

問題

(※審査員特別賞対象問題です)

まさお君は最近カードを使ったゲームにはまっています。このゲームで使うカードには、1枚にひとつの数字が書かれています。数字は1から13のいずれかで、それぞれ4枚ずつ計52枚からなっています。1回のゲームで、各プレイヤーに5枚のカードが配られます。配られたカードの組み合わせによって役が決まります。役には次の5種類があります。

役の名称	役の番号	カードの役	例
ワンペア	1	同じ数字のカードが2枚ある	[5] [5] [3] [10] [11]
ツーペア	2	ワンペアが2組ある	[5] [5] [3] [3] [11]
スリーカード	3	同じ数字のカードが3枚ある	[5] [5] [5] [10] [11]
フォーカード	4	同じ数字のカードが4枚ある	[5] [5] [5] [5] [11]
フルハウス	5	ワンペアとスリーカードがある	[5] [5] [5] [10] [10]

ゲームの数 n 及び各ゲームで配られた5枚のカードを入力とし、各ゲームの役の番号の中で最大のものを出力するプログラムを作成してください。ただし n は1以上10000以下とします。なお、ひとつのゲームで配られたカードが複数の役に該当する場合は、より大きな役の番号をそのゲームの役としてください。すべてのゲームでどの役にも該当しない場合は0を出力してください。

入 力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロひとつの行で示されます。各データセットは以下のとおりです。

- 1 行目 ゲームの数 n (整数)
- 2 行目 第 1 のゲームで配られたカード (整数 整数 整数 整数 整数 ; 半角空白区切り)
- 3 行目 第 2 のゲームで配られたカード
- ⋮
- $n+1$ 行目 第 n のゲームで配られたカード

出 力

入力データセットごとに、最大の役の番号または 0 を出力します。

入力例

```
3
5 11 3 5 10
5 5 5 10 10
12 2 8 7 10
2
3 4 10 2 3
1 2 3 4 5
4
8 8 3 2 4
8 9 9 11 11
10 2 10 10 10
13 13 13 13 2
1
2 2 9 3 3
2
9 8 7 6 5
8 7 6 7 7
3
7 8 3 4 10
2 4 12 11 13
12 13 1 2 7
0
```

出力例

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

問題

会津若松市の鶴賀小学校では、毎年クラス対抗でフットサル大会を行うのが恒例行事となっています。大会は、全クラスが総当たりで一回ずつ対戦するリーグ戦で行われます。しかし、近年、クラス数が増えたため順位の計算が大変です。そこであなたは、得点情報を集計し順位を計算するプログラムを作成することになりました。

得点情報は、対戦相手から奪った得点で構成されます。例えば、参加チーム数が5の場合に与えられる得点情報は図1のようになります。

	チーム1	チーム2	チーム3	チーム4	チーム5
チーム1		1	4	3	3
チーム2	1		2	5	2
チーム3	2	1		1	1
チーム4	0	1	2		4
チーム5	1	0	1	3	

図1 チーム数5の場合の得点情報

順位	チーム	勝利点	総得点
1位	チーム1	10	11
2位	チーム2	10	10
3位	チーム4	6	7
4位	チーム3	1	5
5位	チーム5	1	5

図2 順位の例

図1の表の一行は、各チームの得点情報の例です。例えば、チーム1の行は、チーム1がチーム2、3、4、5からそれぞれ1点、4点、3点、3点を得点したことを意味します。また、図1からチーム1とチーム3の対戦は、4対2でチーム1が勝ったことを意味します。

このような得点情報から、以下の方法に従って順位を決定します。

- それぞれの対戦において、得点が多いチームが「勝ち」、少ないチームが「負け」、同じ場合は「引き分け」とする。
- 試合ごとに、各チームの「勝利点」に以下の値が加算される。
 - ▶ 勝ったチームは3
 - ▶ 引き分けの場合、両チームに1
 - ▶ 負けたチームは0
- 最終的に勝利点の高い順に順位が定められる。
- 勝利点が高点の場合、総得点が多いチームを上位とする。
- 勝利点も総得点も同じ場合、チーム番号の小さいチームを上位とする。

図1の得点情報の場合、上記のルールに従うと順位は図2のようになります。

参加チーム数 n と、参加各チームが得た得点情報を入力とし、チームの順位が高い順にチーム番号を出力するプログラムを作成してください。ただし、 n は2以上100以下の整数、得点情報は0以上10以下の $n-1$ 個の整数で表されます。なお、各チームは1から n までの数で表されます。得点情報は、対戦相手のチーム番号の小さい順に従って入力されます。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロひとつの行で示されます。各データセットは以下のとおりです。

1 行目 チーム数 n (整数)

2 行目 チーム 1 の得点情報 $g_1 \cdots g_{n-1}$ (全て整数: 半角空白区切り)

:

$n+1$ 行目 チーム n の得点情報

出力

入力データセットごとに、順位の高い順にチーム番号を出力します。

入力例

```
5
1 4 3 3
1 2 5 2
2 1 1 1
0 1 2 4
1 0 1 3
3
1 1
1 1
1 1
3
9 5
0 0
5 7
0
```

出力例

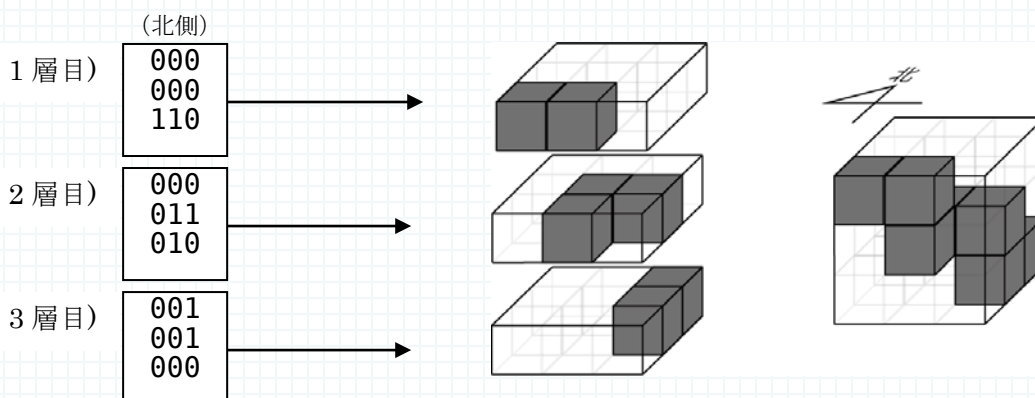
```
1
2
4
3
5
1
2
3
1
3
2
```

問題

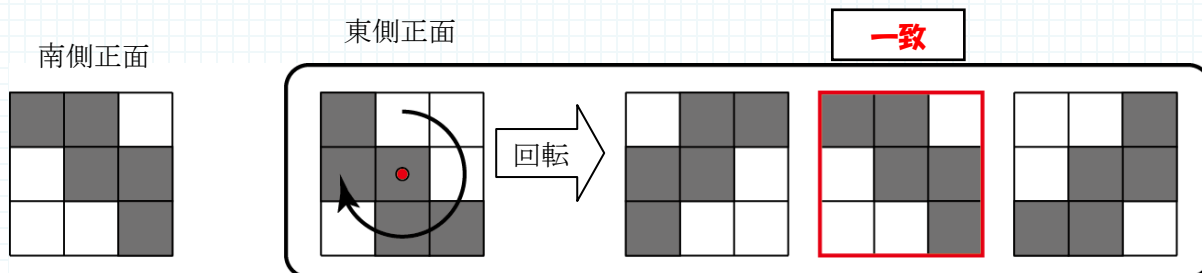
会津芸術館では、立方体ブロックを積み上げて出来た特殊な立方体の装置が人気です。この装置の各ブロックは点灯命令によって透明になったり、黒くなったりして、オブジェを浮かび上がらせることができます。

会津芸術館は創立記念を飾るため、「南から見ても東から見ても同じ絵柄」をテーマとしたオブジェを募集しました。しかし、応募された作品を点灯命令に変換した段階で、テーマに反する絵柄になるものが混ざっていることに気づきました。このような作品を区別するため、点灯命令が「南から見ても東から見ても同じ絵柄」となるかどうかを判断するプログラムを作成してください。

装置は同じ大きさの立方体ブロックが $n \times n \times n$ 個積み重なって出来ています。各ブロックは点灯命令によって透明か、黒かのいずれかの状態になります。点灯命令は装置を構成するブロックの層ごとに与えられます。層ごとの点灯命令は、 $n \times n$ の表として与えられます。この表は、1行目が最も北側のブロックの列、 n 行目が南端のブロック列に対応するように並んでいます。各行は、西側のブロックに対する命令から、透明のまま“0”（ゼロ）か黒くする“1”か、のどちらかで与えられます。下図は、3層分の表が与えられたときのオブジェの例です。



また、南側正面と東側正面から見えるオブジェの絵柄が一致している場合、「南から見ても東から見ても同じ絵柄」となります。なお、絵柄の中心を軸に回転させて一致する場合でも同じ絵柄であるとしします。例えば、上図の例で与えられたオブジェの南からみた絵柄と東から見た絵柄は、下図から一致することが分かります。



装置の大きさ n と点灯命令 r を入力とし、オブジェが「南から見ても東から見ても同じ絵柄」となるかどうかを判定して、同じ絵柄になる場合は yes、そうでない場合は no を出力するプログラムを作成してください。点灯命令 r は n 層分の $n \times n$ の表（要素は 0 か 1）から成り立っています。ただし、 n は 2 以上 100 以下の整数とします。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロひとつの行で示されます。各データセットは以下のとおりです。

1 行目 装置の大きさ n (整数)
2 行目 1 層 1 行目の点灯命令 r_{111} r_{211} \dots r_{n11} (すべて半角数字)
3 行目 1 層 2 行目の点灯命令
:
 $n+1$ 行目 1 層 n 行目の点灯命令
 $n+2$ 行目 2 層 1 行目の点灯命令
:
 n^2+1 行目 n 層 n 行目の点灯命令

出力

入力データセットごとに、**yes** または **no** と出力します。

入力例

```
3
000
000
110
000
011
010
001
001
000
3
000
001
000
000
000
000
000
000
000
000
000
2
10
11
10
00
0
```

出力例

```
yes
no
yes
```

問題

人里離れた森の奥深くに、マリーという名の魔女が住んでいました。彼女は魔女なので、食料・水・燃料など、生活に必要なものを魔法によってまかっています。

彼女は石で作った魔法陣を設置することによって魔法をかけます。魔法陣は、石を配置し、その内のいくつかの石同士を紐で結びつけて作ります。そのようにして作られた魔法陣は、不思議なことに、石の位置や紐の長さにかかわらず同じ効果を発揮するのです。ただし、魔法陣を設置するには以下の制約を守る必要があります。

- 同じ二つの石を直接複数の紐で結んではいけません。
- 一つの紐の両端を同じ石に結んではいけません。
- 紐は、その両端を除いて、他の紐や石に触れてはいけません。
- 同じ場所に石を重ねてはいけません。

ある日マリーは、玄関に足元灯を作るために魔法陣を設置することにしました。しかし、魔法陣を設置できるスペースは一本の直線上しかありませんでした。そこで計算の苦手なマリーに代わって、マリーが作った魔法陣がこの玄関に設置できるものかどうか調べてあげましょう。

彼女が作成した魔法陣の情報を入力とし、それが直線の上に設置できるかどうかを判定し、設置できる場合は **yes** と、そうでない場合は **no** と出力するプログラムを作成して下さい。ただし、石は点として扱い、紐は線分として扱います。魔法陣の情報は、石の個数 n 、紐の本数 m 、 m 個の紐の情報からなりたっています。紐の情報は、その両端の石の番号 u 、 v で表されます。石には 1 から n まで番号が与えられ、 n は 1 以上 100000 以下、 m は 0 以上 1000000 以下となります。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロふたつの行で示されます。各データセットは以下のとおりです。

1 行目 石の個数 n 紐の本数 m (整数 整数 ; 半角空白区切り)

2 行目 第 1 の紐の情報 u v (整数 整数 ; 半角空白区切り)

3 行目 第 2 の紐の情報

:

$m+1$ 行目 第 m の紐の情報

出力

入力データセットごとに、yes または no と出力します。

入力例

```
5 4  
1 2  
2 3  
3 4  
4 5  
4 6  
1 2  
1 3  
1 4  
2 3  
2 4  
3 4  
5 0  
0 0
```

出力例

```
yes  
no  
yes
```

問題

数学とプログラミングが好きな A 君はロボットを作ることを目指して猛勉強中です。彼はロボットアームの関節制御に興味を持ち仕組みを調べました。以下は A 君の勉強メモです。

メモ1 ロボットアームのモデル (図1)

- ・ ロボットアームは、座標系の原点に固定される肩関節、それを中心に回転する上腕、上腕の先端にあるひじ関節、ひじ関節を中心に回転する下腕、および下腕の先端にある手のひらから構成される。
- ・ 初期状態のアームは、x軸方向と重なる。

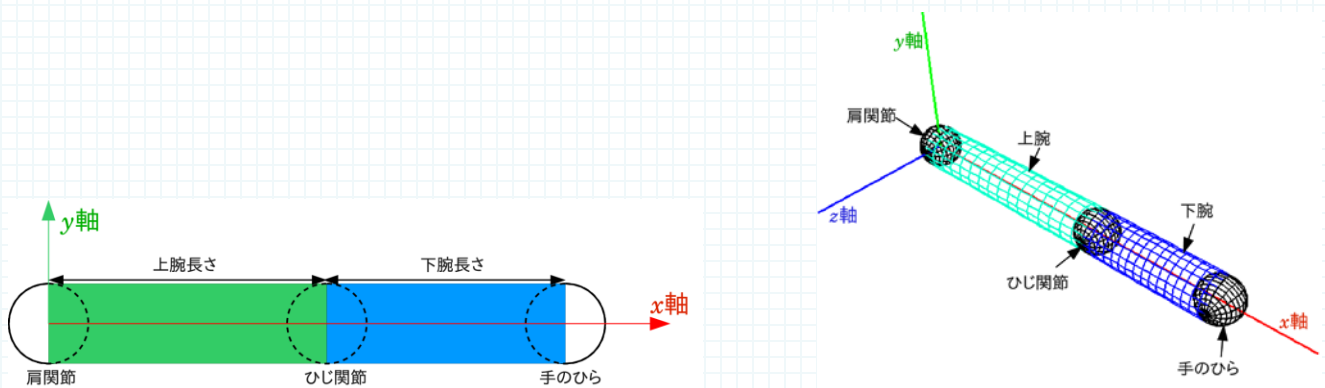


図1 ロボットアーム初期状態の2次元表示 (左) と 3次元表示 (右)

メモ2 ロボットアームを動かす

- ・ アームは一連の「命令セット」に従い、状態 (ひじ関節と手のひらの位置) が次々と変化。
- ・ 命令セットは上腕と下腕に別々に与える「回転情報」からなる。回転情報は回転角 θ と回転軸方向ベクトル \mathbf{v} からなる。
- ・ 先に、上腕回転情報の θ と \mathbf{v} による回転。肩関節から \mathbf{v} の方向へ伸びる仮定の直線が回転軸で、その軸の方向から見て反時計回りになるように、 θ だけ回転。図2の2は図2の1の状態のアームが上腕回転情報

報「 $\theta = -90$ 度、 $\mathbf{v} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$ (Y軸方向)」を実行した結果。当然、上腕が動くと下腕も連動。

- ・ 次に、下腕回転情報の θ と \mathbf{v} による回転。回転軸がひじ関節から伸びる直線となる以外は、上腕回転命令の実行と同じ。なお、下腕回転情報は上腕の回転によっても変わらないことに注意。図2の3は、図

2の2の状態のアームが下腕回転情報「 $\theta = -90$ 度、 $\mathbf{v} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$ (X軸方向)」を実行した結果。

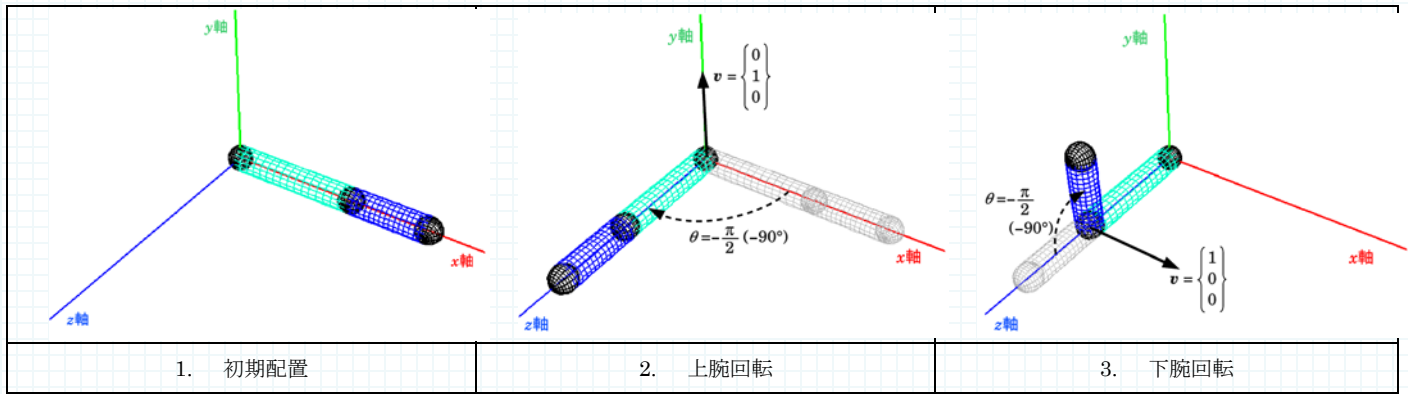


図 2 腕回転の例

メモ 3 回転後の位置の計算

- 回転後のひじ関節と手のひらの位置は、現在位置と与えられる回転情報から計算可能。
- 回転軸が座標系の原点から伸びる場合の、現在位置 r 、回転情報「 θ (度)、 $v = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix}$ 」から、回転後位置 p

を求める手順：

(1) 値 x, y, z, w を以下の式から計算

$$x = \frac{a}{l} \sin \alpha, \quad y = \frac{b}{l} \sin \alpha, \quad z = \frac{c}{l} \sin \alpha, \quad w = \cos \alpha$$

$$\text{但し } l = \sqrt{a^2 + b^2 + c^2}, \quad \alpha = \frac{\theta}{360} \times \pi$$

(2) x, y, z, w で、回転行列 $[R]$ を構成

$$[R] = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(zx + yw) \\ 2(xy + zw) & 1 - 2(z^2 + x^2) & 2(yz - xw) \\ 2(zx - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix}$$

(3) p を計算

$$p = [R]r$$

- 上腕の回転は、回転軸が座標系原点から伸びる直線となるので、上の手順をそのまま使えば回転後のひじ関節と手のひらの位置を計算できる。
- 下腕の回転は、回転軸が座標系の原点を通らないため工夫が必要…

A君は上記メモに基づき、ロボットアームが一連の命令セットを実行した後のひじ関節と手のひらの位置を計算するプログラムを書こうとしましたが、下腕の処理は未完成です。そしてあなたの力を借りることになりました。

ロボットアームの上腕の長さ up と下腕の長さ lo と命令セットの数 n 、それに続く n セットの命令を入力とし、初期状態のロボットアームにこれらの命令を順次に与えた後の、ひじ関節と手のひらの位置座標を出力するプログラムを作成してください。命令セットは、上腕回転情報（回転角度 θ_{up} と上腕の回転軸方向ベクトル

$\begin{Bmatrix} a_{up} \\ b_{up} \\ c_{up} \end{Bmatrix}$ ）と下腕回転情報（回転角度 θ_{lo} と下腕の回転軸方向ベクトル $\begin{Bmatrix} a_{lo} \\ b_{lo} \\ c_{lo} \end{Bmatrix}$ ）からなります。また、腕と関節の

半径を 0 とし、上腕・下腕・関節同士の干渉は考えなくても良いものとします。なお、 up と lo の範囲は 1 以上 1000 以下の整数であり、 n の範囲は 1 以上 10 以下、 θ (単位は度) は -180 以上 180 以下の実数です。出力は、ひじ関節、手のひらの順にそれぞれの座標を x 、 y 、 z の順に出力してください。各座標値は小数点以下 6 桁まで出力してください。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロみっつの行で示されます。各データセットは以下の通りです。

1行目 $up\ lo\ n$ (整数 整数 整数; 半角空白区切り)

2行目 第1の命令セットの上腕の回転情報 $\theta_{up}\ a_{up}\ b_{up}\ c_{up}$ (実数 実数 実数 実数; 半角空白区切り)

3行目 第1の命令セットの下腕の回転情報 $\theta_{lo}\ a_{lo}\ b_{lo}\ c_{lo}$ (実数 実数 実数 実数; 半角空白区切り)

:

2n行目 第nの命令セットの上腕の回転情報

2n+1行目 第nの命令セットの下腕の回転情報

出力

入力データセットごとに、回転後のひじ関節と手のひらの座標値を出力します。

入力例

```
5 3 1
-90.0 0.0 1.0 0.0
-90.0 1.0 0.0 0.0
5 3 2
180.0 1.0 0.0 1.0
30.0 0.0 1.0 0.0
30.0 0.0 1.0 0.0
30.0 0.0 1.0 0.0
5 5 3
90.0 0.0 0.0 1.0
30.0 1.0 0.0 0.0
75.0 1.0 0.0 0.0
-30.0 1.0 0.0 0.0
45.0 0.0 1.0 0.0
30.0 1.0 0.0 -1.0
0 0 0
```

出力例

```
0.000000 0.000000 5.000000 0.000000 3.000000 5.000000
2.500000 0.000000 4.330127 5.500000 0.000000 4.330127
3.415064 1.294095 3.415064 6.830127 0.000000 6.830127
```

問題

テレビ局 channel12 の番組では今年も「D-1 ダジャレグランプリ 2011(通称 D-1)」という番組を放映することになりました。応募のあったダジャレをスタジオの三人の審査員が審査する毎年恒例の番組です。高評価を得ると一躍有名になれるということもあって、年々応募者が増えています。

その D-1 で、不正な審査が行われたとの告発がありました。D-1 では、審査の公平性を保つために誰がどのダジャレに応募したか審査員には分からないようになっていました。ところが、ある審査員が友人を有名にするため、友人が作ったダジャレに無条件に高評価を与えたというのです。幸い番組は収録だけ行われまだ放送されていません。放送前に不正を見つけ、事態を穏便に収拾することになりました。

不正な審査を行った審査員を割り出したいところですが、利用できる情報は各ダジャレに対する審査員の評価の記録だけです。これでは完全な特定は難しいため、不正な審査を行った可能性が一番高い審査員を割り出すことにしました。

審査員とダジャレの情報は以下の通りです。

1. 各ダジャレは「面白いダジャレ」「つまらないダジャレ」「微妙なダジャレ」の三種類のどれかに属します。
2. 三人の審査員はそれぞれ「正直な審査員」「へそ曲がりな審査員」「笑いの敷居の低い審査員」です。
3. 三人の審査員は各ダジャレを「面白い」か「つまらない」のいずれかで評価します。評価の基準は以下の通りです。
 - (ア) 正直な審査員は面白いダジャレは面白い、つまらないダジャレはつまらないと評価し、微妙なダジャレはその時の気分で評価します。
 - (イ) へそ曲がりな審査員は面白いダジャレはつまらない、つまらないダジャレは面白いと評価し、微妙なダジャレはその時の気分で評価します。
 - (ウ) 笑いの敷居の低い審査員は面白いダジャレは面白い、微妙なダジャレは面白いと評価し、つまらないダジャレはその時の気分で評価します。

不正の判断は以下の基準で行います。

1. 不正な審査とは、上記の条件に従うとつまらないと評価されるはずのダジャレを面白いと評価することを指します。
2. 不正な審査を行った審査員は一人だけで、不正な審査が行われたダジャレも一つだけです。
3. 不正な審査を行った可能性のある審査員が複数いる場合は、不正を行ったと想定される事象の数が一番多い審査員を不正な審査を行った可能性が一番高い審査員とします。

ダジャレの数 n 、各審査員の各ダジャレに対する評価 j のリストを入力とし、不正な審査を行った可能性が最も高い審査員の番号を出力するプログラムを作成してください。出力は、「正直な審査員」の場合は 1、「へそ曲がりな審査員」の場合は 2、「笑いの敷居の低い審査員」の場合は 3 を出力します。ただし、 n は 2 以上 100 以下の整数です。評価 j は、「おもしろい」は f 、「つまらない」は b で表されます。また、不正な審査を行った審査員として一番高い可能性をもつ審査員が複数存在する場合、最も番号が小さい審査員を出力してください。

入 力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロひとつの行で示されます。各データセットは以下のとおりです。

- 1 行目 ダジャレの数 n (整数)
- 2 行目 正直な審査員の各ダジャレに対する評価 $j_1 j_2 \dots j_n$ (すべて半角文字 ; 半角空白区切り)
- 3 行目 へそ曲がりな審査員の各ダジャレに対する評価
- 4 行目 笑いの敷居の低い審査員の各ダジャレに対する評価

出 力

入力データセットごとに、不正な審査を行った審査員として一番高い可能性をもつ審査員の番号を出力します。

入力例

```
2
f f
b f
f f
2
f b
f f
f f
3
b b f
b b f
f f b
0
```

出力例

```
1
2
1
```

問題

求悟君は夏休みの自由研究としてアリの巣の観察を行うことにしました。彼のおじいちゃんが孫のために用意してくれた観察用の透明ケースはとてもユニークで、図1のような形をしています。

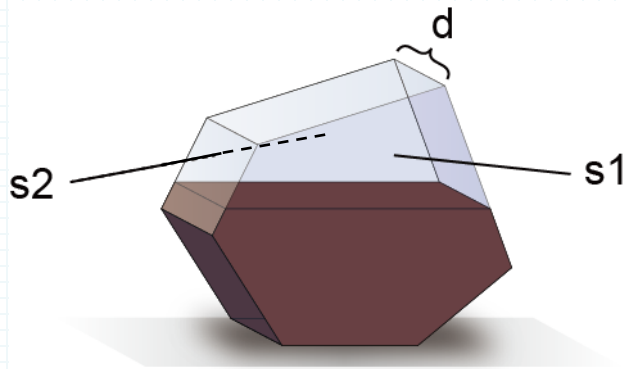


図1

このケースは、2枚の合同な凸多角形 $s1$ 、 $s2$ と何枚かの長方形で構成されています。 $s1$ 、 $s2$ 以外のいずれかの長方形の面を床に接するように置きます。観察を始めた求悟君は、入っている土の量が同じにもかかわらず、底面の選び方によって土の高さが変わること気が付きました（図2）。

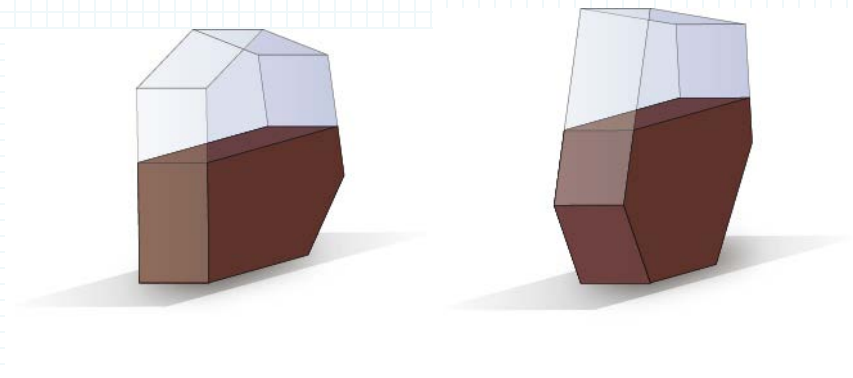


図2

求悟君は、ケースの置き方によって、どのくらい高くなるか知りたくなりました。凸多角形 $s1$ の形、 $s1$ と $s2$ 間の距離 d 、土の体積 V を入力とし、土の高さの最大値を出力するプログラムを作成してください。ただし、ケースに入れる土の粘性は水のように低く、底面を床に置くとすぐに水平になるものとします。透明ケースの $s1$ の形は、二次元の座標平面上の n 個の頂点で与えられます。これらの頂点は、反時計回りの順番で入力されます。 n は3以上100以下で、 d と V はそれぞれ1以上10000以下の整数とします。また多角形の頂点の座標 (x, y) の x, y はそれぞれ-1000以上1000以下の整数とします。土の高さの最大値は小数点以下第7位を四捨五入して、小数点第6位まで出力してください。

入 力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロみっつの行で示されます。各データセットは以下のとおりです。

1 行目 $n d V$ (整数 整数 整数 ; 半角空白区切り)
2 行目 第 1 の頂点の座標 $x y$ (整数 整数 ; 半角空白区切り)
3 行目 第 2 の頂点の座標
:
 $n+1$ 行目 第 n の頂点の座標

出 力

入力データセットごとに、土の高さの最大値を出力します。

入力例

```
4 1 1
0 0
1 0
1 2
0 2
0 0 0
```

出力例

```
1.000000
```

問題

あなたは掃除マシンを使って倉庫を掃除するつもりでしたが、外せない用事があったので、弟に掃除を頼みました。弟は掃除マシンを十分に扱えるのですが、ときどきゴミを残してしまうことがあります。

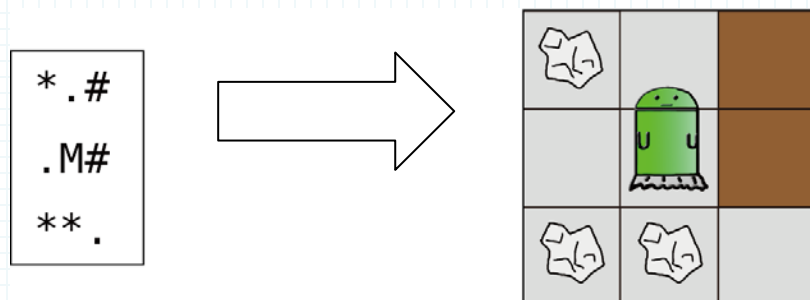
用事を済ませて戻ってきたあなたは、出かける前の倉庫の状態と、弟が操作した掃除マシンの操作履歴から、ゴミが残っていないか確認するためのシミュレーションを行うことにしました。

倉庫の状態は床のパネルに対応した表（横 x 縦 y ）で表されます。表の第1行目（ $y=0$ ）のパネルは、倉庫の一番北に対応します。パネルには以下の状態があります。

- *（半角アスタリスク）： ゴミのあるパネル（ゴミが回収されると「なにもないパネル」になる）
- .（半角ピリオド）： なにもないパネル（ゴミが置かれると「ゴミのあるパネル」になる）
- #（半角シャープ）： 壁
- M（半角英大文字）： 掃除マシン

掃除マシンとゴミのあるパネルは重なることがあります。初期の配置では掃除マシンのあるパネルにはゴミはないものとします。表の外はすべて通れないパネルであるものとして扱います。

また、掃除マシンには向きがあり、東西南北のいずれかの向きを向いています。初期状態では南側を向いています。下図は、横3縦3の表と倉庫の状態の関係をあらわしています。



掃除マシンはリモコン操作式で、リモコンから命令を受けるたびに動作します。命令には大きく以下の3つの種類があります。

- 基本命令 : 前進「go」・左回転「left」・右回転「right」・掃除「vacuum」
- undo : 基本命令を取り消すように動作させる命令
- redo : undoを取り消すように動作させる命令

さらに、この掃除マシンには特殊な機能があります。すなわち、初期設定で基本命令のいくつかをundo禁止命令に指定できます。undo禁止命令に指定されると、undo命令では取り消すことができません。

● 基本命令 (go , left , right , vacuum) について

基本命令の動作は次の通りです。

命令	動作
go	掃除マシンの向いている方向の 1 つ前方のパネルに掃除マシンを移動させる (1 つ前方が壁の場合は何も起きない)。その時、移動先のパネルにゴミがあれば、ゴミを回収してから移動する。
left	掃除マシンの向きを命令実行前の向きから 90 度左方向に回転させる
right	掃除マシンの向きを命令実行前の向きから 90 度右方向に回転させる
vacuum	掃除マシンがあるパネルと、前後左右 (隣接している 4 つ) のパネルにあるごみを回収する。

● undo と redo について

undo は基本命令の動作を取り消す動作を実行させます。redo は undo で取り消された基本命令を再度実行させます。

命令	動作
undo	直前に実行された基本命令を取り消す動作を行う。取り消そうとする基本命令がすでに一度 undo で取り消す動作を行われた基本命令か、初期設定で指定された undo に無視される基本命令の場合には、その基本命令を無視し、さらに 1 つ前の基本命令に対し取り消す動作を行う。即ち一連の基本命令列を時間的に遡り undo に無視されることのない基本命令を探し、それを取り消す動作を行う。ただし該当する基本命令がない場合、取り消し動作は何も行われぬ。各基本命令に対する取り消し動作は次のとおりである。 go を取り消す場合には、1 つ後方のパネルに進む (そのパネルが壁の場合は移動しない)。その後、実行時にゴミを回収していた場合、掃除マシンの 1 つ前方のパネルにゴミを置く (そのパネルが壁の場合はゴミを置かない)。 left を取り消す場合には、掃除マシンの向きを 90 度右方向に回転させる right を取り消す場合には、掃除マシンの向きを 90 度左方向に回転させる vacuum を取り消す場合には、動作時にゴミを回収した方向にゴミを置く (動作時に右のゴミを回収していた場合、取り消し時には右にゴミを置く。ゴミを置こうとするパネルが壁の場合、そのパネルにはゴミを置かない)
redo	直前の undo により取り消された基本命令を実行する。redo が連続した場合、直前に redo された undo の、さらに 1 つ前の undo により取り消された基本命令を実行する。 なお、redo によって実行された基本命令も、undo の対象になりうる。 直前に undo がない場合、もしくは直前の undo で何も取り消し動作が行われていない場合、何も起きない。



初期設定 : right

go → left → go → go →
vacuum → right → go →
left → go → go → undo →
undo → redo → vacuum →
right → undo



倉庫を表す表の横 x 、縦 y 、出かける前の倉庫の状態、undo で無視される命令の個数 n 、操作履歴に記録された命令の個数 m 、undo で無視される命令 p 、操作履歴に記録された命令 c を入力とし、命令どおりに掃除機を動かした場合の最終的な倉庫の状態を出力してください。掃除マシンがゴミのあるパネルにある場合には、掃除マシンであることを優先して出力します。ただし、 y 、 x は 3 以上 100 以下の整数、 n は 0 以上 4 以下の整数、 m は 1 以上 10000 以下の整数です。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロふたつで表されます。各データセットは以下の通りです。

1 行目 x y (整数 整数; 半角空白区切り)

2 行目 n m (整数 整数; 半角空白区切り)

3 行目 undo で無視される命令 $p_1 \dots p_n$ (すべて半角文字列; 半角空白区切り)

4 行目 倉庫の第 1 行目の初期状態 $s_1 \dots s_x$ (すべて半角文字)

:

$y+3$ 行目 倉庫の第 y 行目の初期状態

$y+4$ 行目 1 番目の命令 c (半角文字列)

$y+5$ 行目 2 番目の命令

:

$y+m+3$ 行目 m 番目の命令

出力

入力データセットごとに、最終的な倉庫の状態を出力してください。

入力例

```
6 4
1 16
right
M***##
*.**##
##****
****..
go
left
go
go
vacuum
right
go
left
go
go
undo
undo
redo
vacuum
right
undo
4 3
0 5
####
M**#
####
left
go
go
go
undo
0 0
```

出力例

```
.*.*##
....##
##*M.*
****..
####
.M.#
####
```

問題

愛歌さんの家は、小さな喫茶店を経営しています。お母さんが焼くスコーンはとても美味しくて、店はとても繁盛しています。愛歌さんは、焼きあがったスコーンを、開店直後に一度だけお客様の席まで届けるお手伝いをします。

焼きあがったスコーンはお盆に乗せられます。カウンターの上には、そのようなお盆が一行に並べられています。お盆の大きさは様々で、その上のスコーンの個数も様々です。愛歌さんは、カウンターから連続した範囲のお盆を持ち出し、お客様にスコーンを配ってまわります。一方、この店では各お客様に必ず m 個のスコーンを一度に配ります。ひとつのお盆から複数のお客様に配ったり、複数のお盆からひとりのお客様に配ったりします。

このような配り方をすると、配ることができずにスコーンが余ることがあります。しかし、愛歌さんが配る開店直後にはお客様が十分にいるため、 m 個以上余ることはありません。この余ったスコーンは、ご褒美として愛歌さんがもらえます。

ある日、愛歌さんは、カウンターから持ち出すお盆の範囲によって、余るスコーンの数が違ってくることに気づきました。上手にお盆の範囲を選ぶことでより多くのスコーンが余るようにできるかもしれません。

お盆の数 n 、各お客様に配るスコーンの数 m および n 個のお盆上のスコーンの情報を入力とし、愛歌さんがご褒美としてもらえるスコーンの最大数を出力するプログラムを作成してください。ただし、 n は 1 以上 10000 以下、 m は 1 以上 100000 以下とします。また、スコーンの情報は、 n 個のお盆上のスコーンの数がそれぞれ 0 以上 $2^{31}-1$ 以下の整数として与えられます。

入 力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロふたつの行で示されます。各データセットは以下のとおりです。

1行目 n m (整数 整数 ; 半角空白区切り)

2行目 お盆上のスコアの情報 k_1 k_2 ... k_n (すべて整数 ; 半角空白区切り)

k_i : i 番目のお盆上のスコアの数

出 力

入力データセットごとに、もらえるスコアの最大数を出力します。

入力例

```
5 11
11 27 34 45 56
8 5
0 2 1 5 4 6 8 3
5 2
2 4 2 4 6
10 18
10 15 12 31 12 50 11 23 43 181
1 100
5
0 0
```

出力例

```
8
4
0
17
5
```

問題

宇宙暦 119 年。古代宇宙人が遺したとされる惑星ヅダイの遺跡調査に向かったベコーズ隊は、地下に巨大な神殿があることを突き止めました。探索ロボットを遠隔操作して神殿内部を調査したところ、不思議な力をもつ硬貨を発見しました。

その硬貨には複雑な紋章が刻まれており、神殿内にある同じ紋章が彫られた扉に近づけると、硬貨が 1 枚消えてなくなる代わりに一定時間だけ扉が開くのです。これにより、神殿を崩さずに移動することができました。

硬貨の力を使って一通り神殿の調査を済ませた探索ロボットは、ついに最深部までたどり着きました。ここまで持っていた硬貨は使いきってしまいましたが、神殿内にはまだ他にも落ちているので回収はできます。またベコーズ隊は、これまでの調査で神殿内部の状況を把握することに成功しました。その情報をもとに、なるべく多くの硬貨をロボットに持ち帰らせたいと考えています。

神殿内部は横 w × 縦 h のマスで構成される 2 次元グリッドの構造をしています。各マスは通れる道・壁・扉・硬貨・初期位置・神殿出口のいずれかです。ロボットは壁以外のマスに移動でき、1 マス移動する度に時間が 1 ステップ経過します。移動は縦横のみで斜めには移動できません。“硬貨”のマスには、初期状態で 1 枚の硬貨が置かれており、そのマスに移動するとその硬貨は入手したものとします（入手した硬貨はマス上から消え、そのマスはただの道として扱われます）。“扉”のマスには対応する硬貨を持っていないと進入することができません。扉のあるマスに進入すると手持ちの対応する硬貨が 1 枚失われます。扉は半角大文字の $A \cdot B \cdot C$ 、扉を開けるための硬貨は半角小文字の $a \cdot b \cdot c$ で表され、 A と a 、 B と b 、 C と c がそれぞれ対応しています。一度開いた扉のマスは一定時間通行可能で、その後また閉じられます。開いている時間は以下の通りです。

A : 30 ステップ、 B : 20 ステップ、 C : 10 ステップ

扉が開いている間は硬貨を持っていなくても移動ができ、硬貨を持って移動した場合は硬貨を失いません。扉は開けられたステップ時より 0 から時間の計測を始め、規定されたステップ数以上に達した場合に閉じられます。なお、計測はステップごとにロボットの移動が完了した後に行われ、扉が閉じられるマスにロボットがいる場合は、そのステップでは扉が閉じられることは無く、次のステップで他のマスにロボットが移動した時に閉じられます。

初期状態では扉は全て閉じられており、硬貨は 1 枚も持っていません。

神殿通路の状態を入力とし、神殿から脱出可能か判断し、可能であれば脱出時に最も多く硬貨を残せたときの 3 種類の硬貨の数の合計を出力するプログラムを作成してください。ただし、 w は 1 以上 50 以下の整数、 h は 1 以上 50 以下の整数とし、落ちている硬貨は最大で 10 枚以下です。また、扉は同じ種類の扉が 2 つ以上存在することはありません。脱出出来ない場合は NA と出力します。

入力

複数のデータセットの並びが入力として与えられます。入力の終わりはゼロふたつの行で示されます。各データセットは以下のとおりです。

1 行目 神殿の大きさ 縦 w 横 h (整数 整数 ; 半角空白区切り)

2 行目 神殿の 1 行目の情報 $d_1 d_2 \dots d_w$ (半角文字)

各記号の意味は以下の通り。

通れる道 : - (半角ハイフン)

通れない道 (壁) : * (半角アスタリスク)

ロボットの位置 : R

神殿出口 : G

扉 A : A

扉 B : B

扉 C : C

硬貨 a : a

硬貨 b : b

硬貨 c : c

3 行目 神殿の 2 行目の情報

:

$h+1$ 行目 神殿の h 行目の情報

出力

入力データセットごとに、脱出時に持っている硬貨の数の合計または NA を出力します。

入力例

```
6 5  
R-----  
A****-  
a-b*c-  
-*.**-  
b-a-CG  
0 0
```

出力例

```
3
```