

パソコン甲子園 2009

予選問題集

問題

会津若松市では、毎年1月10日に「十日市」という初市があります。この十日市は、約600年の歴史があり会津地方最大の初市です。会津地方ではお馴染みの縁起物である、起き上がり小法師（おきあがりこぼし）が売られていることでもよく知られています。起き上がり小法師は、大きさ3cm程度の重心が底にある張り子で、転がしてもすぐに起きあがることからこの名前が付けました。各家庭では、必ず家族より一個多く買って神棚に供えます。この一個は、「家族が増えるように」や「厄を背負ってくれる」との意味があります。



十日市実行委員会では、次回の十日市に向けて、起き上がり小法師の販売個数が一番多い出店を調べることになりました。今年の出店の数は5店（A、B、C、D、E：半角英字）で、販売個数は、午前と午後に分けて十日市実行委員会に報告されています。

各出店の情報を入力とし、一日の販売個数が一番多い出店の名前とその個数を出力するプログラムを作成してください。ただし、一日の販売個数が同じ出店は無いものとし、販売個数 s は1以上10000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 出店 A の情報 $s_1 s_2$ （整数 整数；半角空白区切り）

各記号の意味は以下のとおりです。

s_1 : 午前の販売個数

s_2 : 午後の販売個数

2 行目 出店 B の情報

3 行目 出店 C の情報

4 行目 出店 D の情報

5 行目 出店 E の情報

出力

入力データセットごとに、一日の販売個数の一番多い出店の名前とその個数を出力します。

入力例

1593 4311
4321 2155
1256 6421
5310 1455
2152 5421
1549 3386
4528 3719
1234 4321
3330 3109
2739 2199
0 0

出力例

C 7677
B 8247

問題

野球の国別対抗戦 WBC で、日本が 2 連覇達成！！

野球人気が高まる中、会津学園高校を会場に野球大会が行われました。この大会では、総当りのリーグ戦を行い、以下のような方法で順位を決めることになりました。

- 1) 勝ち数の多いチームを上位とする
- 2) 勝ち数が同数の場合は負け数の少ないチームを上位とする

各チームの成績を入力とし、チーム名を上位のチームから順に出力するプログラムを作成してください。同順位のチームが存在する場合は、入力順に出力してください。ただし、チーム数 n は 2 以上 10 以下の整数、チーム名 t は 1 文字の半角英字、試合毎の成績 r は $n-1$ 個の数字で表され、勝ちの場合は 0、負けの場合は 1、引き分けの場合は 2 とします。また、チーム名に重複はないものとします。



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 チームの数 n (整数)
2 行目 第 1 のチームの成績 $t r_1 \cdots r_{n-1}$ (半角英字 整数 \cdots 整数 ; 半角空白区切り)
各記号の意味は以下のとおりです。
 t : チーム名
 $r_1 \cdots r_{n-1}$: チーム t の試合毎の成績
3 行目 第 2 のチームの成績
:
 $n+1$ 行目 第 n のチームの成績

出力

入力データセットごとに、チーム名を上位のチームから順に出力します。

入力例

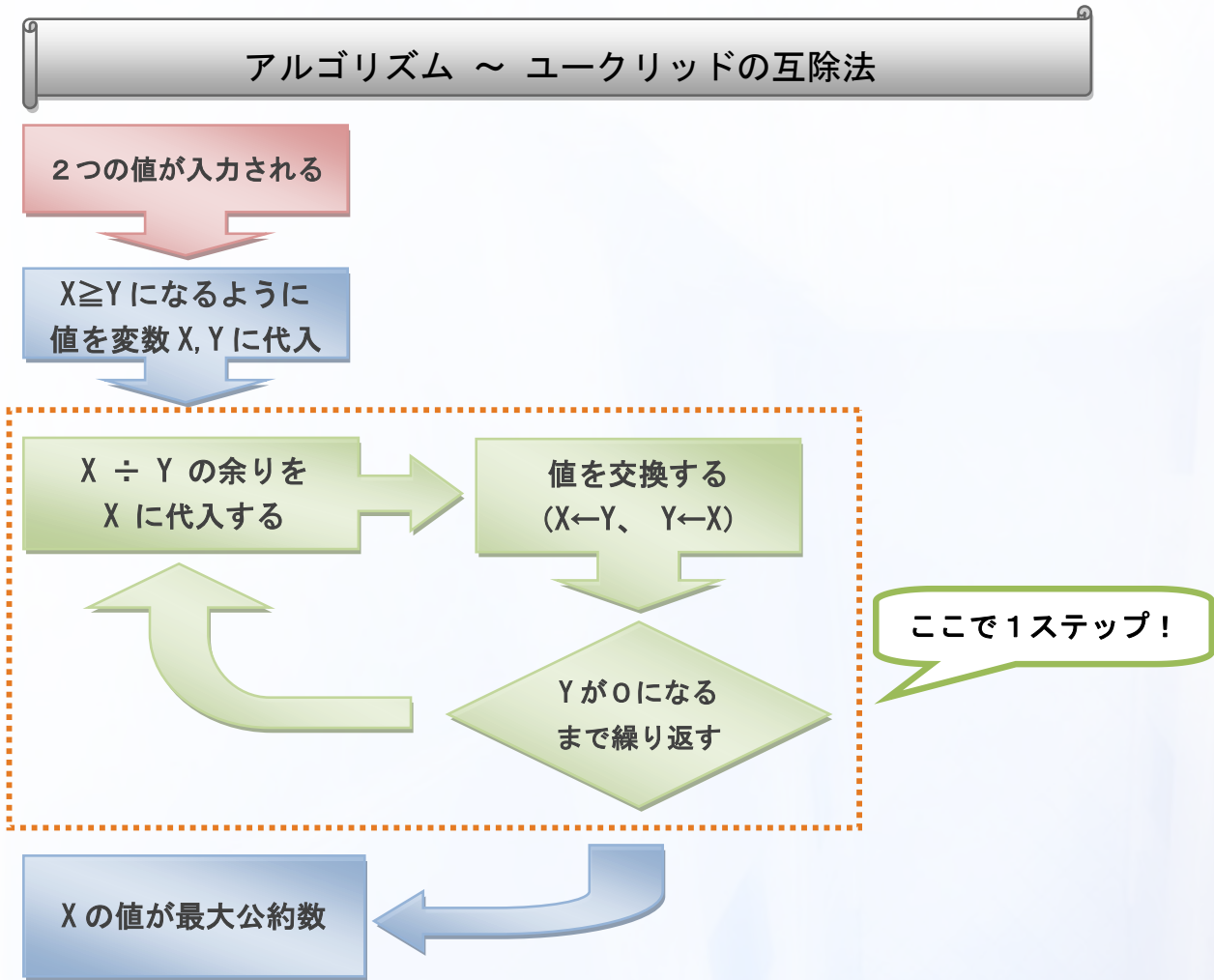
```
6
A 1 0 0 2 0
B 0 0 1 1 0
C 1 1 1 1 1
D 1 0 0 1 2
E 2 0 0 0 0
F 1 1 0 2 1
4
g 1 1 1
h 0 1 2
w 0 0 0
b 0 2 1
0
```

出力例

```
E
A
B
D
F
C
w
h
b
g
```

問題

最大公約数を利用するというのは、コンピュータ上で行う数学には欠かせない要素です。最大公約数を使うことで、計算の効率が大きく変動することもあります。最大公約数を求めるアルゴリズムにはいろいろありますが、「ユークリッドの互除法」と呼ばれるアルゴリズムもその一つです。その処理の流れを以下に示します。



例えば 1071 と 1029 の場合、1071 を X 、1029 を Y に代入して、
 $1071 \div 1029$ の余りは 42、 X に 42 を代入して X と Y を入れ替える。(1 ステップ)
 $1029 \div 42$ の余りは 21、 X に 21 を代入して X と Y を入れ替える。(2 ステップ)
 $42 \div 21$ の余りは 0、 X に 0 を代入して X と Y を入れ替える。(3 ステップ)
 Y が 0 になったので、この時の X が最大公約数となる。よって最大公約数は 21。

このように、たったの 3 ステップで 1071 と 1029 の最大公約数を求めることが出来ました。ユークリッドの互除法は約数を出して比較していく方法に比べ、圧倒的に早く結果を出してくれます。

ふたつの整数を入力とし、ユークリッドの互除法を用いて最大公約数を求め、その最大公約数と計算にかかったステップ数を出力するプログラムを作成してください。
 ただし、与えられる整数は 2 以上 $2^{31}-1$ 以下とします。

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。

ふたつの整数（整数 整数；半角空白区切り）

出力

入力データセットごとに、入力されたふたつの整数の最大公約数と、計算にかかったユークリッドの互除法のステップ数を出力します。

入力例

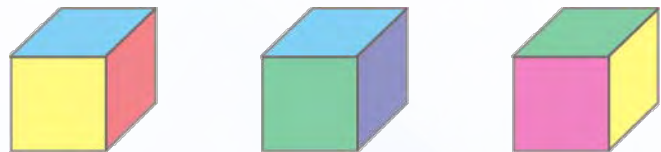
```
1071 1029
5 5
0 0
```

出力例

```
21 3
5 1
```

問題

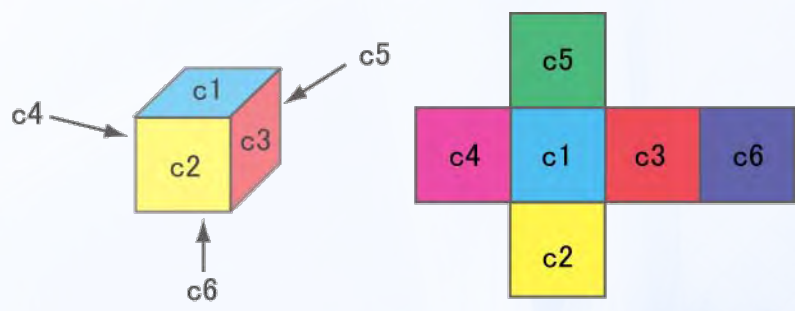
芸術家品川は n 点の作品を出展するように依頼されました。そこで、立方体の 6 面をペンキで色付けしたものを作品として出展することにしました。作品は、Red、Yellow、Blue、Magenta、Green、Cyan の 6 色全てが使われていて、各面は 1 色で塗りつぶされています。品川は形が同じ立方体の作品でも色の配置の仕方を変え、異なる作品として n 点作成しました。



彼の友人であるあなたは、友人のよしみで作品を出展前に閲覧させてもらいましたが、そこであることに気が付きました。それらの作品の中に異なる色付けをされているように見えても、実は同じ色の組み合わせをした立方体が存在していたのです。このままでは、n 点の作品を出展することができなくなってしまいます。

作成した作品の数と各作品の色の情報を入力とし、出展するのにあと何点必要かを出力するプログラムを作成してください。ただし、作成した作品の数は 1 以上 30 以下とします。

立方体の各面の色は c1 から c6 までの記号で表され、以下のような配置となっています。また、c1 から c6 のそれぞれは Red、Yellow、Blue、Magenta、Green、Cyan のいずれか 1 色となります。



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  作品の数  n  (整数)
2 行目  第 1 の作品の色の配置  c1 c2 c3 c4 c5 c6
      (それぞれ半角文字列; 半角空白区切り)
3 行目  第 2 の作品の色の配置
      :
n+1 行目  第 n の作品の色の配置
```

出力

入力データセットごとに、出展するのにあと何点の作品が必要かを出力します。

入力例

```
3
Cyan Yellow Red Magenta Green Blue
Cyan Yellow Red Magenta Green Blue
Red Yellow Magenta Blue Green Cyan
4
Red Magenta Blue Green Yellow Cyan
Red Yellow Magenta Blue Green Cyan
Magenta Green Red Cyan Yellow Blue
Cyan Green Yellow Blue Magenta Red
0
```

出力例

```
1
1
```

問題

1/2

4つの国の中心にある中立都市アイズシティには、大陸横断列車バンダイ号のホームがあります。ホームには、バンダイ号を待つ乗客用の椅子が一行に用意されており、ホームに入ってきた人は自由に椅子を利用することができます。

このバンダイ号、安いし早いし快適ということで、周囲の4つの国からの利用者が絶えません。今日は開通記念日ということで、ホームに座っている人たちに特別なことをしてあげようと思っています。そのためには、改札を通った人たちがどこに座っているか把握しなければなりません。4つの国の人たちの難儀な性格を考慮して、椅子がどのように埋まっているかをシミュレートするプログラムを作成してください。改札を通った人たちは、次々に一行に並んだ椅子に座っていきます。4つの国の人たちは、国ごとに決まった性格と座り方があります。それぞれの座り方は以下の通りです。

<p>A 国人の性格</p>	<p>A 国人は座ればばいい。左端から見ていき空いている椅子に座る。</p>	
<p>B 国人の性格</p>	<p>B 国人は A 国人が苦手。A 国人の隣以外で、右端から空いている椅子に座る。ただし、A 国人の隣しか空いていない場合、我慢して左端から空いている椅子に座る。</p>	
<p>C 国人の性格</p>	<p>C 国人は人の隣に座りたい。左側から順に座っている人を見ていき、一番左側に座っている人の右隣に座ろうとするが、そこが埋まっているならその人の左隣に座ろうとする。そこも埋まっているなら次の人の隣に同条件で座ろうとする。どの椅子にも人が座っていなければ真ん中（椅子の数が奇数の場合 $(n+1)/2$、偶数の場合 $n/2+1$）の椅子に座る。</p>	
<p>D 国人の性格</p>	<p>D 国人は人の隣に座りたくない。一番近い人との距離が、一番大きくなる椅子に座ろうとする。同条件の椅子が複数ある場合や、どうしても誰かの隣に座らなければならない場合、その中で一番左側にある椅子に座る。誰も座っていない場合は左端の椅子に座る。</p>	

問題

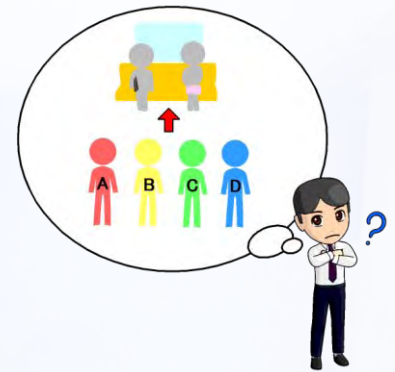
2/2

バンダイ号に乗ろうとしている乗客の情報を入力とし、椅子にどのように座っているかを出力するプログラムを作成してください。ただし、椅子の数は 1 以上 10000 以下とし、乗客の数は椅子の数を超えることはありません。左から順番に座っている人の国籍を出力します。ただし、空席の場合は # (半角シャープ) 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 椅子の数 n 乗客の数 m (整数 整数 ; 半角空白区切り)
- 2 行目 1 人目の情報 a (半角文字)
A ならば A 国人
B ならば B 国人
C ならば C 国人
D ならば D 国人
- 3 行目 2 人目の情報
- 4 行目 3 人目の情報
- ⋮
- $m+1$ 行目 m 人目の情報



出力

入力データセットごとに、最終的な椅子の状態を出力します。

入力例

```
5 4
A
B
C
D
5 4
D
C
B
A
0 0
```

出力例

```
ACD#B
DCA#B
```

問題

高校二年生の会津太郎君は夏休みに電車で長旅をする計画を立てています。しかし高校生の身である太郎君が一ヵ月しかない夏休みで可能な限り遠くに旅をするには、出来るだけ安い行き方と出来るだけ早い行き方をそれぞれ見つけなければうまく計画が立てられません。太郎君が素敵な旅を満喫できるように、太郎君の計画の助けになるプログラムを作ってあげましょう。



線路の情報、駅の数を入力とし、問い合わせに応じて、最小金額または最短時間を出力するプログラムを作成してください。線路の情報の数 n は 1 以上 300 以下の整数、駅の数 m は 1 以上 100 以下の整数とし、各駅は 1 から m まで順番に番号が付けられているものとします。また、線路の情報は、線路がつなぐ二つの駅 (a, b) 、料金 $cost$ 、移動時間 $time$ で表され、 $time$ と $cost$ はそれぞれ 1 以上 1000 以下の整数とします。なお、 a と b が線路でつながっていれば、 a から b 、 b から a の両方の移動が同じ料金と時間で可能とします。問い合わせは、出発駅 p 、到着駅 q 、出力する値の種類 r で表され、 r が 0 の時は最小金額を、1 の時は最短時間を出力します。問い合わせの数 k は 1 以上 100 以下です。なお、問い合わせには必ず経路があるものとします。

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 線路の情報の数 n 駅の数 m (整数 整数; 半角空白区切り)
2 行目 第 1 の線路の情報 a b $cost$ $time$ (整数 整数 整数 整数; 半角空白区切り)
3 行目 第 2 の線路の情報
:
 $n+1$ 行目 第 n の線路の情報
 $n+2$ 行目 問い合わせの数 k (整数)
 $n+3$ 行目 第 1 の問い合わせ p q r (整数 整数 整数; 半角空白区切り)
 $n+4$ 行目 第 2 の問い合わせ
:
 $n+2+k$ 行 第 k の問い合わせ

出力

入力データセットごとに、最小金額もしくは最短時間を出力します。

入力例

```
6 5
1 2 200 10
1 4 400 15
1 3 250 25
2 4 100 10
4 5 150 20
3 5 300 20
2
1 5 0
1 5 1
0 0
```

出力例

```
450
35
```

問題

勇者はついに魔法の釜、錬金釜を手に入れました。錬金釜に複数のアイテムを入れると、新しいアイテムを作ることができます。新しく作ったアイテムは、他のアイテムを作るために錬金釜へ入れることもできます。アイテムを作るために必要なアイテムのリストを記したものを錬金レシピと呼ぶことにします。以下の3つは錬金レシピの例です。

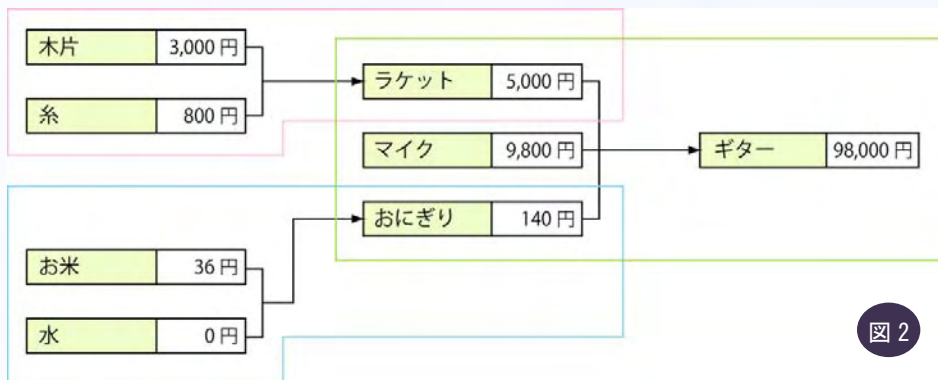
- 1) 木片と糸でラケットができる。
- 2) お米と水でおにぎりができる。
- 3) ラケットとマイクとおにぎりでギターができる。

アイテムはお金を使って手にいれることもできますが、錬金釜で作った方が安くすむ場合もあります。例えば、各アイテムの購入価格が図1のように与えられているとします。

図1

アイテム名	購入価格 (円)
木片	3,000
糸	800
お米	36
水	0
ラケット	5,000
マイク	9,800
おにぎり	140
ギター	98,000

ラケットは 5,000 円で購入できますが、木片と糸を購入して錬金すれば 3,800 円で手に入れます。更に、錬金を重ねることによりとてもお得にアイテムを手に入れます。図2は上記3つの錬金レシピ例を組み合わせたものです。ギターは 98,000 円で購入できますが、木片、糸、お米、水、マイクを購入して錬金すれば、たったの 13,636 円で手に入れます



勇者は冒険を楽に進めるため、なるべく安く目的のアイテムを手に入れようと考えました。

アイテムのリスト、錬金レシピのリスト、指定されたアイテムを入力とし、指定されたアイテムを作るために必要な金額の最小値を出力するプログラムを作成して下さい。

アイテムリストに含まれるアイテムの種類 n は 1 以上 100 以下の整数とし、各アイテムを購入する場合の値段 p は 0 以上 1000000 以下とします。アイテムの名前 i は 1 文字以上 100 文字以下のアルファベットからなる半角文字列です。錬金レシピの種類 m は 0 以上 100 以下の整数です。1 つの錬金レシピはアイテムの名前 o 、それを作るために必要なアイテムの個数 k 、 k 個のアイテム名 q のリストによって与えられます。 k は 100 以下とします。

各アイテムの数量に限りはないものとします。アイテムは複数のレシピに使われることがあります。1 つのアイテムを作るためのレシピはただか1つです。

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 リストに含まれるアイテムの数 n (整数)
- 2 行目 第 1 のアイテムの情報 $i p$ (半角文字列 整数;半角空白区切り)
- 3 行目 第 2 のアイテムの情報
- :
- $n+1$ 行目 第 n のアイテムの情報
- $n+2$ 行目 錬金レシピの数 m (整数)
- $n+3$ 行目 第 1 の錬金レシピの情報 $o k q_1 q_2 \dots q_k$
(半角文字列 整数 半角文字列 \dots 半角文字列 ; 半角空白区切り)
- $n+4$ 行目 第 2 の錬金レシピの情報
- :
- $n+m+2$ 行目 第 m の錬金レシピの情報
- $n+m+3$ 行目 指定されたアイテム名 (半角文字列)

出力

入力データセットごとに、最小の値段を出力します。

入力例

```
8
wood 3000
string 800
rice 36
water 0
racket 5000
microphone 9800
onigiri 140
guitar 98000
3
racket 2 wood string
onigiri 2 rice water
guitar 3 racket microphone onigiri
guitar
1
computer 300000
0
computer
0
```

出力例

```
13636
300000
```

問題

会津太郎さんの会社には、割り切れない事が大嫌いな上司がいます。太郎さんがその上司と食事に行くときは、割り勘で会計をしているのですが、支払金額が参加人数で割り切れないときは、いつも上司がおごってくれています。

ある日、太郎さんは食事会の幹事になりました。お金の少ない太郎さんは、その上司を誘ってなんとかおごってもらえるように出来ないか考えました。もう料理屋に注文をしなければならないのですが、まだ何人参加するかは分からないので、どんな人数が参加してもおごってもらえるような注文をしておきたいようです。太郎さんの同期で、同じく食事会に参加する予定のあなたは、太郎さんに協力して、予算額以下で最大のどんな人数でも割り切れない金額を算出することにしました。

200 円		¥ 200
400 円		¥ 200
800 円	25,000 円	¥ 1600
1,600 円	14,000 円	¥ 800
	9,000 円	¥ 800
		¥ 9000
		:
		合計 ?

料理の種類、各料理の料金、予算額を入力とし、予算額以下で最大のどんな数字でも割り切れない合計金額（ただし、1 と合計金額は除く）を出力するプログラムを作成してください。なお、各種類の料理は複数個注文できますが、全種類の料理を注文する必要はありません。ただし、料理の種類は 1 以上 30 以下、予算額は 1 以上 1000000 以下とします。また、このような合計金額がない場合は、NA 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 料理の種類 n 予算額 x (整数 整数; 半角空白区切り)
2 行目 1 種類目の金額 (整数)
3 行目 2 種類目の金額
:
n+1 行目 n 種類目の金額

出力

入力データセットごとに、予算額に最も近い合計金額、または NA を出力します。

入力例

```
4 15000
305
260
129
500
3 400
10
20
30
3 200909
5
9
12
0 0
```

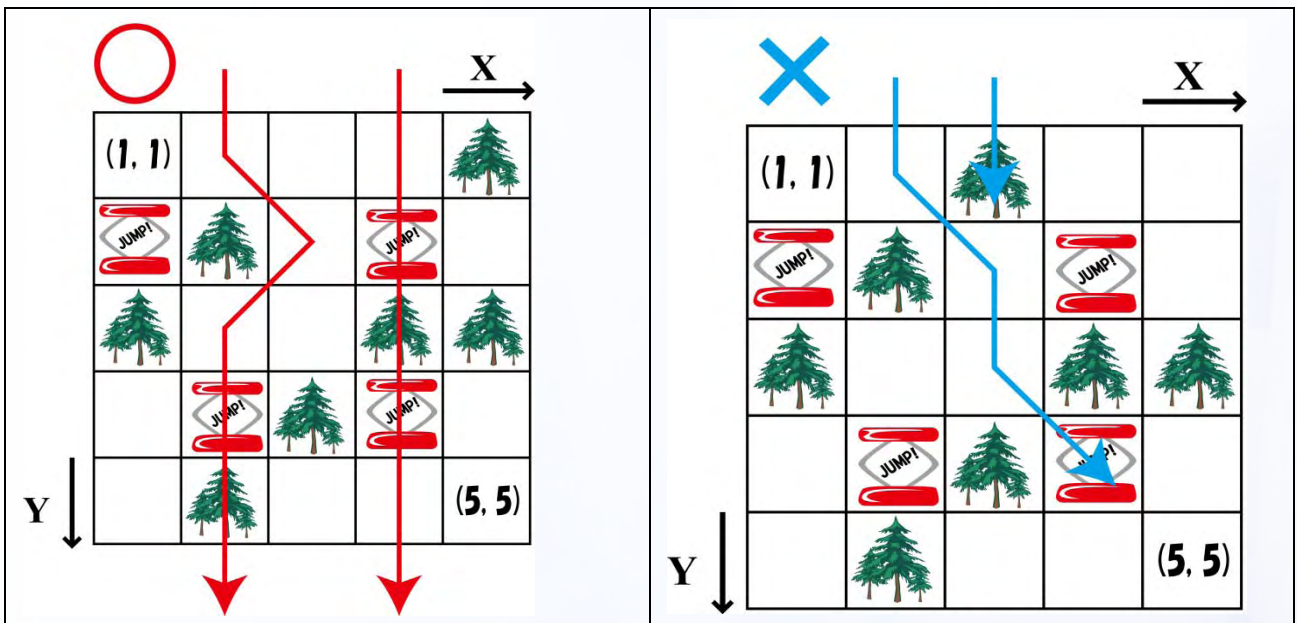
出力例

```
14983
NA
200909
```

問題

会津山スキー場の経営者である油木屋さんは、上級者向けとして障害物やジャンプ台を配置したコースを用意しました。コースにはいろいろな滑り方があり、シーズン中に全てのパターンの滑り方が出来た利用者には、プレゼントを贈ることになっています。

しかし、オープン直前になって、油木屋さんは滑り方のパターン数を記録した書類をなくしてしまいました。どうにかして書類を作り直したい油木屋さんは、コースの見取り図を片手にあなたに助けを求めてきました。油木屋さんのために、コースの見取り図をもとに滑り方のパターン数を出力するプログラムを作ってあげましょう。



コースは上図のようなグリッドで表されます。左上を原点とし、x 座標は右に行くにつれて大きくなり、y 座標は下にいくにつれて大きくなるものとします。ただし、コースの大きさ X と Y はそれぞれ 1 以上 15 以下とします。

各滑り方のパターンは、最も高いところ ($y = 1$ 、ただし障害物の無いところ) からスタートし、ゴール ($y = Y$) に向かって進んでいきます。グリッドのマス (x, y) にいる滑走者は、($x-1, y+1$)、($x, y+1$)、($x+1, y+1$) のいずれかに移動することができます。マスには、障害物やジャンプ台があり、障害物のあるマスには進入できず、ジャンプ台があるマスに進入すると ($x, y+2$) へ移動します。ただし、いちばん高いマス ($y = 1$ のマス) にはジャンプ台は存在せず、ジャンプ台のあるマスに進入する際には x 座標が同じマスからしか進入できません。コースの上端 ($y = 1$) からスタートし、コースからはずれることなく下端を超えれば ($y \geq Y$) 1つの滑り方とみなし、滑り終わります。

コースの情報を入力とし、滑り方の総数を出力するプログラムを作成してください。



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ふたつの行で示されません。各データセットは以下のとおりです。

- 1 行目 コースの大きさ横 X 縦 Y (整数 整数 ; 半角空白区切り)
- 2 行目 コースの 1 行目の情報 $c_1 c_2 \dots c_X$ (整数 整数 \dots 整数 ; 半角空白区切り)
各記号の意味は以下のとおりです。
 c_i : この行における $x = i$ 番目のグリッド の情報 (0 , 1 , 2 のいずれか)
0 : 移動可能である
1 : 障害物がある
2 : ジャンプ台がある
- 3 行目 コースの 2 行目の情報
:
 $Y+1$ 行目 コースの Y 行目の情報

出力

入力データセットごとに、コースの滑り方のパターン数を出力します。

入力例

```
5 5
0 0 0 0 1
2 1 0 2 0
1 0 0 1 1
0 2 1 2 0
0 1 0 0 0
5 5
0 0 1 0 0
2 1 0 2 0
1 0 0 1 1
0 2 1 2 0
0 1 0 0 0
15 15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0

```

出力例

```
8
6
52694573
```

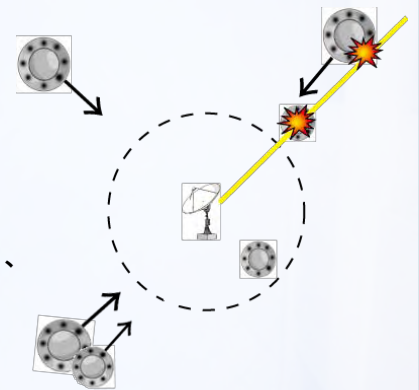
問題

40XX 年、地球は宇宙人の侵攻を受けていた！すでに地球のほとんどは宇宙人により制圧されており、残る防衛拠点は鶴ヶ城要塞のみになってしまった。その鶴ヶ城要塞にも、制圧部隊が次々に迫ってきている。3 時間後には最初の戦闘が始まるだろう。このままでは地球最後の砦も制圧されてしまう。

しかし、希望は残されている。防衛軍の最終兵器、超長距離貫通レーザー砲が完成したのだ。欠点といえば、威力が出るまでに一定の距離が必要で、近すぎる敵はただ通りぬけてしまうということだ。

威力が出ない範囲に侵入した敵は、他の戦力でどうにかするしかない。防衛軍の参謀は、侵入してきた UFO に対処するために、その数を知らなければならないのだが、あまりにも敵が多いせいでうまく数えられそうにない。そこで参謀はあなたに、レーザーに撃墜されなかった UFO の数を出力するプログラムを用意するように命令した。戦闘開始までにプログラムを作成し、鶴ヶ城要塞を守る力となるろう。

敵の UFO はただまっすぐにレーザー砲のある拠点を狙って突撃してきます（UFO には互いにすり抜ける処理が施されており、衝突してしまふことはありません）。レーザー砲は、初期状態から 1 分後に最も近い UFO の中心を狙ってレーザーを発射し始め、それから 1 分ごとに同じ条件でレーザーを発射し続けます。レーザーは貫通し、その先にある UFO もレーザーがかすただけで撃墜することができます。しかし、このレーザーには威力の出ない範囲があり、その範囲に入ってしまった UFO は狙っても意味がないので、狙わないように設計されています。撃墜できるだけ撃墜したとき、レーザーの威力が出ない範囲には何機の UFO が侵入したでしょうか。



レーザーの威力が出ない範囲の半径 R と、襲来する UFO の情報を入力とし、撃墜されず侵入している UFO が何機いるかを出力するプログラムを作成してください。襲来する UFO の情報は、UFO の数 N 、各 UFO の初期座標 (x_0, y_0) 、各 UFO の半径 r 、各 UFO の分速 v で構成されます。

座標の原点 $(0, 0)$ をレーザー砲の位置とします。レーザー砲と UFO の距離は原点から UFO の中心までの距離で与えられ、この距離が R 以下の UFO はレーザーの威力が出ない範囲に入ったとします。狙うべき対象が同時に複数存在する場合はないものとして考えます。計算は全て平面上で考え、入力はすべて整数で与えられます。ただし、 N は 1 以上 100 以下、 R と r 及び v はそれぞれ 1 以上 500 以下、 x_0 、 y_0 は -1000 以上 1000 以下とします。

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 レーザーの威力が出ない範囲の半径 R UF0 の数 N (整数 整数 ; 半角空白区切り)
- 2 行目 1 機目の UF0 の情報 x0 y0 r v (整数 整数 整数 整数 ; 半角空白区切り)
- 3 行目 2 機目の UF0 の情報
- ⋮
- N+1 行目 N 機目の UF0 の情報

出力

入力データセットごとに、レーザーの威力が出ない範囲に侵入した UF0 の数を出力します。

入力例

```
100 5
101 101 5 5
110 110 2 3
-112 -100 9 11
-208 160 82 90
-110 108 10 2
10 11
15 0 5 1
25 0 5 1
35 0 5 1
45 0 5 1
55 0 5 1
65 0 5 1
75 0 5 1
85 0 5 1
95 0 5 1
-20 0 5 20
-30 0 500 5
0 0
```

出力例

```
1
1
```