



2人の星座 (Constellation 2)

JOI ちゃんと IOI ちゃんは親友同士である。ある日、JOI ちゃんと IOI ちゃんは山の上にある展望台で天体観測をすることにした。

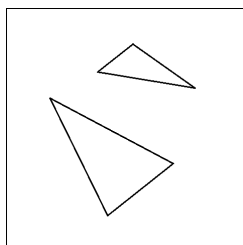
展望台では、 N 個の星を観測することができる。それぞれの星には 1 から N まで番号が付いていて、それぞれの星は、赤色、青色、黄色のいずれかの色をしている。

この展望台で観測された星は座標平面上の点で表される。この座標平面において、星 i ($1 \leq i \leq N$) に対応する点は $P_i(X_i, Y_i)$ である。座標平面上の点 P_1, \dots, P_N は、どの 2 点も異なる。また、点 P_1, \dots, P_N は、どの 3 点も同一直線上にない。

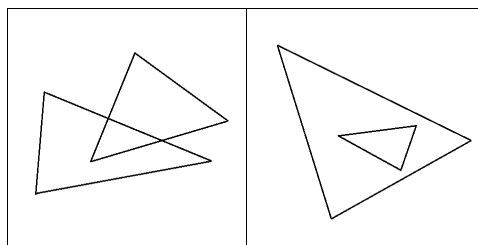
JOI ちゃんと IOI ちゃんは、**JOIOI 座** という星座を作ることにした。まず 2 人は、赤色、青色、黄色の 3 つの星を結んだ三角形を使うことを考えた。このような三角形のことを**良い三角形**と呼ぶ。

2 人は、以下の条件を満たす 2 つの良い三角形の組 (順序を問わない) を **JOIOI 座の候補** とすることにした。

- 2 つの良い三角形 (三角形の周および内部) には共有点がない。すなわち、2 つの良い三角形同士が重なったり、一方が他方に含まれることはない。



条件を満たす例



条件を満たさない例

JOI ちゃんと IOI ちゃんは、**JOIOI 座** の候補として考えられるものがいくつあるか数えることにした。ただし、**JOIOI 座** の候補を構成する 6 個の星が一致していても、良い三角形の結び方が異なる場合は、それらを別の候補として数えるものとする。

課題

展望台で観測された星の情報が与えられたとき、**JOIOI 座** の候補の総数を出力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 が書かれている. これは展望台で観測された星の個数が N であることを表す.
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には, 3 つの整数 X_i, Y_i, C_i が空白を区切りとして書かれている. これは, 星 i の座標が $P_i(X_i, Y_i)$ であることを表し, C_i は星 i の色を表している. 星 i の色は, C_i が 0 なら赤色であり, 1 なら青色であり, 2 なら黄色である.

出力

標準出力に, JOIOI 座の候補の総数を表す整数を 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.

- $6 \leq N \leq 3000$.
- $-100000 \leq X_i \leq 100000$.
- $-100000 \leq Y_i \leq 100000$.
- $0 \leq C_i \leq 2$.
- どの色の星も 1 個以上存在する.
- $P_i \neq P_j$ ($1 \leq i < j \leq N$)
- P_i, P_j, P_k は同一直線上にない ($1 \leq i < j < k \leq N$).



小課題

小課題 1 [15 点]

- $N \leq 30$ を満たす.

小課題 2 [40 点]

- $N \leq 300$ を満たす.

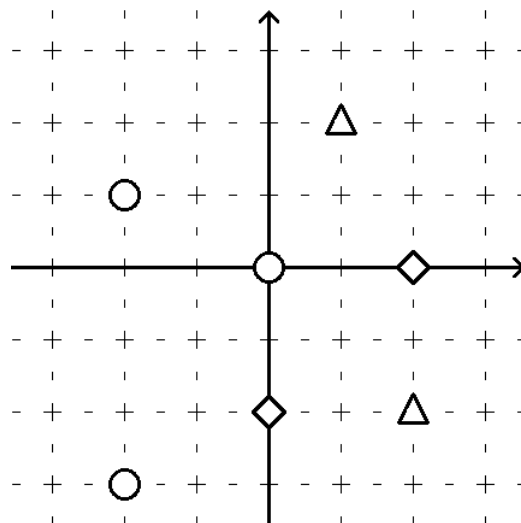
小課題 3 [45 点]

追加の制限はない.

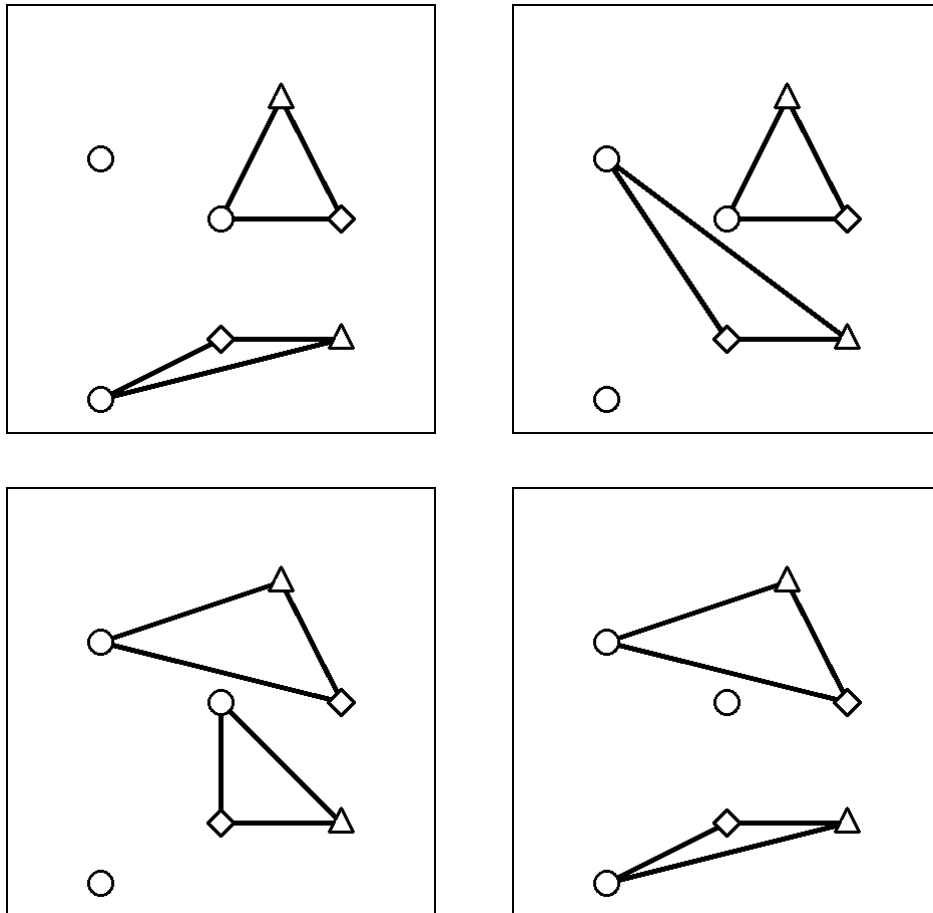
入出力例

入力例 1	出力例 1
7 0 0 0 2 0 1 1 2 2 -2 1 0 -2 -3 0 0 -2 1 2 -2 2	4

この入力例では、星の配置は下図のようになる。この図において、赤色の星は丸、青色の星は菱形、黄色の星は三角形で表す。



この入力例では、JOIOI 座の候補は以下に示す 4 つである。



条件を満たす 4 つの例



入力例 2	出力例 2
8 16 0 0 17 0 0 0 7 2 0 -7 2 -1 -1 1 -1 1 2 -6 4 1 -6 -4 1	12

入力例 3	出力例 3
21 1 20 0 4 20 0 0 22 0 5 22 0 6 25 0 8 25 0 4 26 0 11 11 1 7 12 1 14 13 1 8 15 1 15 16 1 11 17 1 18 0 2 13 2 2 16 2 2 19 4 2 18 6 2 21 8 2 24 8 2 19 10 2	7748



漢字しりとり (Kanji Shiritori)

Anna と Bruno は学校で漢字の試験を受けることになった。Anna と Bruno は漢字 0 から漢字 $N-1$ までの N 個の漢字と、単語 0 から単語 $M-1$ までの M 個の単語を知っている。単語はふたりの知っている漢字から成っており、単語 i の最初の文字は漢字 A_i で、最後の文字は漢字 B_i である。すべての単語について $A_i \neq B_i$ である。また、すべての単語の (A_i, B_i) は異なる。すなわち、 $i \neq j$ なら $(A_i, B_i) \neq (A_j, B_j)$ である。また、それぞれの単語には書くための所要時間 C_i が定まっている。

試験では、次のような問題が Q 個出題される。

問題 j : 漢字 S_j から始まり漢字 T_j で終わるような漢字しりとりを 1 つ答えよ。

すべての問題について $S_j \neq T_j$ である。また、すべての問題の (S_j, T_j) は異なる。すなわち、 $i \neq j$ なら $(S_i, T_i) \neq (S_j, T_j)$ である。

漢字しりとりとは、図 1 のように単語の最後の文字と 1 つ後の単語の最初の文字が同じになるような単語の列である。漢字 S_j から始まり漢字 T_j で終わるような漢字しりとりとは、最初の単語の最初の文字が漢字 S_j で最後の単語の最後の文字が漢字 T_j であるような漢字しりとりである。

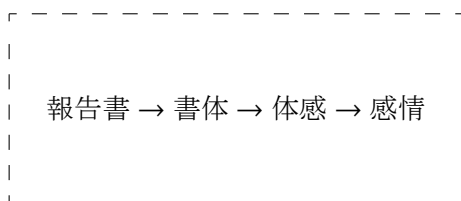


図 1: 漢字 $S_j =$ 「報」 から始まり、漢字 $T_j =$ 「情」 で終わるような漢字しりとりの例

答えとなる漢字しりとりはいくつか考えられるが、試験時間が短いため、その中で所要時間が最短となるものを答えなければならない（複数ある場合は、その中のどれを答えてもよい）。漢字しりとりを書くための所要時間は、それに含まれるすべての単語の所要時間の合計である。

試験直前になって、Bruno は単語 U_0, U_1, \dots, U_{K-1} の所要時間 $C_{U_0}, C_{U_1}, \dots, C_{U_{K-1}}$ を忘れてしまった。この K 個の単語は、偶然にも最初の文字が共通している。Anna はこのことを Bruno から伝えられたが、試験開始まで時間がなかったため、試験中に Bruno に情報を伝えることにした。試験中に Anna は机を叩く音で Bruno に 0 または 1 を送ることができる。Anna は 0 または 1 を送る回数をできるだけ少なくしたい。

はたして Bruno は試験で満点を取ることができるだろうか？

課題

漢字の個数 N 、単語の個数 M とそれぞれの単語の最初と最後の文字、問題の個数 Q とそれぞれの問題の情報、Bruno が所要時間を忘れてしまった単語の個数 K とその単語の番号が Anna と Bruno のそれぞれに与えられる。また、Anna には M 個すべての単語の所要時間が与えられ、Bruno には所要時間を忘れてしまった K 個の単語を除く $M-K$ 個の単語の所要時間が与えられる。Anna が Bruno に情報を送り、Bruno が試験で満点を取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 `Anna.c` または `Anna.cpp` という名前である。このファイルは `Anna` 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 Anna(int N, int M, int A[], int B[], long long C[], int Q, int S[], int T[], int K, int U[])`

このルーチンは、最初に1回だけ呼び出される。

- 引数 N は、漢字の個数 N である。
- 引数 M は、単語の個数 M である。
- 引数 A は、長さ M の配列であり、要素 $A[i]$ は、単語 i の最初の文字の番号 A_i である。
- 引数 B は、長さ M の配列であり、要素 $B[i]$ は、単語 i の最後の文字の番号 B_i である。
- 引数 C は、長さ M の配列であり、要素 $C[i]$ は、単語 i を書くための所要時間 C_i である。
- 引数 Q は、問題の個数 Q である。
- 引数 S は、長さ Q の配列であり、要素 $S[j]$ は、問題 j の答えの最初の文字の番号 S_j である。
- 引数 T は、長さ Q の配列であり、要素 $T[j]$ は、問題 j の答えの最後の文字の番号 T_j である。
- 引数 K は、Bruno が所要時間を忘れてしまった単語の個数 K である。
- 引数 U は、長さ K の配列であり、要素 $U[0], U[1], \dots, U[K-1]$ は、Bruno が所要時間を忘れてしまった単語の番号 U_0, U_1, \dots, U_{K-1} である。

また、プログラム中で次のルーチンを呼び出し、Bruno に0または1を送ることができる。

- `void Tap(int x)`

引数 x は、Bruno に送る0または1である。

- x は0または1でなければならない。これを満たさない場合、不正解 [1] となる。
- `Tap` の呼び出しの回数が1000回を超えた場合、不正解 [2] となる。

`Tap` 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の実行は終了される。



2つ目のファイルは `Bruno.c` または `Bruno.cpp` という名前である。このファイルは Bruno の戦略を実装したファイルであり、次のルーチンを実装していなければならない。

- `void Bruno(int N, int M, int A[], int B[], long long C[], int Q, int S[], int T[], int K, int U[], int L, int X[])`

このルーチンは、Anna が呼び出された後に、1 回だけ呼び出される。

- 引数 $N, M, A, B, Q, S, T, K, U$ は、Anna と同じである。
- 引数 C は、長さ N の配列であり、要素 $C[i]$ は、単語 i を書くための所要時間 C_i である。ただし、 i が U_0, U_1, \dots, U_{K-1} のいずれかである場合、 -1 である。
- 引数 L は Anna から送られてきた 0 または 1 の個数である。
- 引数 X は、長さ L の配列であり、Anna から 0 または 1 が $X[0], X[1], \dots, X[L-1]$ の順に送られてきたことを表す。

プログラム中では、次のルーチンを呼び出す事ができる。

- `void Answer(int w)`
 - 引数 w は、 0 以上 $M-1$ 以下の整数または -1 でなければならない。これを満たさない場合、**不正解 [3]** となる。
 - 引数 w が -1 である呼び出しが Q 回行われた後にこのルーチンが呼び出された場合、**不正解 [4]** となる。

`Answer` 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の実行は終了される。

プログラムはこのルーチンを用いて、1つの問題に答える操作を Q 回繰り返さなければならない。 $j+1$ 回目 ($0 \leq j \leq Q-1$) の操作では次を行う。

1. 問題 j の答えの単語の列に対し、列の最初の単語から順に最後の単語まで、単語の番号を引数として `Answer(w)` を呼び出す。
2. その後、`Answer(-1)` を呼び出す。

Bruno が呼び出された後、答えの判定が行われる。

- 引数 w が -1 である呼び出しの回数が Q 未満だった場合、**不正解 [5]** となる。
- ある問題の答えの列の長さが 0 だった場合、**不正解 [6]** となる。
- ある問題の答えで、ある単語の最後の文字と次の単語の最初の文字が異なる場合、**不正解 [7]** となる。
- ある問題 j の答えで、最初の単語の最初の文字が S_j でない、または最後の単語の最後の文字が T_j でない場合、**不正解 [8]** となる。
- ある問題について、答えの所要時間が最短でない場合、**不正解 [9]** となる。



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。

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 Anna.c Bruno.c -o grader
```

- C++ の場合

```
g++ -O2 grader.cpp Anna.cpp Bruno.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。

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行目には整数 N , M , Q , K が空白を区切りとして書かれており、漢字の個数が N 、単語の個数が M 、問題の個数が Q 、Bruno の忘れてしまった単語の個数が K であることを表す。
- 続く M 行のうちの $i+1$ 行目 ($0 \leq i < M$) には整数 A_i , B_i , C_i が空白を区切りとして書かれており、単語 i の最初の文字が漢字 A_i 、最後の文字が漢字 B_i 、所要時間が C_i であることを表す。
- 続く Q 行のうちの $j+1$ 行目 ($0 \leq j < Q$) には整数 S_j , T_j , Z_j が空白を区切りとして書かれており、問題 j の答えの最初の文字が漢字 S_j 、最後の文字が漢字 T_j 、最短の所要時間が Z_j であることを表す。
- 続く K 行のうちの $k+1$ 行目 ($0 \leq k < K$) には整数 U_k が書かれており、Bruno が所要時間を忘れてしまった単語が単語 U_0, U_1, \dots, U_{K-1} であることを表す。



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、Tap が呼び出された回数が “Accepted : L = 100” 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.

制限

すべての入力データは以下の条件を満たす.

- $2 \leq N \leq 300$.
- $1 \leq M \leq N \times (N - 1)$.
- $0 \leq A_i < N$ ($0 \leq i < M$).
- $0 \leq B_i < N$ ($0 \leq i < M$).
- $A_i \neq B_i$ ($0 \leq i < M$).
- $(A_i, B_i) \neq (A_j, B_j)$ ($0 \leq i < j < M$).
- $1 \leq C_i \leq 10^{16}$ ($< 2^{54}$) ($0 \leq i < M$).

- $1 \leq Q \leq 60$.
- $0 \leq S_j < N$ ($0 \leq j < Q$).
- $0 \leq T_j < N$ ($0 \leq j < Q$).
- $S_j \neq T_j$ ($0 \leq j < Q$).
- $(S_i, T_i) \neq (S_j, T_j)$ ($0 \leq i < j < Q$).
- 漢字 S_j から始まり漢字 T_j で終わる漢字しりとりが存在する ($0 \leq j < Q$).

- $1 \leq K \leq 5$.
- $0 \leq U_k < M$ ($0 \leq k < K$).
- $U_i \neq U_j$ ($0 \leq i < j < K$).
- Bruno が忘れてしまった単語の最初の文字は共通である. すなわち $A_{U_0} = A_{U_1} = \dots = A_{U_{K-1}}$.



小課題

小課題 1 [10 点]

- $Q \leq 10$ を満たす.
- それぞれの問題について、使う単語が 10 個以下であるような答えが存在する.
- Anna は 1000 回までしか Tap を呼び出すことはできない.

小課題 2 [22 点]

- Anna は 180 回までしか Tap を呼び出すことはできない.

小課題 3 [8 点]

- Anna は 160 回までしか Tap を呼び出すことはできない.

小課題 4 [40 点]

- Anna は 90 回までしか Tap を呼び出すことはできない.

小課題 5 [20 点]

- Tap の呼び出し回数のこの小課題の全てのテストケースにおける最大値を L とおく. この小課題の得点は
 - $L \leq 64$ のとき, 20 点,
 - $64 < L < 90$ のとき, $\left\lfloor \left(\frac{90-L}{90-64} \right)^2 \times 20 \right\rfloor$ 点 (ただし, $\lfloor x \rfloor$ とは x 以下の最大の整数を表す),
 - $L \geq 90$ のとき, 0 点

で与えられる.



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。

入力例	ルーチンの呼び出しの例	
	Anna 側	Bruno 側
	Anna(...)	
4 5 3 2	Tap(0)	
2 1 10	Tap(0)	
0 2 20	Tap(1)	
3 1 30	Tap(0)	
0 1 40		Bruno(...)
3 0 50		Answer(4)
3 0 50		Answer(-1)
3 1 30		Answer(2)
0 1 30		Answer(-1)
1		Answer(1)
3		Answer(0)
		Answer(-1)

この例での Tap の呼び出しは、必ずしも意味のある呼び出しとは限らないことに注意せよ。このとき、Anna(...) と Bruno(...) に渡される引数は次の通りである。

引数	Anna(...)	Bruno(...)
N	4	4
M	5	5
A	{2, 0, 3, 0, 3}	{2, 0, 3, 0, 3}
B	{1, 2, 1, 1, 0}	{1, 2, 1, 1, 0}
C	{10, 20, 30, 40, 50}	{10, -1, 30, -1, 50}
Q	3	3
S	{3, 3, 0}	{3, 3, 0}
T	{0, 1, 1}	{0, 1, 1}
K	2	2
U	{1, 3}	{1, 3}
L	—	4
X	—	{0, 0, 1, 0}

C の要素 C[1] と要素 C[3] に注意すること。



ストラップ (Straps)

JOI 君は、携帯電話に取り付けるためのストラップを N 個持っている。ストラップには 1 から N までの番号が付けられている。JOI 君は、これらのストラップのうちいくつかを携帯電話に取り付けようとしている。

JOI 君の持っているストラップは少し変わっていて、いくつかのストラップには、他のストラップを取り付けるための端子が何個か付いている。それぞれのストラップは、携帯電話に直接取り付けるか、あるいは他のストラップの端子に取り付けることができる。携帯電話に直接取り付けられるストラップは 1 個までである。

また、それぞれのストラップには、取り付けたときに得られる**嬉しさ**が決まっている。この嬉しさは 1 つの整数で表される。JOI 君が嫌いなストラップもあり、その場合は嬉しさは負の値になる。

JOI 君は、携帯電話につながっているストラップの嬉しさの総和を最大化したい。ただし、必ずしもすべての端子にストラップが取り付けられている必要はなく、ストラップを 1 つも取り付けなくてもかまわない。

課題

JOI 君の持っている N 個のストラップの情報が与えられる。適切にストラップを取り付けたとき、携帯電話につながっているストラップの嬉しさの総和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データを読み込め。

- 1 行目には、整数 N が書かれている。 N はストラップの個数を表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 A_i, B_i が空白を区切りとして書かれている。これはストラップ i には端子が A_i 個あり、そのストラップを取り付けたときの嬉しさが B_i であることを表す。

出力

標準出力に、携帯電話につながっているストラップの嬉しさの総和の最大値を表す整数を 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。

- $1 \leq N \leq 2000$.



- $0 \leq A_i \leq N$ ($1 \leq i \leq N$).
- $-1\,000\,000 \leq B_i \leq 1\,000\,000$ ($1 \leq i \leq N$).

小課題

小課題 1 [5 点]

- $N \leq 15$ を満たす.

小課題 2 [5 点]

- $B_i \geq 0$ ($1 \leq i \leq N$) を満たす.

小課題 3 [45 点]

- $A_i \leq 15$ ($1 \leq i \leq N$) を満たす.

小課題 4 [45 点]

追加の制限はない.

入出力例

入力例 1	出力例 1
5	5
0 4	
2 -2	
1 -1	
0 1	
0 3	

この入力の場合、次のようにストラップを取り付けると嬉しさの総和は5となり、これが最大となる.

- ストラップ 2 を, 携帯電話に直接取り付ける.
- ストラップ 1 を, ストラップ 2 の端子に取り付ける.
- ストラップ 5 を, ストラップ 2 の端子に取り付ける.



入力例 2	出力例 2
6	0
2 -3	
3 -1	
0 -4	
0 -2	
1 -3	
4 -1	

この入力の場合、どのストラップの嬉しさも 0 未満である。よって、ストラップを 1 つも取り付けない場合に嬉しさの総和が最大となる。

入力例 3	出力例 3
15	43417
1 -4034	
1 3406	
0 6062	
4 -6824	
0 9798	
0 4500	
0 -1915	
1 2137	
0 9786	
0 7330	
0 -9365	
2 2730	
0 -5797	
0 6129	
0 8925	