, 0, 0}		{0, 0, 1}



鉄道旅行 (Railway Trip)

JOI 鉄道は 1 本の鉄道路線を運営している。JOI 鉄道線には一直線上に並んだ N 個の駅があり、それぞれ 1 以上 N 以下の整数で番号がつけられている。各 $i (1 \leq i \leq N - 1)$ に対して、駅 i と駅 $i + 1$ の間は線路で結ばれている。

JOI 鉄道は、 K 種類の列車を双方向に走らせている。列車種別には 1 以上 K 以下の整数で番号がつけられている。また、各駅にはレベルと呼ばれる 1 以上 K 以下の整数が 1 つ定まっており、駅 $i (1 \leq i \leq N)$ のレベルは L_i である。両端の駅、すなわち駅 1 および駅 N のレベルは K である。

種別 $j (1 \leq j \leq K)$ の列車はレベルが j 以上の駅すべてに停車し、それ以外の駅には停車しない。両端の駅、すなわち駅 1 および駅 N のレベルは K なので、すべての列車が停車する。

毎日多くの利用者が JOI 鉄道線を利用する。JOI 鉄道線を利用する際には、途中で目的地と反対の方向の列車に乗ったり、目的地を通り過ぎても構わないが、旅程の最後では目的地に停車する必要がある。利用者は、駅での停車を極端に嫌うため、途中で通過する駅の数や、乗り換えの回数などによらず、できるだけ途中の停車の回数が少なくなるような経路で駅間を移動しようとする。ただし、乗り換えのための停車は、1 回の停車と数える。途中の停車の回数には、出発駅や目的地における停車は含めない。

あなたの仕事は、利用者の出発駅と目的地が与えられたときに、この 2 駅の間を移動するための途中の停車の回数の最小値を求めよという質問に回答するプログラムを作成することである。

課題

JOI 鉄道線の情報と、各利用者の出発駅、目的地が与えられたとき、各利用者について、出発駅から目的地まで移動するための途中の停車の回数の最小値を求めよという質問に回答するプログラムを作成せよ。

入力

標準入力から以下のデータを読み込め。

- 1 行目には、整数 N, K, Q が空白を区切りとして書かれている。これは、JOI 鉄道線には N 個の駅、 K 種類の列車があり、2 駅間の移動の質問が Q 個与えられることを表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 L_i が書かれている。これは、駅 i のレベルが L_i であることを表す。
- 続く Q 行のうちの k 行目 ($1 \leq k \leq Q$) には、整数 A_k, B_k が空白を区切りとして書かれている。これは、 k 番目の利用者の出発駅、目的地がそれぞれ駅 A_k 、駅 B_k であることを表す。

出力

標準出力に Q 行で出力せよ。 k 行目 ($1 \leq k \leq Q$) には、駅 A_k から駅 B_k まで移動するときの、途中の停車の回数の最小値を出力せよ。



制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 100\,000$.
- $1 \leq K \leq N$.
- $1 \leq Q \leq 100\,000$.
- $1 \leq L_i \leq K$ ($1 \leq i \leq N$).
- $1 \leq A_k \leq N$ ($1 \leq k \leq Q$).
- $1 \leq B_k \leq N$ ($1 \leq k \leq Q$).
- $A_k \neq B_k$ ($1 \leq k \leq Q$).

小課題

この課題では小課題は全部で 4 個ある。各小課題の配点および追加の制限は以下の通りである。

小課題 1 [5 点]

- $N \leq 100$.
- $K \leq 100$.
- $Q \leq 50$.

小課題 2 [15 点]

- $Q \leq 50$.

小課題 3 [25 点]

- $K \leq 20$.

小課題 4 [55 点]

追加の制限はない。



入出力例

入力例 1	出力例 1
9 3 3	1
3	3
1	0
1	
1	
2	
2	
2	
3	
3	
2 4	
4 9	
6 7	

この入力例では、2 駅間の移動の質問が 3 個与えられる。

- 1 番目の質問は、駅 2 から駅 4 まで移動するものである。このとき、駅 2 から駅 4 まで種別 1 の列車を利用すると、途中停車する駅は駅 3 の 1 つのみとなる。
- 2 番目の質問は、駅 4 から駅 9 まで移動するものである。このとき、まず駅 4 から駅 5 まで種別 1 の列車を利用し、次に駅 5 から駅 1 まで種別 2 の列車を利用し、最後に駅 1 から駅 9 まで種別 3 の列車を利用すると、途中停車する駅は順に駅 5、駅 1、駅 8 の 3 つとなる。
- 3 番目の質問は、駅 6 から駅 7 まで移動するものである。このとき、駅 6 から駅 7 まで種別 2 の列車を利用すると、途中他の駅に停車せずに移動することができる。

入力例 2	出力例 2
5 2 1	1
2	
1	
1	
1	
2	
1 4	

途中で目的地の駅を通り過ぎても構わないことに注意せよ。



入力例 3	出力例 3
15 5 15	2
5	1
4	1
1	3
2	2
3	0
1	3
1	4
2	0
4	1
5	3
4	4
1	1
5	2
3	2
5	
8 1	
11 1	
5 3	
6 11	
9 12	
15 14	
15 2	
3 12	
2 1	
4 8	
15 5	
12 6	
1 13	
13 8	
14 9	