



Links

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#)

Problem A

期末試験の成績

私は、中学校の教師である。ちょうど期末試験が終わったところで、すべての科目について全生徒の点数が手元にある。どれぐらい高い合計点を得た生徒がいるのか知りたいのだが、科目ごとの得点データになっているので、作業が容易でない。そこで、優秀なプログラマーであるあなたに手助けしてほしい。具体的には、合計点が最も高い生徒の合計点を求めるプログラムを書いてほしい。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

```

n m
p1,1 p1,2 ... p1,n
p2,1 p2,2 ... p2,n
...
pm,1 pm,2 ... pm,n

```

データセットの最初の行は、二つの整数 n と m からなる。 n は生徒の人数 ($1 \leq n \leq 1000$)、 m は科目数 ($1 \leq m \leq 50$) である。それに続く m 行のそれぞれには、特定の科目に対する n 人の生徒の得点がある。 $p_{j,k}$ は、生徒 k の科目 j に対する得点を表す整数である ($1 \leq j \leq m$, $1 \leq k \leq n$)。この値は、 $0 \leq p_{j,k} \leq 1000$ を満たす。

入力の終わりは二つのゼロからなる行で表される。データセットの個数は 100 を超えない。

Output

各データセットについて、合計点が最も高い生徒の合計点を出力せよ。生徒 k の合計点 s_k とは $s_k = p_{1,k} + \dots + p_{m,k}$ のことである。

Sample Input

```

5 2
10 20 30 40 50
15 25 35 45 55
6 3
10 20 30 15 25 35
21 34 11 52 20 18
31 15 42 10 21 19
4 2
0 0 0 0
0 0 0 0
0 0

```

Output for the Sample Input

```

105
83
0

```

(End of Problem A) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#)

Problem B

スクリーンキーボード

リモコンでスクリーンキーボードを操作して文字列を入力したい。リモコンには 4 方向の矢印と OK のボタンがついている (図 B-1)。与えられた文字列を与えられたスクリーンキーボードで入力するのに必要な最小ボタン押下回数を求めよ。

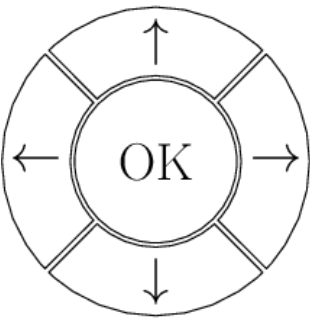


図 B-1 リモコン

A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	

図 B-2 スクリーンキーボードの例

入力する文字	強調表示の動き	ボタン操作																											
I		→,→,→,→,→,→,→,→,OK (9回)																											
C	<table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td></tr><tr><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td></tr><tr><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td></td></tr></table>	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		←,←,←,←,←,←,OK (7回)
A	B	C	D	E	F	G	H	I																					
J	K	L	M	N	O	P	Q	R																					
S	T	U	V	W	X	Y	Z																						
P	<table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td></tr><tr><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td></tr><tr><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td></td></tr></table>	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		↓,→,→,→,→,OK (6回)
A	B	C	D	E	F	G	H	I																					
J	K	L	M	N	O	P	Q	R																					
S	T	U	V	W	X	Y	Z																						
C	<table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td></tr><tr><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td></tr><tr><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td></td></tr></table>	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		↑,←,←,←,←,OK (6回)
A	B	C	D	E	F	G	H	I																					
J	K	L	M	N	O	P	Q	R																					
S	T	U	V	W	X	Y	Z																						

図 B-3 図 B-2 のスクリーンキーボードで “ICPC” を入力する最短の手順

スクリーンキーボードは格子状に並んだセルからなり、各セルには文字がひとつあるか空である。複数のセルが同じ文字を含むことはない。

スクリーンキーボードのセルのひとつは強調表示されていて、空でないセルが強調表示されているときにリモコンの OK ボタンを押すと、そのセルの文字が入力される。

最初はスクリーンキーボードの左上隅のセルが強調されている。矢印ボタンのひとつを押すと、強調セルはその矢印の示す方向の隣のセルに移る。強調セルがスクリーンキーボードの端にあるときには、スクリーンキーボードから出るような方向の矢印ボタンを押しても何も起きない。

例えば図 B-2 に示すスクリーンキーボード上で文字列 “ICPC” を入力するには、図 B-3 に示す手順でボタンを 28 回押せばよい。これが最小ボタン押下回数である。

スクリーンキーボードのセルに入っている文字は、英小文字 (‘a’, ‘b’, ..., ‘z’), 英大文字 (‘A’, ‘B’, ..., ‘Z’), 数字 (‘0’, ‘1’, ..., ‘9’), コンマ (‘,’), ハイフン (‘-’), ピリオド (‘.’), スラッシュ (‘/’), コロン (‘:’), セミコロン (‘;’), アットマーク (‘@’) のいずれかである。

Input

入力は次の形式の最大 100 個のデータセットからなる。

hw
r1
...
rh
s

データセットの最初の行のふたつの整数 h と w は、それぞれスクリーンキーボードの高さと幅である。これらは空白で区切られており、 $1 \leq h \leq 50$ と $1 \leq w \leq 50$ を満たす。

続く h 行に、スクリーンキーボードの各行が与えられる。 i 番目の行 r_i はスクリーンキーボードの i 行目を表している。 r_i は長さ w の文字列であり、スクリーンキーボードの i 行目の文字が左から順に並んだものである。ただし、スクリーンキーボード上の空のセルは、アンダースコア (‘_’) で表現されている。

スクリーンキーボードは上述の条件を満たす。

続く行の s は、長さ 1 以上 1000 以下の、入力したい文字列である。 s に含まれる文字は、必ず与えられたスクリーンキーボード上に存在する。 s はアンダースコアを含めないことに注意。

入力の終わりはゼロふたつからなる行で表される。

Output

各データセットについて、与えられた文字列を与えられたスクリーンキーボードで入力するのに必要な最小ボタン押下回数を表す整数ひとつをもつ行を出力せよ。

Sample Input

```
3 9
ABCDEFGHI
JKLMNOPQR
STUVWXYZ_
ICPC
5 11

_ A
_ M
_
_ C
_
ACM
4 21
1_2_3_4_5_6_7_8_9_0_-
QqWwEeRrTtYyUuIiOoPp@
AaSsDdFfGgHhJjKkLl;:_:
ZzXxCcVvBbNnMm,_. /_
ICPC2019,AsiaYokohamaRegional,QualificationRound
0 0
```

Output for the Sample Input

```
28
23
493
```

(End of Problem B) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#)

Problem C

天秤

実験化学者のあなたは粉末薬品の計量用に天秤ひとつと分銅のセットを持っている。

作業効率上、1 回の計量には天秤を 1 回使うだけにしたい。セット中の分銅は同時にいくつ使ってもよく、天秤の薬品と反対側に置いて同じ側に置いてもいい。たとえば 2 単位と 9 単位の分銅があれば、2 単位、9 単位だけでなく、薬品と反対側に両方の分銅を置けば 11 単位 (図 C-1 左)、片方を薬品と同じ側に置けば 7 単位の薬品を量り取ることができる (図 C-1 右)。効率的に量り取れる量はこれらだけである。

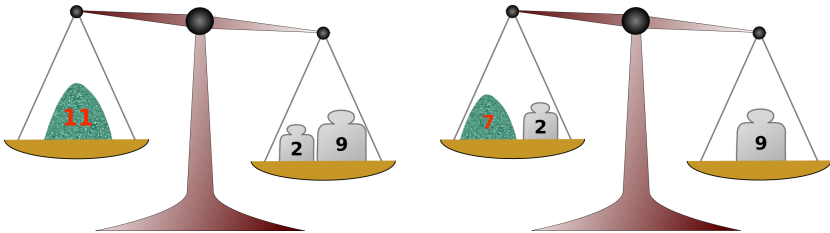


図 C-1 薬品 11 単位と 7 単位の計測法

手元に今日計量する薬品量のリストがある。しかし手持ちの分銅セットだけでは、計量リスト中の量すべてを効率的に量れるとは限らない。その場合はもうひとつだけ分銅を買ってきて補ってもよいのだが、重い分銅ほど高価なので、なるべく軽いもので済ませたい。

なお, どんな正の重さの分銅でも売っているが, 負の重さのものはない.

Input

入力は次の形式の最大 100 個のデータセットからなる.

$n\ m$
 $a_1\ a_2\ \dots\ a_n$
 $w_1\ w_2\ \dots\ w_m$

データセットの最初の行の n と m は, それぞれ計量リスト中の薬品量の数とセット中の分銅の個数である. これらは空白で区切られており, $1 \leq n \leq 100$ と $1 \leq m \leq 10$ を満たす整数である.

次の行には計量リスト中の n 種の量 a_1 から a_n が空白区切りで並ぶ. 各 a_i は $1 \leq a_i \leq 10^9$ を満たす整数で, $i \neq j$ ならば $a_i \neq a_j$ である.

データセットの最後の 3 行目には, 手持ちの m 個の分銅の重さ w_1 から w_m が空白区切りで並ぶ. 各 w_j は $1 \leq w_j \leq 10^8$ を満たす整数である. 分銅は同じ重さのものがいくつもあるかもしれない.

入力の終わりはゼロふたつからなる行で表される.

Output

各データセットについて以下に指示する整数ひとつからなる 1 行を出力せよ.

- 分銅を追加せずに計量リスト中のすべての量が量り取れるなら 0.
- 分銅ひとつの追加で計量リスト中のすべての量が量り取れるようになるなら, そのような分銅のうちもっとも軽いものの重さ. 追加する分銅は 10^8 より重くてもよい.
- 分銅ひとつの追加では計量リスト中のすべての量を量り取れるようにならないなら, -1.

Sample Input

4 2
9 2 7 11
2 9
6 2
7 3 6 12 16 9
2 9
5 2
7 3 6 12 17
2 9
7 5
15 21 33 48 51 75 111
36 54 57 93 113
0 0

Output for the Sample Input

0
5
-1
5

(End of Problem C) [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)

Problem D

計数カウンタ

計数カウンタが何個か並んでいる. カウンタのボタンを押すと表示されている値が一つ増える. ただし, 値が既に最大値なら, 値は 1 に戻る. カウンタは全て同じモデルで最大値も同じである.

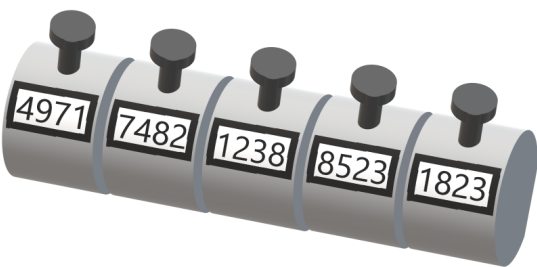


図 D-1 計数カウンタ

各カウンタに表示されている値から始めて、各々について指定される目的の値に変えたい。しかし多数のカウンタのボタンを一つ一つ押していくのは面倒なので、特殊な道具を作った。これを使うと、1回の操作で隣接する一連のカウンタのボタンを1回ずつ押せるのだ。対象とするカウンタの範囲は、操作1回ごとにどこからどこまででも自由に決められる。

各カウンタに表示されている値を目的の値に変えるのに最低何回の操作が必要だろうか。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

n m
 a_1 a_2 ... a_n
 b_1 b_2 ... b_n

各データセットは3行からなる。1行目には、カウンタの個数を表す n ($1 \leq n \leq 1000$) と、カウンタに表示される最大値を表す m ($1 \leq m \leq 10000$) が与えられる。2行目には、各カウンタの最初の値を表す a_i ($1 \leq a_i \leq m$) が空白区切りで与えられる。3行目には、各カウンタの目的の値を表す b_i ($1 \leq b_i \leq m$) が空白区切りで与えられる。

入力の終わりは、二つのゼロからなる行で表される。データセットの個数は100を超えない。

Output

各データセットについて、全てのカウンタにそれぞれ目的の値を表示させるのに必要な最小の操作回数を1行に出力せよ。

Sample Input

```
4 5
2 3 1 4
2 5 5 2
3 100
1 10 100
1 10 100
5 10000
4971 7482 1238 8523 1823
3287 9013 9812 297 1809
0 0
```

Output for the Sample Input

```
4
0
14731
```

(End of Problem D) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#)

Problem E

立方体表面パズル

立方体表面パズルとは、6つのピースを適切に組み合わせて表面に欠けのない中空の立方体を作るパズルである。パズルの各ピースは、単位小立方体を平面上に格子状に並べてできている。1辺の長さ n の立方体を作るピースは以下の2領域からなる。

- 中央領域 (青): 1辺の長さ $n-2$ の正方形領域。単位立方体で埋め尽くされている。
- 周辺領域 (赤): 中央部分の外縁の幅1の領域。この領域の単位正方形の上には何もなければ、単位立方体があるかである。

各ピースの単位立方体は面がつながっている。各ピースは自由に回転して使ってよく、表裏のどちらが組み立てた立方体の内側・外側になってもよい。各ピースの中央領域上の単位立方体は組み立てた立方体の各面の中央に来なくてはならない。

たとえば、図 E-1 の 6 つのピース (Sample Input の最初のデータセットである) が与えられたとする。このとき、図 E-2 のように組み立てることで立方体を作れる。

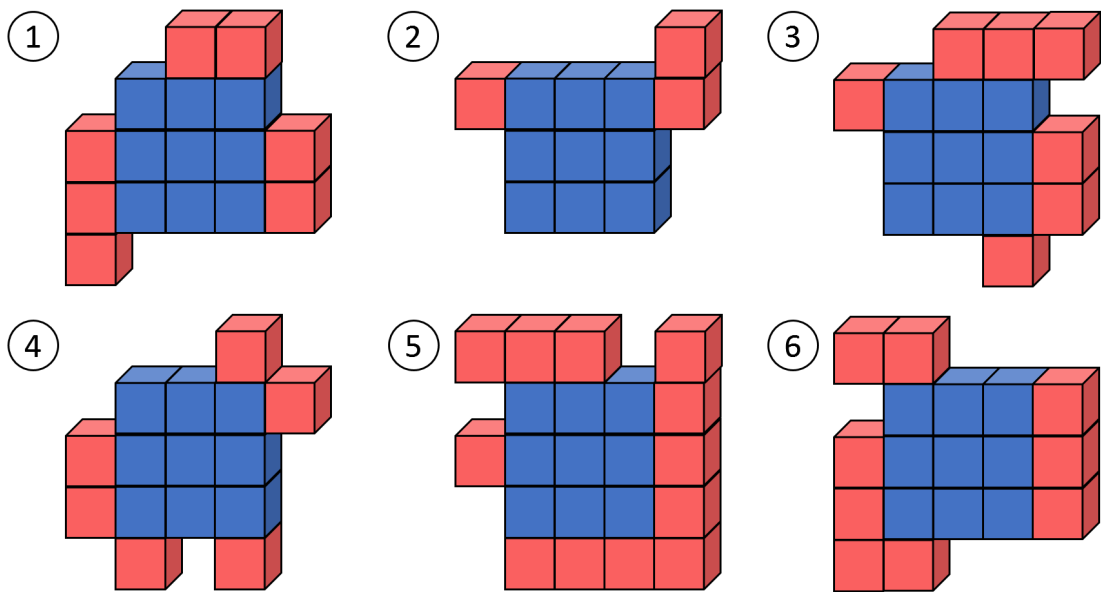


図 E-1 Sample Input の最初のデータセットのピース

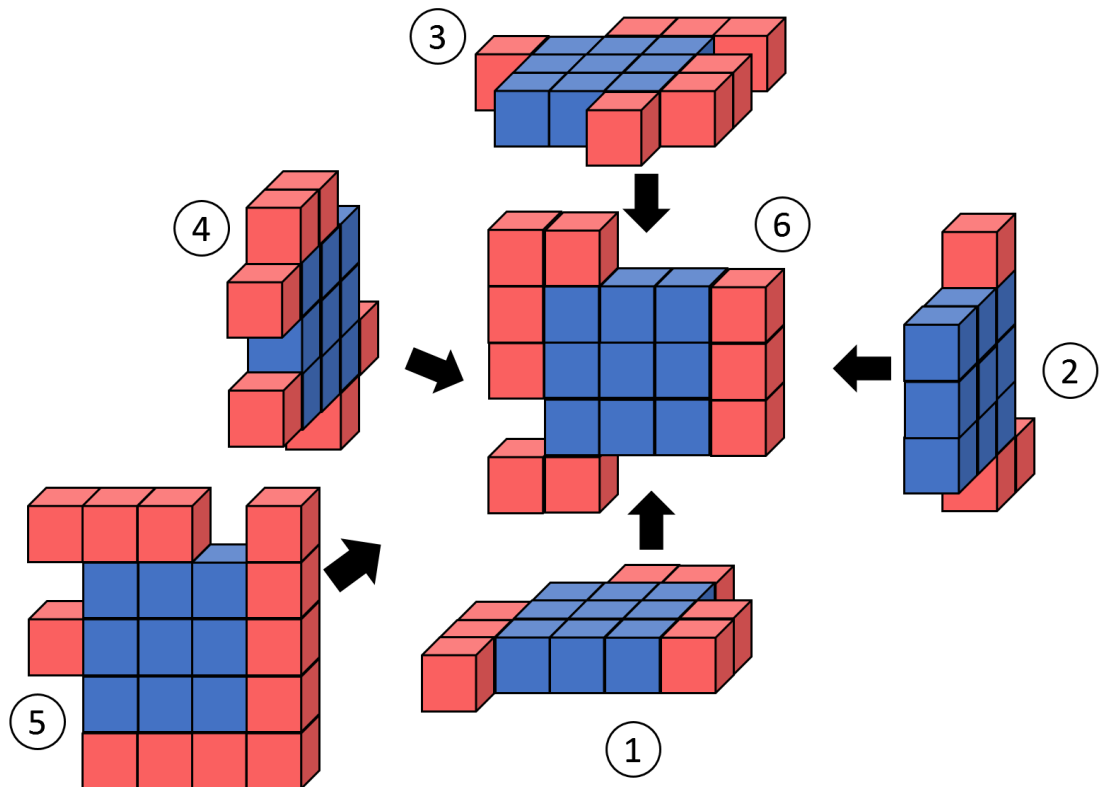


図 E-2 立方体の組み立て

Hadrian Hex 氏はたくさんの立方体表面パズルを収集していた。あるとき、それらのピースが混ざってしまい、どのピースを組み合わせれば立方体を作れるか分からなくなってしまった。そこで、与えられたピースの組合せによって立方体ができるかを判定するプログラムを作って Hex 氏を助けて欲しい。

Input

入力は最大 200 データセットからなり、各データセットは次の形式で表される。

```
n
x1,1x1,2 ... x1,n
x2,1x2,2 ... x2,n
```

...

$$x_{6n,1}x_{6n,2} \cdots x_{6n,n}$$

各データセットの1行目には作る立方体の1辺の長さ n ($3 \leq n \leq 9$, n は奇数) が与えられる。続く $6n$ 行に、6つのピースが与えられる。各ピースは n 行で表される。各行は格子の1行に対応し、その行の各文字 ('X' または '.') は対応する単位正方形上の単位立方体の有無を意味する。'X' なら単位立方体があり、'.' なら空である。

各ピースの中央領域は各ピースのデータの中央に位置する。

入力の終わりは、ゼロだけからなる行で表される。

Output

各データセットについて、立方体ができる場合には “Yes”，できない場合には “No” を出力せよ。

Sample Input

```
5
..XX.
.XXX.
XXXXX
XXXXX
X....
....X
XXXXX
.XXX.
.XXX.
.....
..XXX
XXXX.
.XXXX
.XXXX
...X.
...X.
.XXXX
XXXX.
XXXX.
.X.X.
XXX.X
.XXXX
XXXXX
.XXXX
.XXXX
XX...
.XXXX
XXXXX
XXXXX
XXXXX
XX...
5
..XX.
.XXX.
XXXXX
XXXX.
X....
....X
XXXXX
.XXX.
.XXX.
.....
.XXXX
XXXX.
.XXXX
.XXXX
...X.
...X.
.XXXX
XXXX.
XXXX.
.X.X.
XXX.X
.XXXX
XXXXX
.XXXX
.XXXX
XX...
XXXXX
```

XXXXX
.XXXX
XX...
0

Output for the Sample Input

Yes
No

(End of Problem E) [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)

Problem F

色の切り替え

無向完全グラフが与えられる。どの頂点どうしもそれぞれ1本の赤か黒の辺で結ばれている。各辺にはペナルティと呼ばれる整数値が設定されている。

このグラフに対してある操作を繰り返すことによって、赤い辺だけからなる「全域木」を作りたい。すなわち、ちょうど（頂点数 − 1）本の辺だけが赤く、その赤い辺だけで任意の頂点間が直接または間接的につながっているようにしたい。そういう木がいくつも作れるのなら、赤い辺のペナルティの合計が最小のものを選びたい。

1回の操作では、頂点をひとつ選び、それに繋がっている全ての辺の色を反転、つまり赤い辺は黒く、黒い辺は赤くする。

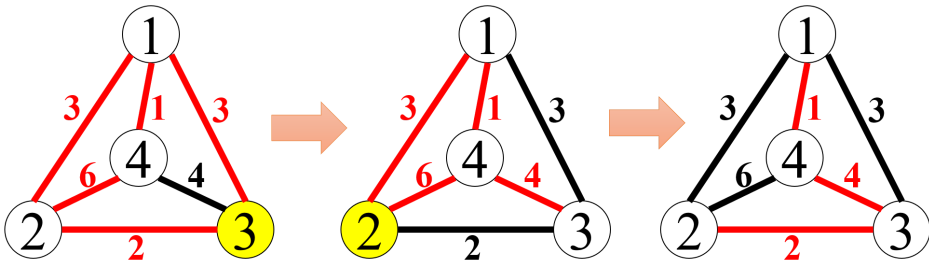


図 F-1 サンプル入力の最初のデータセットとその解

例として、図 F-1 の左端のグラフはサンプル入力の最初のデータセットを図示したものである。頂点 3 に繋がる全ての辺の色を反転し、次に頂点 2 に繋がる辺の色を反転することで、図の右端に示すように赤の辺からなる全域木を構成することができる。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

n
 $e_{1,2} \ e_{1,3} \ \dots \ e_{1,n-1} \ e_{1,n}$
 $e_{2,3} \ e_{2,4} \ \dots \ e_{2,n}$
 \dots
 $e_{n-1,n}$

整数 n ($2 \leq n \leq 300$) は頂点数を表す。頂点には 1 から n までの番号がついている。整数 $e_{i,k}$ ($1 \leq |e_{i,k}| \leq 10^5$) は、頂点 i と頂点 k を結ぶ辺のペナルティと初期状態での色を表す。絶対値 $|e_{i,k}|$ は辺のペナルティを表す。 $e_{i,k} > 0$ のとき、頂点 i と頂点 k を結ぶ辺の色が初期状態で赤であることを、 $e_{i,k} < 0$ のとき黒であることを示す。

入力の終わりは、ゼロだけからなる行で表される。データセットの個数は 50 を超えない。

Output

各データセットに対し、上述の操作で作れる赤い全域木のうちペナルティの合計が最小のものの値を出力せよ。上述の操作では赤い全域木を作ることができないならば −1 を出力せよ。

Sample Input

4
3 3 1


```
2 6
-4
3
1 -10
100
5
-2 -2 -2 -2
-1 -1 -1
-1 -1
1
4
-4 7 6
2 3
-1
0
```

Output for the Sample Input

```
7
11
-1
9
```

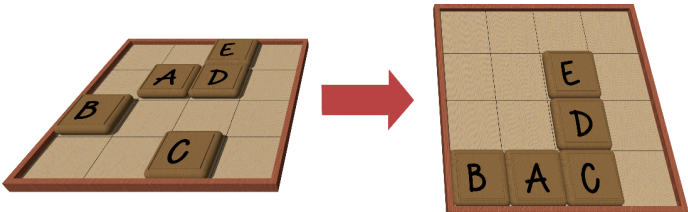
(End of Problem F) [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)

Problem G

タイルを動かそう！

正方形のマスを描いた枠付きの正方形の盤がある．各マスは空か、英字を刻印したちょうどマスにはまるタイルがあるかのどちらかである．盤を向こう側、手前側、左側、右側の4方向のいずれかに傾けると、すべてのタイルは盤の枠に押し詰められるまで上下左右に滑っていく．

下図に盤を手前に傾ける前後のタイルの配置の例を示す．



文字 U, D, L, R のそれぞれで、盤を向こう側、手前側、左側、右側に傾ける操作を表すことにしよう．さらに、これらの文字の列で操作の列を表すことにしよう．たとえば文字列 DRU は、盤をまず手前、次に右、最後に向こう側に傾ける一連の操作を表す．タイルは最初に下に、次いで右に、最後に上に動くことになる．

非常に長い操作列を扱うために操作列の繰り返しの表記を使うことにする．ある空ではない操作列 seq を k 回繰り返すことを $(seq) k$ と記す．ここで k は整数である．例えば $(LR) 3$ は $LRLRLR$ と同じ操作列を表す．繰り返し表記はネストしていてもよい．例えば $((URD) 3L) 2R$ は $URDURDURDLURDURDURDLR$ を意味する．

タイルの初期位置と操作列が与えられたとき、その操作列に従って操作したあとのタイルの位置を計算するプログラムを書いてほしい．

Input

入力は 100 個以下のデータセットからなる．各データセットは次の形式で表される．

```
n
s11 ... s1n
...
sn1 ... snn
seq
```

最初の行の n は盤面の 1 行 (および 1 列) に並ぶマス目の数である ($2 \leq n \leq 50$)．次の n 行は盤上の各マス目の初期状態を表す． s_{ij} は盤面の上から i 番目、左から j 番目のマス目 (どちらも 1 から数える) の初期状態を表す文字で、何も置いていないことを表す $'.'$ (ピリオド) であるか、タイルがあることとそこに刻印した文字を表す英大文字 $'A' \cdots 'Z'$ のいずれかである．

入力の終わりは、ゼロだけからなる行で表される.

各データセットについて、初期状態から始めて指定された操作列に従って操作したあとの各マスの状態を出力せよ。盤面の状態は入力で盤上のマスの初期状態を与えたのと同じフォーマットで n 行に出力せよ。

[illegible]

```

. . . .
. . E .
. . D .
BAC .

```

```
....
...E
...D
.BAC
....
..C
.AB
....P
PPPPP
..PPP
PPPPP
....P
.....
.....
.....
.....
.....
XXXX.....
PXXXX.....
CXXXX.....
XXXX.....
XXPCXX.....
CCXXCX.....
CXXXXX.....
CXXXXX.....
CPCIXXX.....
CXPCCXX.....
PCXXXIC.....
CPPCCXI.....
CXCIPXXX.....
XCPCICIXX.....
PIPPICIXII.....
.....
.....
N.....
JNG...
RMMKFE
BEAIFN
```

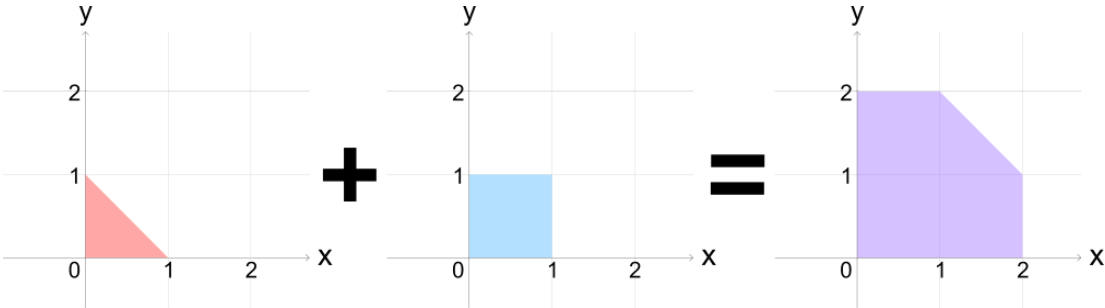
(End of Problem G) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#)

Problem H

凸多角形の加法

凸郎君は凸多角形が大好きである。凸郎君は凸多角形を調べているうちに、以下の素敵な凸多角形の加法を思いついた。

xy 平面上の2点 (x_1, y_1) と (x_2, y_2) の和を $(x_1 + x_2, y_1 + y_2)$ と定義する。多角形を、その周上と内部の点すべての集合として扱う。このとき2つの多角形 S_1 と S_2 の和 $S_1 + S_2$ を $v_1 \in S_1$ および $v_2 \in S_2$ を満たす2点の和である点 $v_1 + v_2$ すべての集合と定義する。こう定義すると、2つの凸多角形の和が必ず凸多角形になるのだ! 非負の整数と多角形の積を、 $0S = \{(0, 0)\}$ および正整数 k に対し $kS = (k-1)S + S$ と帰納的に定める。この問題では線分や一点も凸多角形であるとする。



凸郎君はこの加法の性質を調べるため、2つの凸多角形 P と Q を基にたくさんの凸多角形を作ってみたが、ある日誤って元の多角形と計算過程を書いた紙を捨ててしまった。手元に残ったのは P, Q の非負整数係数の線形結合:

$$\begin{aligned} R &= aP + bQ, \\ S &= cP + dQ, \end{aligned}$$

で表される2つの凸多角形 R, S だけである。幸いにも凸郎君は、 P と Q のすべての頂点の座標が整数であることと、係数 a, b, c, d が $ad - bc = 1$ を満たしていることは覚えていた。

凸郎君は、プログラマーである友人のあなたに R と S から P と Q を復元するプログラムを依頼した。すなわち、凸多角形 R, S が与えられるので、非負の整数 $a, b, c, d (ad - bc = 1)$ と頂点が整数座標の点であるような凸多角形 P, Q で上の方程式を満たすものを求めてほしいというのである。この方程式は複数の解を持つかもしれない。方程式を満たす P と Q の面積の和の最小値を求めるプログラムを作成せよ。

Input

入力は高々 40 個のデータセットからなる。各データセットは次の形式で表される。

```
n m
x1 y1
...
xn yn
x'1 y'1
...
x'm y'm
```

n は凸多角形 R の頂点の個数を表し、 m は凸多角形 S の頂点の個数を表す。 n と m は整数であり、 $3 \leq n \leq 1000$ および $3 \leq m \leq 1000$ が成り立つ。 (x_i, y_i) は凸多角形 R の i 番目の頂点の座標を表し、 (x'_i, y'_i) は凸多角形 S の i 番目の頂点の座標を表す。 x_i, y_i, x'_i, y'_i は、それぞれ整数であり -10^6 以上、 10^6 以下である。凸多角形 R, S の頂点はそれぞれ反時計回りの順番で与えられる。また、1 つの凸多角形の異なる 3 つの頂点は一直線上に並んでいない。

入力の終わりは、2 つのゼロからなる行で表される。

Output

各データセットについて、それぞれ 1 行に P と Q の面積の和の最小値を 2 倍した整数を出力せよ。 P と Q の頂点の座標は整数なので、面積の 2 倍は常に整数となることに注意せよ。

Sample Input

```
5 3
0 0
2 0
2 1
1 2
0 2
0 0
1 0
0 1
4 4
0 0
5 0
5 5
0 5
0 0
2 0
2 2
0 2
3 3
0 0
1 0
0 1
0 1
0 0
1 1
3 3
0 0
1 0
1 1
0 0
1 0
1 1
4 4
0 0
2 0
2 2
0 2
0 0
1 0
```

1 2
0 2
4 4
0 0
3 0
3 1
0 1
0 0
1 0
1 1
0 1
0 0

Output for the Sample Input

3
2
2
1
0
2

(End of Problem H) [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)