

問題

三度の飯よりゲーム好きのだigo君は暇さえあればゲームばかりしていました。ですが、だigo君も高校3年生。クラスはもう受験モードに入っています。

受験勉強をするからには、当然ゲームで遊べる時間が短くなってしまいます。まずは1日に勉強をする時間を決めて、残った時間をゲームなどに使うことにしましたが、だigo君は今まで計画を守れたためしがありませんでした。

見かねたあなたは、だigo君の勉強時間を管理するプログラムを書くことにしました。



1日にやらなければならない勉強の目標時間を t 、勉強を開始した時刻を s 、終了した時刻を f とおきます。1日で勉強した時間の合計が t 以上であるかをチェックし、達していない場合は足りない時間を求めます。なお、時間は1時間を1単位とし、分や秒は考えないものとします。時刻は24時間表記で1時間単位で表します。

また、勉強できる時間帯は6時から22時までで、それぞれ行った勉強時間は重複しません。

1日の勉強の目標時間と、実際に勉強した時間の情報（勉強の回数 n 、それぞれの勉強の開始時刻 s と終了時刻 f ）を入力とし、勉強時間の合計が目標に達しているかを調べ、達していれば”OK”（半角英字）を、達していない場合は足りない時間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10以下とし、 s 、 f は、6以上22以下とします

例：目標時間を勉強した時間の計算

目標時間	勉強した時間	
10 時間	6 時~11 時 = 5 時間 12 時~15 時 = 3 時間 18 時~22 時 = 4 時間	OK!!
14 時間	6 時~11 時 = 5 時間 13 時~20 時 = 7 時間	2 時間不足

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 1 日の目標時間 t (整数)

2 行目 勉強の回数 n (整数)

3 行目 第 1 の勉強の開始時刻 s 終了時刻 f (整数 整数 ; 半角空白区切り)

:

$n+2$ 行目 第 n の勉強の開始時刻 s 終了時刻 f

出力

入力データセットごとに、OK または足りない時間を出力します。

入力例

```
10
3
6 11
12 15
18 22
14
2
6 11
13 20
0
```

出力例

```
OK
2
```

問題

天高く馬肥ゆる秋。そろそろ食べ物が美味しくなる季節です。高校生のよしお君も、テレビや読書のお供にいろいろお菓子里に手が伸びてしまいます。そのせいか最近体重は増える一方で、健康的にもよろしくありません。しかしどうしても直ぐに間食がやめられないよしお君は、お菓子里の栄養素やカロリーが制限された場合に、食べたいお菓子里を、食べてもよいお菓子里と食べてはいけないお菓子里に分類することにしました。



食べ物には3大栄養素と呼ばれる「たんぱく質」「脂質」「炭水化物」の3つの栄養素が含まれています。1g(グラム)あたりたんぱく質と炭水化物は4kcal(キロカロリー)、脂質は9kcalと計算されます。たとえば、下の表によると番号1のケーキにはたんぱく質7g、脂質14g、炭水化物47gが含まれています。これをもとに含まれるカロリーを計算すると、 $4 \times 7 + 9 \times 14 + 4 \times 47 = 342$ kcalとなります。その他も同様に計算されています。

番号	名前	たんぱく質(g)	脂質(g)	炭水化物(g)	カロリー(kcal)
1	ケーキ	7	14	47	342
2	ポテトチップス	5	35	55	555
3	どら焼き	6	3	59	287
4	プリン	6	5	15	129

分類の対象となるお菓子里の個数 n と各お菓子里の情報、制限の情報を入力とし、そのお菓子里一つだけなら制限を超えることがない(食べてもよい)お菓子里の一覧を出力するプログラムを作成してください。

お菓子里の情報は、お菓子里の番号 i 、そのお菓子里に含まれるたんぱく質の重量 p 、脂質の重量 q 、炭水化物の重量 r から成り立っています。制限の情報には、含むことができる最大のたんぱく質の重量 P 、脂質の重量 Q 、炭水化物の重量 R 、および摂取することができる最大のカロリー C からなっており、たんぱく質、脂質、炭水化物、カロリーのいずれかの一つでも超えると制限違反となり、「食べてはいけないお菓子里」と判断されます。

お菓子里の個数 n は1以上1000以下の整数、お菓子里の番号 i は1以上1000以下の整数、各栄養素の重量 p, q, r は0以上100以下の整数で与えられます。各栄養素の制限値 P, Q, R は0以上100以下の整数、カロリー C は0以上1700以下の整数とします。食べてもよいお菓子里の一覧は、食べてもよいお菓子里の番号を入力順に出力してください。なお、食べてもよいお菓子里がなければ"NA"と出力してください。

上の表にある4個のお菓子里について、 $P=10, Q=15, R=50, C=400$ という制限の場合、ケーキとプリンはそれぞれの栄養素及びカロリーが制限値未満なので食べてもよいお菓子里に分類されますが、ポテトチップスは炭水化物とカロリー、どら焼きは炭水化物の量が制限値を超えているため食べてはいけないお菓子里に分類されます。

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 お菓子の数 n (整数)

2 行目 第 1 のお菓子の情報 $i p q r$ (整数 整数 整数 整数 ; 半角空白区切り)

3 行目 第 2 のお菓子の情報

:

$n+1$ 番目 第 n のお菓子の情報

$n+2$ 番目 各栄養素及びカロリーの制限値 $P Q R C$ (整数 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、食べてもよいお菓子の番号または"NA"を出力します。

入力例

```
4
1 7 14 47
2 5 35 55
3 6 3 59
4 6 5 15
10 15 50 400
2
1 8 10 78
2 4 18 33
10 10 50 300
0
```

出力例

```
1
4
NA
```

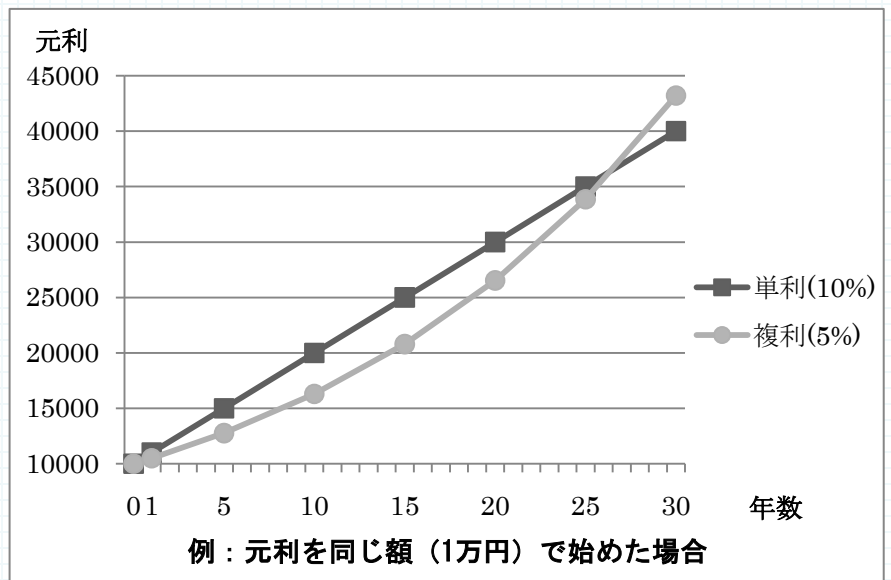
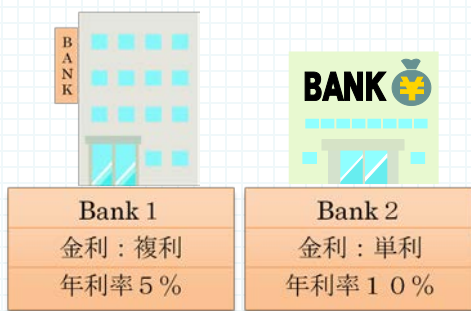
問題

金利は銀行に預けているお金に付き、その計算方法や利率は銀行によって様々です。利息と元金を合わせたものを元利と呼びますが、元利の計算方法として、利息を元金に組み入れずに計算する「単利」と利息を元金に組み入れて計算する「複利」というものがあり、より多くの元利を得るためにはこの差異を理解していなければなりません。

元利の計算方法は以下ようになります。

➤ 単利: $元利 = 元金 \times \left(1 + 年数 \times \frac{年利率}{100}\right)$

➤ 複利: $元利 = 元金 \times \left(1 + \frac{年利率}{100}\right)^{年数}$



できれば、わずかな金利でもより多くお金を増やしたいと思うものです。相沢さんの家庭では、今まで預けていた銀行は金利が低い事に気づき、新しい銀行に乗り換えようとしています。しかし、相沢さん一家は金利のことについてあまり詳しく知らなかったので、金利に詳しいあなたに助けを求めてきました。比較すべき銀行の数は多いうえ、相沢さん一家以外にも助けを求められそうです。作業を効率化するためにより良い銀行を計算するプログラムを作成しておきましょう。

銀行の数 n 、お金を預ける年数 y 、各銀行の情報（銀行番号 b 、金利の種類 t 、年利率 r パーセント）を入力とし、最も元利が高くなる銀行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 、 b はそれぞれ 1 以上 50 以下の整数、 y は 1 以上 30 以下の整数、 r は 1 以上 100 以下の整数とします。また、金利の種類 t は、単利の場合は 1、複利の場合は 2 で与えられます。また、最も元利が高くなる銀行は一つだけです。

入 力

複数のデータセットが入力として与えられます。入力の終わりはゼロひとつで示されます。各データセットは以下の通りです。

1 行目 n (整数)

2 行目 y (整数)

3 行目 第 1 の銀行の情報 $b r t$ (すべて整数 ; 半角空白区切り)

4 行目 第 2 の銀行の情報

:

$n+2$ 行目 第 n の銀行の情報

出 力

入力データセットごとに、最も元利が高くなる銀行番号を出力します。

入力例

```
2
8
1 5 2
2 6 1
2
9
1 5 2
2 6 1
0
```

出力例

```
2
1
```

問題

複素数を拡張したものに四元数と呼ばれるものがあります。物体の回転などを表現するのに便利なので、ロボットの腕の制御などに使える便利な数です。四元数は、4つの実数 x 、 y 、 z 、 w と特別な数（虚数を拡張したもの） i 、 j 、 k を用いて、 $x + yi + zj + wk$ と表されます。このような四元数の和は次のように定義されます。

$$(x_1 + y_1i + z_1j + w_1k) + (x_2 + y_2i + z_2j + w_2k) = (x_1 + x_2) + (y_1 + y_2)i + (z_1 + z_2)j + (w_1 + w_2)k$$

一方、 1 、 i 、 j 、 k の間の積は次のように与えます。

		B			
		1	i	j	k
A	1	1	i	j	k
	i	i	-1	k	$-j$
	j	j	$-k$	-1	i
	k	k	j	$-i$	-1

この表は、2つの特別な数AとBの積ABを表しています。例えば、 i と j の積 ij は k であり、 j と i の積 ji は $-k$ であることを表しています。

一般の四元数の積は、この関係を満たすように計算されます。例えば、2つの四元数、 $1 + 2i + 3j + 4k$ と $7 + 6i + 7j + 8k$ の積は次のように計算します。

$$\begin{aligned} (1 + 2i + 3j + 4k) \times (7 + 6i + 7j + 8k) &= 7 + 6i + 7j + 8k \\ &\quad + 14i + 12i^2 + 14ij + 16ik \\ &\quad + 21j + 18ji + 21j^2 + 24jk \\ &\quad + 28k + 24ki + 28kj + 32k^2 \end{aligned}$$

となり、上の表を適用することにより

$$= -58 + 16i + 36j + 32k$$

となります。

4つの係数 x 、 y 、 z 、 w が整数でかつすべてがゼロとはならない2つの四元数 $(x_1 + y_1i + z_1j + w_1k)$ と $(x_2 + y_2i + z_2j + w_2k)$ を入力とし、その積 $(x_3 + y_3i + z_3j + w_3k)$ をとする時の x_3 、 y_3 、 z_3 、 w_3 を出力するプログラムを作成してください。ただし、与えられる係数はすべて-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 四元数の組の個数 n (整数)

2 行目 第1の四元数の組の情報 $x_1 y_1 z_1 w_1 x_2 y_2 z_2 w_2$ (すべて整数; 半角空白区切り)

3 行目 第2の四元数の組の情報

:

$n + 1$ 行目 第 n の四元数の組の情報

出力

入力データセットごとに、四元数の積を出力します。

入力例

```
2
1 2 3 4 7 6 7 8
5 6 7 8 3 2 3 4
0
```

出力例

```
-58 16 36 32
-50 32 28 48
```

問題

携帯電話には、メールなどの文章を効率良く入力するために入力候補を表示する機能が搭載されています。これは、使われる頻度が高い単語を記録しておき、入力された文字を頭文字に持つ単語を入力候補として提示するものです。例えば、普段”computer”という単語を多く入力しているなら、”c”と入力するだけで候補として”computer”が提示されます。

このような機能を自作するために、その基本部分として、記録されている文章の中から、指定された文字を頭文字に持つ単語を出現回数が多い順に出力するプログラムを作成しましょう。

記録された文章、文字 k (半角英字) を入力とし、その文字 k を頭文字に持つ単語を出現数の順に出力するプログラムを作成してください。ただし、記録された文章は半角英小文字と半角空白からなり、文字数は1文字以上1024文字以下とします。空白によって区切られる単語は1文字以上の20文字以下であり、また空白が2つ以上連続して入力されることはありません。出現数が同じ単語が複数ある場合は、アルファベット順に、より若い文字が先にある単語から出力します。出力する単語は最大5つまでとします。該当する単語が存在しない場合は”NA”と出力してください。



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で表されます。各データセットは以下のとおりです。

- 1 行目 文章（半角英小文字及び半角空白の列）
- 2 行目 頭文字 k（半角英字）

出力

入力データセットごとに、指定された文字を頭文字にもつ単語または”NA”と出力します。

入力例

```
ben likes bananas the best among many fruits because bananas are sweet and cheap
b
winners get to visit aizu and the university of aizu and make many friends as well
a
ask alex about the answer for the assignment on android apps
a
programming is both a sport and an intellectual puzzle
c
0
```

出力例

```
banana because ben best
aizu and as
about alex android answer apps
NA
```

問題

タブレット型の端末の流行は留まるところを知らず、携帯電話もその影響を受けたりしながら、私たちの生活のありとあらゆる所に浸透してきています。なぜここまで人々に受け入れられているのか。理由としては、指で画面にタッチして操作できる、直感的なインターフェースはその一端を担っていると言えるでしょう。

もちろんゲームの分野でもこの技術は利用され、新しい操作感を持つ様々な種類のゲームが作られています。現在、AZ社が開発しているゲームもその一つです。AZ社のプログラマーであるあなたは、このゲームのロジックの一部分を開発することになりました。

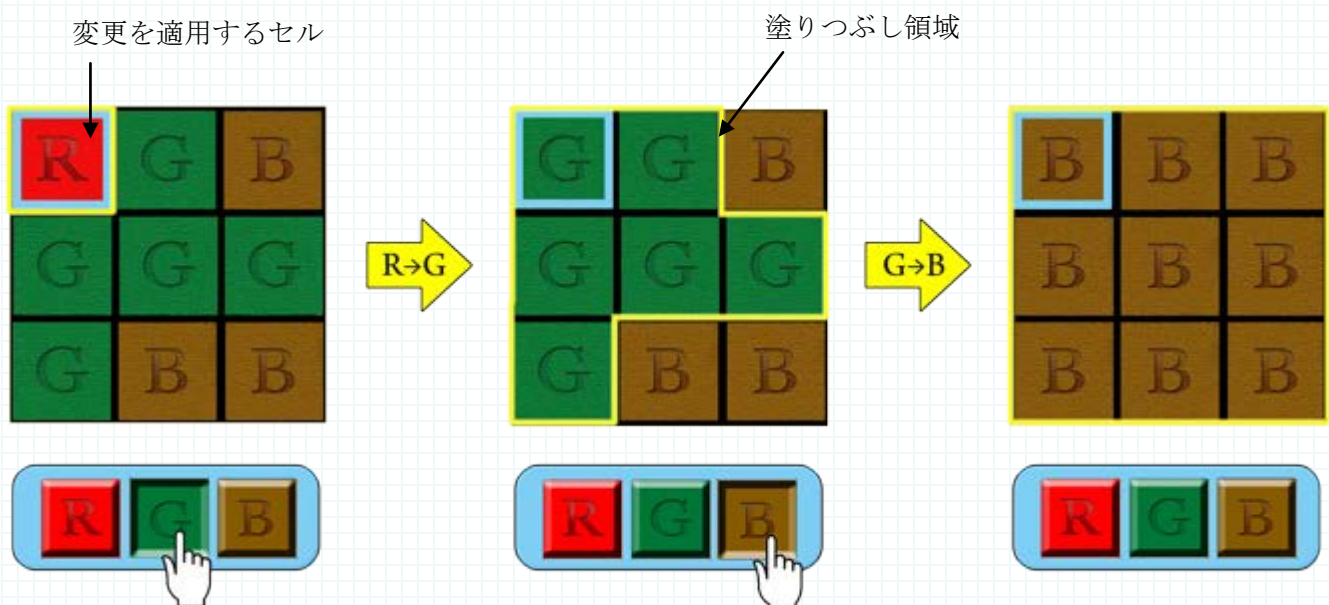
このソフトウェア（ゲーム）の要件は以下の通りです。

- 画面には x 列 \times y 行の 2 次元グリッドがあります。
- グリッドのセルは赤(R)、緑(G)、茶(B)の 3 色のいずれかで塗られています。
- セルの色を変更するボタン R、G、B があり、このボタンが押下されると一番左上(0,0)のセルがその色に変更されます。
- 一番左上のセルの色が変更されると、そのセルの元の色と同じ色に塗られた隣接するすべてのセルが指定された色に変わります。すなわち、元の色が同じ色のセルでつながったセルはすべて指定された色に変更されます。

このようにグリッドを塗りつぶしていき、最終的にグリッド全体を一色にするというゲームです。少ない手順で解くと高得点が与えられます。

まずは、この最高得点を計算するため、このゲームを最短で解いた場合、最低何回選択ボタンを操作すればグリッドが一色になるかをもとめるプログラムを作成することにしました。

2次元のグリッドのサイズ $x \times y$ と各セルの初期状態の色 (R、G、B のいずれか) を入力とし、すべてのセルを同一色にするために必要な選択ボタンの操作回数の最小値を出力するプログラムを作成してください。ただし、 x と y は 2 以上 10 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で示されます。

1 行目 x y (整数 整数 ; 半角空白区切り)

2 行目 1 行目のセルの色 s_1 \dots s_x (すべて半角英字 ; 半角空白区切り)

:

$y+1$ 行目 y 行目のセルの色

出力

入力データセットごとに、ボタン操作の最少回数を出力します。

入力例

```
3 3
R G B
G G G
G B B
2 4
R G
G G
R B
B R
4 3
G G B R
G R R G
B G G R
0 0
```

出力例

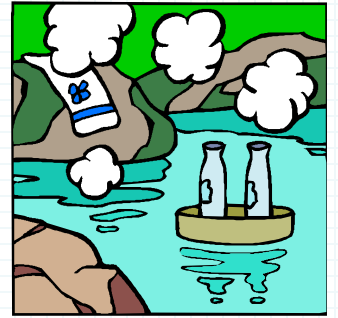
```
2
4
4
```

問題

温泉好きのたけしさんは、次の長期休暇を利用してとある温泉地への旅行を計画しています。移動は長距離バスを乗り継ぎ、なるべくお金をかけずに目的地へたどり着きたいと思っています。

貯金があるとはいえ、資金に心許ないたけしさんは、おじいさんに相談することにしました。計画を聞いて感心したおじいさんは、たけしさんに特別な切符を渡しました。

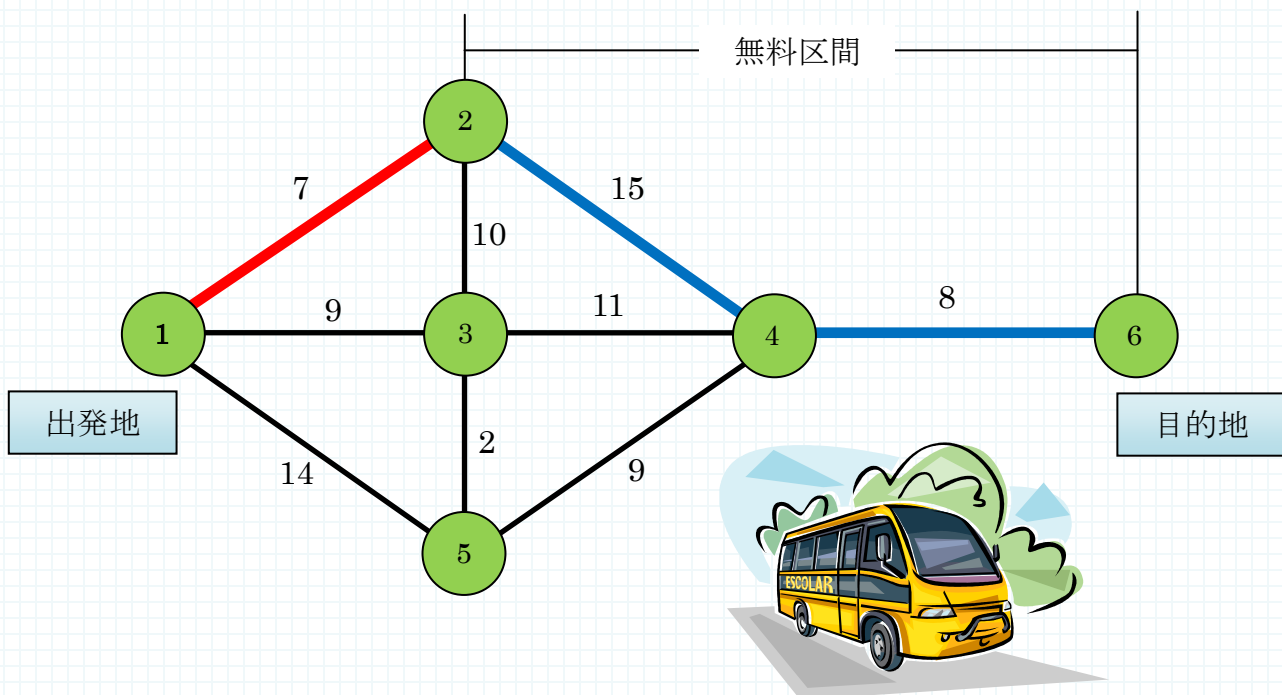
その切符は、長距離バスの連続した2区間を1回だけ無料で乗れるというものでした。使いようによってはかなりの移動費削減が見込めますが、より大きな効果を発揮させるためにはしっかりした計画を練る必要があります。



出発地と目的地、及び中継地点が合わせて n 個、2つの地点を結ぶ路線が m 個与えられます。各地点にはそれぞれ1から n までの数字が割り振られています。出発地は1、目的地は n です。路線の情報は、その路線が結ぶ2つの地点 a と b 、及びその料金 c で表されます。

おじいさんからもらった特別な切符の効力により、任意の地点から、一度だけ連続した2つの路線を料金0で通過することができます。ただし、途中で目的地を通過しても、目的地にたどり着いたことにはなりません。

出発地、目的地、及び中継地点の総数 n と路線の数 m 、各路線の情報を入力とし、料金の最小値を出力するプログラムを作成してください。ただし、必ず出発地から目的地へと到達する経路が存在するものとします。また n は2以上100以下、 c は1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通りです。

1 行目 n m (整数 整数; 半角空白区切り)

2 行目 第 1 の路線の情報 a_1 b_1 c_1 (整数 整数 整数; 半角空白区切り)

:

$m+1$ 行目 第 m の路線の情報 a_m b_m c_m

出力

入力データセットごとに、料金の最小値を出力します。

入力例

```
2 1
1 2 5
3 2
1 2 5
2 3 5
6 9
1 2 7
1 3 9
1 5 14
2 3 10
2 4 15
3 4 11
3 5 2
4 5 9
4 6 8
0 0
```

出力例

```
5
0
7
```

問題

より良い物を、より安く。今日もどこかのスーパーマーケットで行われるタイムセールでは、激しい闘いが繰り広げられています。

ここ会津にある「LL 堂」もそんなスーパーマーケットのひとつで、他の全国チェーン店と対抗すべく、少し変わったタイムセールを実施しています。一般的なタイムセールでは複数の商品が同じ時間に安くなるものですが、LL 堂では対象となる商品によって、タイムセールを開始する時間が別々に設定されているのです。こうすることによって他店との差別化を図っています。

坂井家は LL 堂をよく利用する家庭の一つです。そんな坂井家では、奥様主導のもと、次の日曜に行われるタイムセールに向けて作戦会議を開き、どのような順番で買い物をすれば値引きが最大となるかを分析することになりました。店内を熟知している坂井家は売り場の見取り図を書き、また欲しい品物がどこにあるか、値引きは幾らであるか、売り切れるまでの時間を把握することに成功しました。

ここまでは完璧だった坂井家ですが、分析を行える人がいませんでした。そこで坂井家と親交があったあなたはプログラムを書くことにしました。一人の人が動くことを想定してシミュレーションを行います。

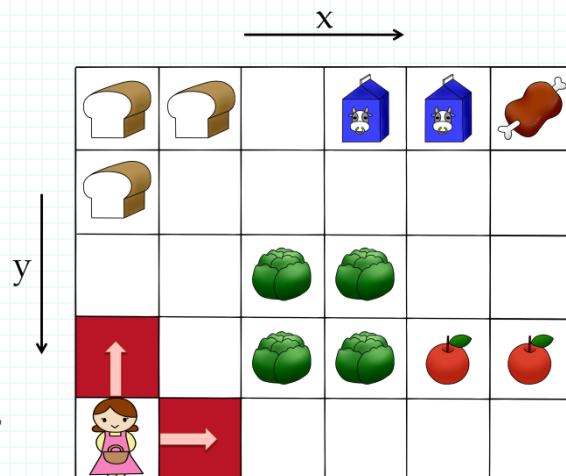
商品は 10 種類あり、数字 1 桁の商品番号 g によって区別されます。タイムセール情報には商品番号 g 、値引き額 d 、タイムセールの開始時刻 s と売り切れ時刻 e があります。

店内は横 x 、縦 y のマスで構成される 2 次元グリッドで表され、マスごとに通路、商品棚のどちらかが割り当てられています。一つの商品棚には 1 種類の商品があり、それは商品番号 g で区別されます。どの商品を買ってもよいですが、同じ商品番号の商品を複数買ってはいけません。商品棚からは、上下左右に隣接する通路のマスならどんな向きでも商品を取れます。

タイムセールが始まる時刻から商品を取ることができますが、売り切れる時刻からは商品を取ることができません。また、時間は入店した時点で 0 から始まります。

移動は現在いるマスから上下左右の隣接する通路マスに移動することができますが、商品棚のマスに移動することはできません。グリッドで表される店の外に出ることもできません。1 回移動する毎に 1 単位時間経過します。また、商品を取る時間は考えないものとします。

店内の見取り図と買い物をする人の初期位置と商品のタイムセール情報を入力とし、取ることのできた商品の値引き額の合計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、店舗の大きさ x, y は 3 以上 20 以下、タイムセール情報の数 n は 1 以上 8 以下、タイムセールが始まる時刻 s 、商品が売り切れる時刻 e はともに 0 以上 100 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通りです。

1行目 店舗の大きさ x y (整数 整数; 半角空白区切り)

2行目 1行目の店内情報 m_1 m_2 \dots m_x (すべて半角文字; 半角空白区切り)

各記号の値と意味は次の通りです。

. (ピリオド) : 通路のマス

数字 : 商品の番号

P : 買い物をする人の初期位置のマス

3行目 2行目の店内情報

:

$y+1$ 行目 y 行目の店内情報

$y+2$ 行目 タイムセール情報の数 n (整数)

$y+3$ 行目 第1のタイムセール情報 g d s e (すべて整数; 半角空白区切り)

$y+4$ 行目 第2のタイムセール情報

:

$y+n+2$ 行目 第 n のタイムセール情報

出力

入力データセットごとに、取ることのできる商品の値引き額合計の最大値を出力します。

入力例

```
6 5
1 1 . 0 0 4
1 . . . . .
. . 2 2 . .
. . 2 2 3 3
P . . . . .
5
0 50 5 10
1 20 0 10
2 10 5 15
3 150 3 5
4 100 8 9
0 0
```

出力例

```
180
```


問題

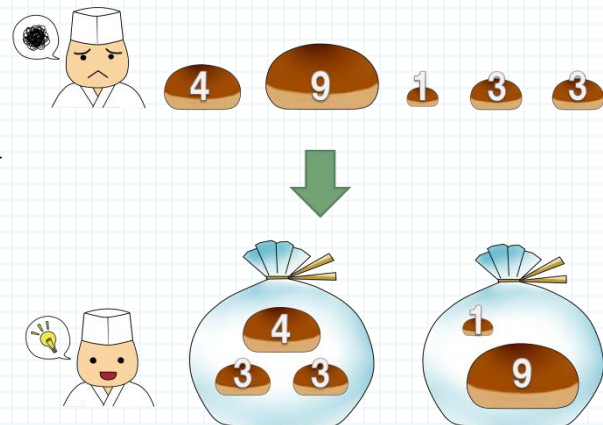
会津若松市にある和菓子屋・友栗堂の店長さんは、とても腕のいい職人さんなのにちょっと気分屋なのが玉にキズです。店長さんの作る饅頭はとても美味しいのですが、その時の気分によって大きさがまちまちになってしまいます。時には普通の饅頭の何倍もの大きさで作ってしまうこともあるほどです。大きさや重さをいちいち量り、一つ一つ違う値段をつけるのはお客さんにとっても店にとっても不便なので困りものでした。

見かねた店長さんの奥さんは、この大きさと重さがばらばらな饅頭を袋に詰めて売り出すことを思いつきました。一定の重さになるように饅頭を袋に詰めて売れば、一つ一つの大きさはばらばらでも一定の値段で売り出すことが出来ますし、どんな饅頭が入っているのか開けるまでわからないというサプライズも売りになるかもしれません。

「ばらばら饅頭」という名前で売りだしたところ、新しいアイデアが話題となり友栗堂はたちまち評判となりました。しかし、問題もあり、袋に入れる饅頭の組み合わせ方によって作れる「ばらばら饅頭」の数が変わり、無駄が発生してしまうことがありました。アルバイトで友栗堂に来ていたあなたは無駄なく袋に詰めるためにプログラムをつくることにしました。

饅頭の重さは1から9の9種類あります。

袋に詰める時には、饅頭の重さの合計がぴったり10になるように詰め込みます。饅頭の組み合わせ方は $1+1+2+4+2=10$ や $1+1+1+1+1+5=10$ など、同じ重さの饅頭をいくつ使っても構いません。



作られる饅頭の個数 n と饅頭の重さの情報を入力とし、作ることのできる「ばらばら饅頭」の最大の数を出力と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で示されます。各データセットは以下の通りです。

1 行目 饅頭の個数 n (整数; 半角空白区切り)

2 行目 各饅頭の重さ $m_1 \cdots m_n$ (すべて整数; 半角空白区切り)

出 力

入力データセットごとに、詰め込むことのできた「ばらばら饅頭」の最大の数を出力します。

入力例

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

出力例

```
1
4
4
```

問題

四角のマス縦横にならべた長方形の迷路があります。この迷路では東西南北の隣接するマスへ移動しながら、スタートのマス S を出発し、ゴールのマス G を目指します。マスの種類には、平原、山、氷の3種類があります。S と G は、平原のマスに置かれています。平原のマスには移動できますが、山のマスには移動できません。氷のマスは移動できますが、条件によって氷が割れ動けなくなります。

- 氷のマスは東西南北に隣接するもの全体でひとつの塊として扱われます。
- 氷の塊のなかで、マスの個数の半分より多くのマスに移動すると、塊全体が割れます。

例えば、図1は与えられた迷路の最短経路のステップ数で11であることを示しています。

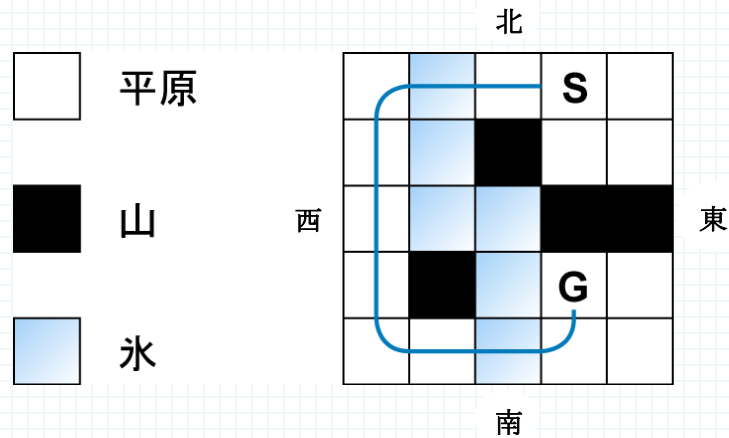


図1

しかし、図2のように氷のマスを通して近道をしようとすると、大きさが6の氷の塊に対する4回目の移動後に動けなくなってしまいGには到達できません。

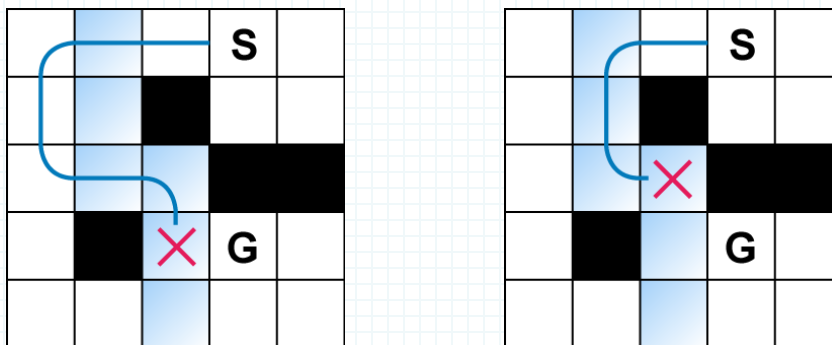


図2

このような迷路の情報を入力とし、SからGまでの最短経路のステップ数を求めるプログラムを作成してください。マスの種類はそれぞれ以下の文字で表されます：

文字 (半角)	マスの種類
.	平原
#	山
X	氷

与えられる迷路は必ず解けるものとします。迷路はその東西方向のマスの数 x 、南北方向のマスの数 y と $x \times y$ 個の文字として与えられます。 x 、 y はともに2以上12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 x y (整数 整数 ; 半角空白区切り)
- 2行目 迷路の1行目の情報 (長さ x の半角英文字列)
- :
- $y+1$ 行目 迷路の y 行目の情報

出力

入力データセットごとに、最小のステップ数を出力します。

入力例

```
5 5
.X.S.
.X#..
.XX##
.#XG.
..X..
7 3
SXX.XXG
X.#.#X.
XXX.XX#
4 4
S...
X.X.
GX..
...X
10 10
..XXXXX.XX
.X.#.#X.XX
SX.#X.X..X
#X.##.X.XX
..XXXX#.XX
##.##.##XX
....X.XX#X
.##X..#X#X
....XX#..X
...#XXG..X
0 0
```

出力例

```
11
10
10
33
```