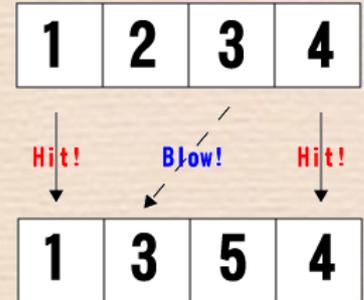


問題

「ヒットアンドブロー」という2人用のゲームがあります。ルールは、以下の通りです。

- ▶ 出題者と回答者に分かれて行う。
- ▶ 出題者は、重複した数を含まない4桁の数字（正解）を決める。
- ▶ 回答者は、その4桁の数字（回答）を言い当てる。
- ▶ 回答に対して、出題者はヒットとブローの数でヒントを与える。
- ▶ 回答と正解を比べて、数と桁位置の両方が同じであることをヒットと呼び、数だけが同じで桁位置が異なることをブローと呼ぶ。たとえば、正解が1234で、回答が1354なら、出題者は「2ヒット、1ブロー」というヒントを与え、正解までこれを繰り返す。
- ▶ 出題者と回答者は交代してゲームを行い、より少ない回答で正解を言い当てた方を勝ちとする。



太郎君と花子さんはこのヒットアンドブローで遊ぶこととしました。しかし二人は、ヒットの数とブローの数をその都度判断することが少し面倒に感じているようです。そんな二人のために、あなたはヒットの数とブローの数が即座に分かるプログラムを作成してあげましょう。

正解 r と回答 a を入力とし、ヒットの数とブローの数を出力するプログラムを作成してください。 r 、 a はそれぞれ 0 - 9 の数字 4 つからなる数字の列です。

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 $r a$ (半角数字列 半角数字列 ; 半角空白区切り)

出力

入力データセットごとに、ヒットの数とブローの数を出力します。

入力例

```
1234 5678  
1234 1354  
1234 1234  
1230 1023  
0123 1234  
0 0
```

出力例

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

問題

悪天候が続き野菜の価格が高騰する中、セブンマートではお客様に野菜のまとめ買いセールを実施しています。日ごろなかなか店頭には並ばない野菜もお手頃価格で手に入るとあって、店内はとても賑わっています。

ある日、松長団地では仲良し3人組がセブンマートの広告を手話に花を咲かせていました。今回のセールは「お客様大感謝祭」と銘打っただけに、袋詰めした野菜の中で最も安いものが無料になるのが目玉となっています。広告を読んでもみると、どうやら以下のようなセールのようです。

- ▶ 1つの袋には m 個まで野菜を詰められる。
- ▶ 野菜が m 個詰めてある袋については、その中で最も安い野菜が無料となる。
- ▶ 野菜の個数が m 個に達しない袋は割引の対象外。

3人は早速セブンマートへ買い物に行きました。

買い物が終わり、お店の外で待ち合わせた3人は安くたくさん購入できたことに満足した様子で話をしていると、どうやら3人とも同じ野菜を購入していたことが分かりました。ある一人が、「本当に安いわよねえ。これで XXX 円だもの！」と言うと、もう一人は、「え？私はそれより**円高かったわ！どうして？」と驚き、また、残りの一人はレシートを見て自分が一番安く購入したことに気付きました。

さて、どのように袋詰めすれば購入価格を一番安くできるでしょうか。

購入する野菜の個数 n 、袋に入る野菜の個数 m 、各野菜の値段 p を入力とし、最低購入価格を出力するプログラムを作成してください。ただし、 n 、 m は1以上1000以下の整数、 p は10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 n m (整数 整数；半角空白区切り)

2行目 各野菜の値段 p_1 p_2 $p_3 \dots p_n$ (それぞれ整数；半角空白区切り)

出力

入力データセットごとに、最低購入価格を出力します。

入力例

```
4 2
50 40 100 80
7 3
400 300 100 700 200 600 500
0 0
```

出力例

```
150
2100
```

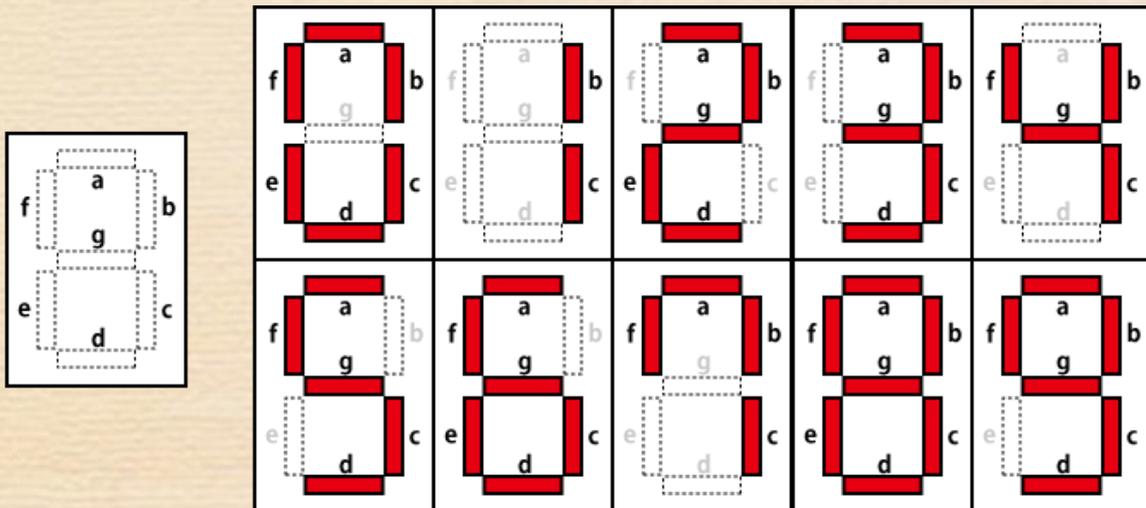
問題

電卓などでよく目にするデジタル数字を表示している画面は、デジタル数字が7つの部分（セグメント）で構成されることから、「7セグメントディスプレイ」と呼ばれています。

ワカマツ社で新しく売り出す予定の製品は、デジタル数字の表示が必須のため、7セグメントディスプレイを製品に組み込むことになり、社員であるあなたは、与えられた数字を7セグメントディスプレイに表示するプログラムを作成することになりました。

この7セグメントディスプレイは、次の切り替えの指示が送られてくるまで表示状態は変わりません。7ビットからなるシグナルを送ることで、それぞれ対応したセグメントの表示情報を切り替える事ができます。ビットとは1か0の値を持つもので、ここでは1が「切り替え」、0が「そのまま」を表します。

各ビットとセグメントの対応関係は下の図のようになっています。シグナルは7つのビットを“gfedcba”の順番になっています。例えば、非表示の状態から「0」を表示するためには“0111111”をシグナルとしてディスプレイに送らなければなりません。「0」から「5」に変更する場合には“1010010”を送ります。続けて「5」を「1」に変更する場合には“1101011”を送ります。



表示したいn個の数字を入力とし、それらの数字を順に7セグメントディスプレイに正しく表示するために必要なシグナル列を出力するプログラムを作成してください。なお、7セグメントディスプレイの初期状態は全て非表示の状態であるものとします。ただし、数字の個数nは1以上100以下の整数であり、各数字は0から9までの数字とします。

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-1 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 数字の個数 n (整数)
2 行目 第 1 の数字 (整数)
3 行目 第 2 の数字
:
 $n+1$ 行目 第 n の数字

出力

入力データセットごとに、数字をディスプレイに正しく出力するために必要なシグナルを出力してください。

入力例

```
3
0
5
1
1
0
-1
```

出力例

```
0111111
1010010
1101011
0111111
```

問題

おおあたり

あいつ学園大学附属高校の大阿足あたる君は友人達と、大阿足君が製作したスロットマシンで遊ぶことになりました。

このマシンは、メダルを投入すると、3つのリールが回転を始め、各リールが自動的に止まります。通常の1ゲーム（通常ゲーム）は3枚のメダルを投入し、図柄が揃うと、その図柄に応じて次のとおりメダルが得られます。

図柄	読み方	メダルの枚数	特別なサービス
7 7 7		15枚	ビッグボーナス開始
BAR BAR BAR		15枚	レギュラーボーナス開始
	(ブドウ)	7枚	ボーナスゲーム中は15枚
	(チェリー)	2枚	
	(スター)	0枚	無料ゲーム開始

図柄の揃い方によっては特別なサービスが開始されます。7の図柄が3つ揃うとビッグボーナスが始まり、ボーナスゲームを5ゲーム行うことができます。また、BARの図柄が3つ揃うとレギュラーボーナスが始まり、ボーナスゲームを3ゲーム行うことができます。

スターの図柄が3つ揃うと無料ゲームが開始され、メダルを得ることはできませんが、次のゲームをメダルの投入なく始めることができます。

ボーナスゲーム中は1ゲームあたり2枚のメダルを投入すると、自動でブドウの図柄が3つ揃い、メダルを15枚得ることができます。

大阿足君とその友人たちは、それぞれ100枚のメダルを持ってマシンで遊び始めました。しばらく遊び、通常ゲームになった状態で終了しました。手元に残ったメダルは何枚となったのでしょうか。

プレイ情報を入力とし、手元に残ったメダルの数を出力するプログラムを作成して下さい。

プレイ情報とは、ビッグボーナスの回数 b 、レギュラーボーナスの回数 r 、通常ゲーム中にブドウが揃った回数 g 、チェリーが揃った回数 c 、スターが揃った回数 s 、総ゲーム数 t です。

なお、 t にはボーナスゲームの回数を含みます。また、メダルはゲームの途中になくなることはありません。ただし、 b 、 r 、 g 、 c 、 s はそれぞれ0以上200以下の整数、 t は1000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む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 プレイ情報 b r g c s t (整数 整数 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、手元に残ったメダルの枚数を出力します。

入力例

```
3 2 30 3 26 226
9 0 18 3 20 118
5 5 12 2 15 203
7 4 19 2 22 197
7 4 24 4 17 209
0 0 0 0 0 0
```

出力例

```
127
793
414
629
617
```

問題

忍者のあつしさんは、毎日朝早くから夜遅くまで忍者ビルの屋上から町を警備しています。この忍者ビルは、隣接する2つの同じ階数のビルであり、あつしさんは警備のために、ビルとビルの間をジャンプしながら屋上へ向かうことを日課としています。

この2つのビルはよく外壁や窓の清掃を行うため、ビル登りの助けとなるはしごや障害となる滑りやすい部分があります。しかもはしごや滑りやすい部分の位置は毎日変わります。そのためあつしさんは、屋上へ向かう方法を毎日考えなければいけません。あつしさんは、何回のジャンプで屋上にたどり着けるでしょうか？

あつしさんは二つ並んだ同じ階数のビルの壁を跳び移りながら、ビルの屋上を目指します。ジャンプはどちらか一方のビルの1階から始められます。向かい側のビルへジャンプするときには、同じ階・1つ上の階・2つ上の階の、いずれかに飛び移ることができます。

壁には以下の3種類があり、それぞれの壁にジャンプした後の移動が決まっています。

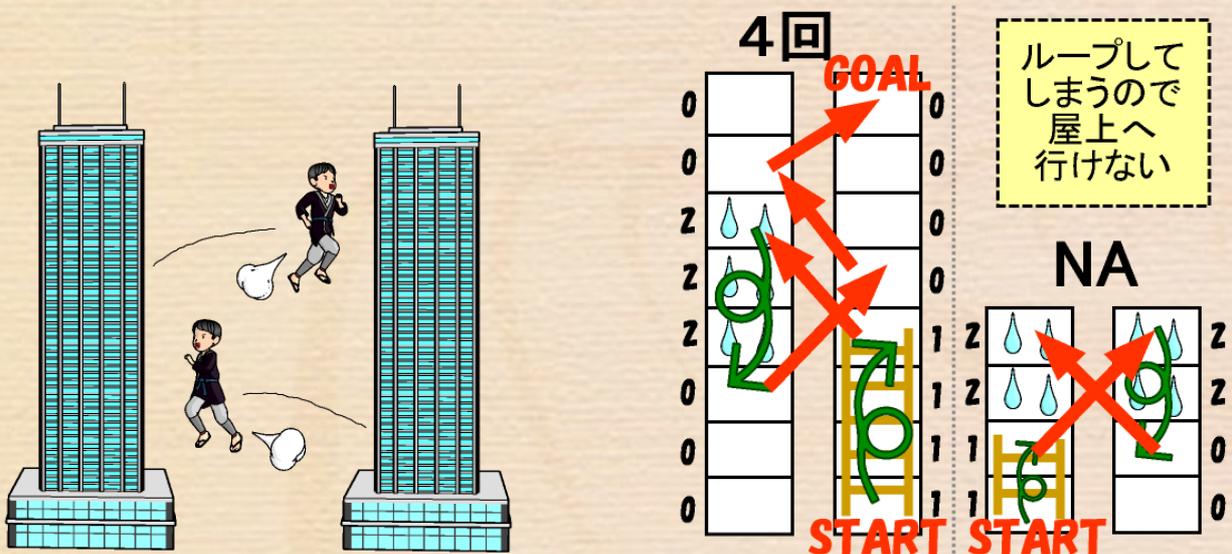
0. 普通の壁：上下の移動はしない。次のジャンプはそこから行う。

1. はしご：はしごは2つ以上の階にまたがってかかっており、今いるはしごの一番上まで移動する。次のジャンプはそこから行う。

2. すべる壁：普通の壁かはしごの一番上まで滑り落ちる。次のジャンプはそこから行う。

また、壁は1階から屋上のすぐ下の最上階まであり、屋上へはそのビルの最上階からのみ行くことができます。また、ビルの最下階の壁はすべる壁にはなりません。

2つのビルの階数 n と2つのビルの壁の種類を入力とし、最少で何回目のジャンプで最上階までたどり着き、屋上まで行くことができるかを出力するプログラムを作成してください。なお、どちらのビルの屋上にたどり着いてもよいものとします。ただし、 n は3以上100以下の整数とします。また、あつしさんがどちらのビルの屋上へもたどり着けない場合は“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 ビルの階数 n (整数)

2行目 1つ目のビル a の1階から n 階までの壁の情報 $a_1 a_2 \dots a_n$ (すべて整数; 半角空白区切り)
各 a_i は、 i 階目の壁の情報を表し、意味は以下のとおりです。

0: 普通の壁

1: はしご (i 階と $i+1$ 階にまたがる)

2: ずべる壁

3行目 2つ目のビル b の1階から n 階までの壁の情報 $b_1 b_2 \dots b_n$ (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、ジャンプの回数を出力します。

入力例

```
8
0 0 0 2 2 2 0 0
1 1 1 1 0 0 0 0
4
1 1 2 2
0 0 2 2
0
```

出力例

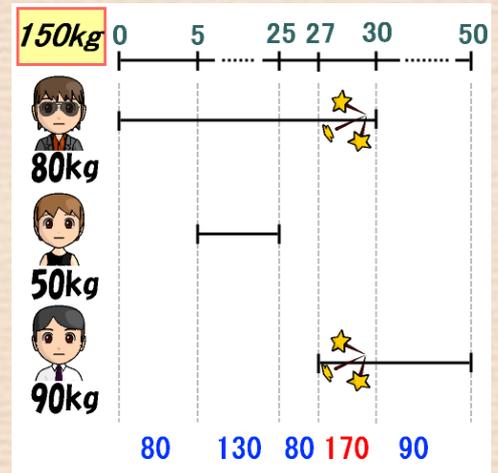
```
4
NA
```

問題

会津若松市よりずっと北に位置する会津古松村では、「ヤ橋（やばし）」と呼ばれる橋が周辺の村へ移動する唯一の手段となっています。その橋は通行人が多いにも関わらず、今にも壊れそうなほど老朽化が進んでしまっています。

ヤ橋は 150kg まで耐えられる強度を持っています。例えば、80 kg の人と 50 kg の人が同時に渡ることはできますが、80 kg の人が渡っているときに 90 kg の人が渡り始めると、ヤ橋は壊れてしまいます。

もしヤ橋が壊れてしまったら、会津古松村の人々は周辺の村へ移動する手段を失ってしまいます。そこで、村で唯一のプログラマーであるあなたは、村民の生活を守るために村長から、橋の渡り方から橋が壊れるか否かを判定するように頼まれました。



橋を渡る通行人の人数 n 、各通行人の体重 m 、橋を渡り始める時刻 a 、渡り終える時刻 b を入力とし、橋が壊れない場合は“OK”と出力し、壊れる場合は“NG”と出力するプログラムを作成してください。なお、橋の上にいる通行人の体重の総和が 150kg を超えてしまうとヤ橋は壊れてしまいます。また、 a の時点で通行人は橋の上にはいますが、 b の時点では橋の上にはいません。ただし、 n 、 m は 1 以上 100 以下の整数、 a 、 b は 0 以上 2^{31} 未満の整数とします。

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 n (整数; 半角空白区切り)
 2 行目 第 1 の通行人の情報 $m a b$ (整数 整数 整数; 半角空白区切り)
 3 行目 第 2 の通行人の情報
 ⋮
 $n+1$ 行目 第 n の通行人の情報

出力

入力データセットごとに、橋が壊れるか否かを出力します。

入力例

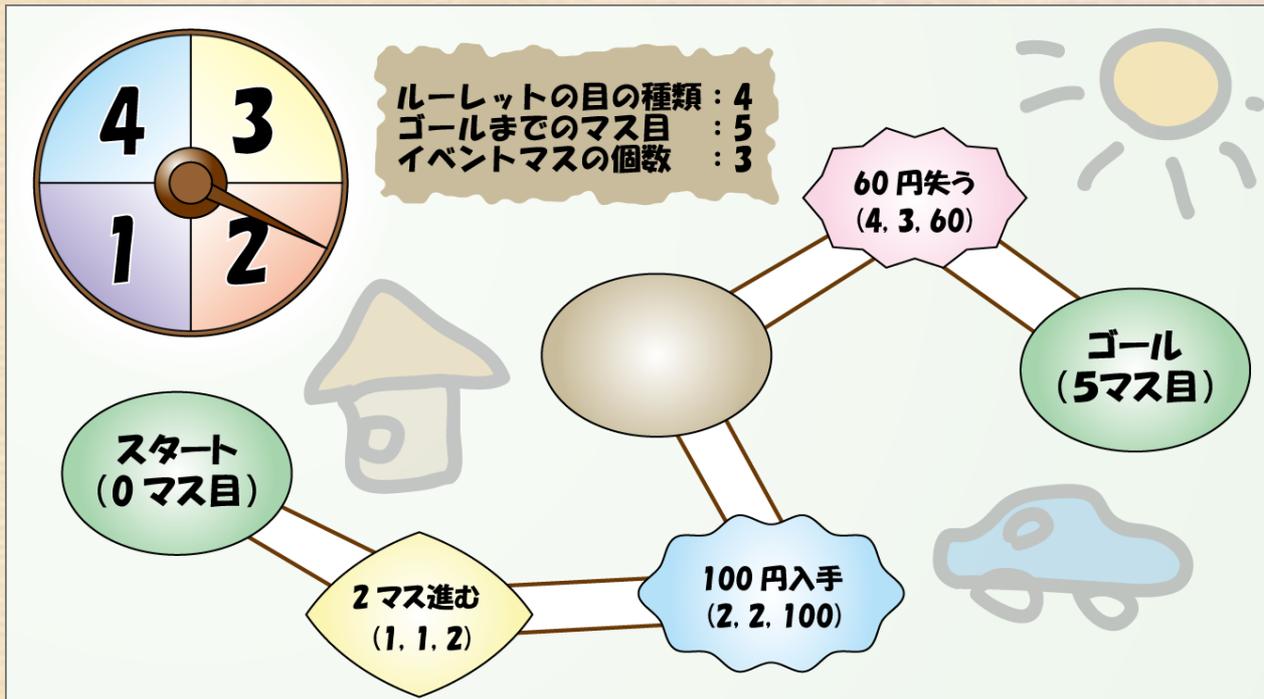
```
3
80 0 30
50 5 25
90 27 50
3
80 0 30
70 5 25
71 30 50
0
```

出力例

```
NG
OK
```

問題

すごろくが大好きな太郎君は、おもちゃ屋さんに会津ホビー社製の人生ゲームを買いに行きました。人生ゲームは、マス目の書かれたボードとルーレットを使って遊びます。ボードには図のようにスタート地点とゴール地点が一つずつあり、それらはひとつつながりのマス目につながっています。最初に、コマはスタート地点のマスに置かれ、ルーレットを回して出た数によってコマを進めます。マスによっては、そこに止まったり通過したりすることでお金を得たりコマの位置を変えたりするイベントマスがあります。最終的な勝敗は、コマがゴール地点に到達した時点の所持金の多寡で決まります。



この会社の人生ゲームの面白いところは、ルーレットの出る目の大きさ、ゴールまでのマス目、イベントマスの配置がひとつひとつパッケージごとに異なることです。それらはケースに書かれており、それを読むことで確認することができます。お金を最も得られる人生ゲームを選びたい太郎君は、得るお金の期待値が一番大きいものを買いたがっています。そこで、友達であるあなたは、太郎君のゲーム選びを手伝うことにしました。

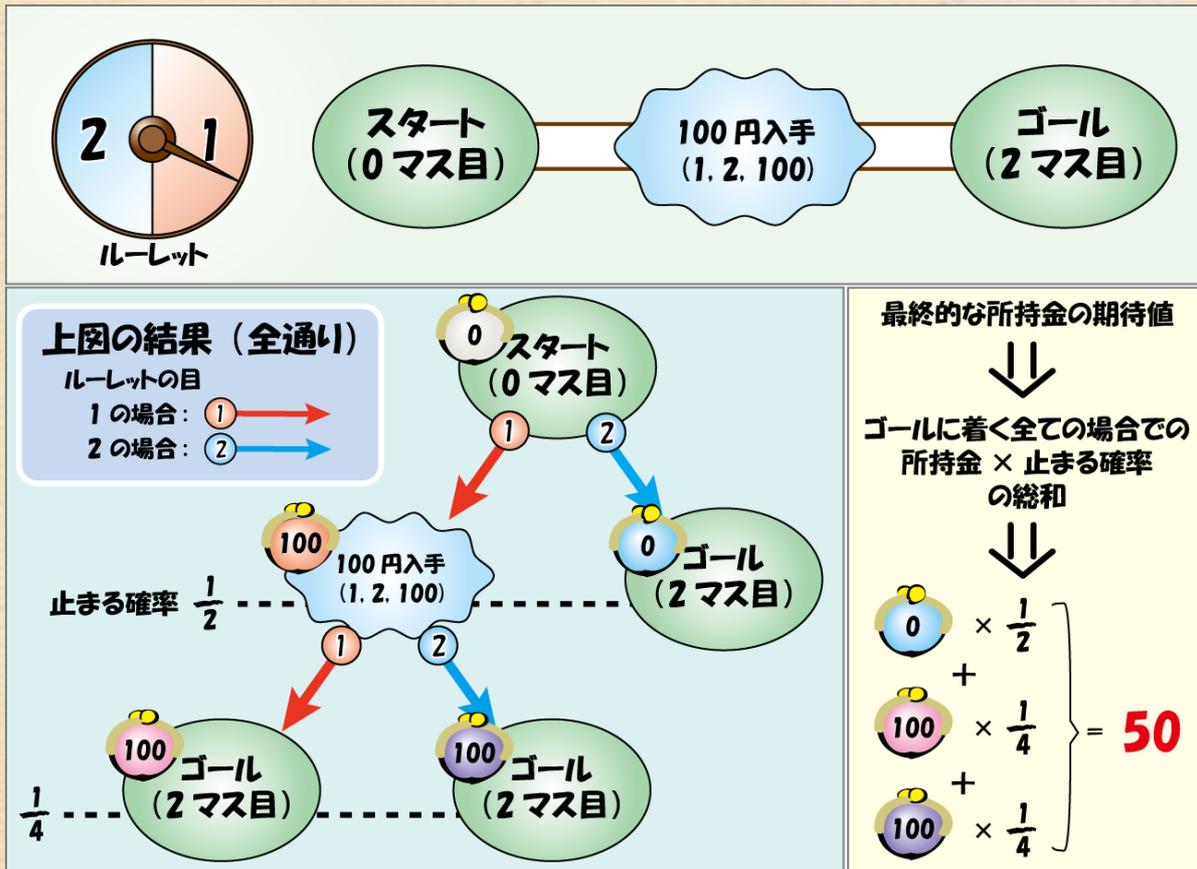
ルーレットは、円周を X 等分に区分され、それぞれに V_1, V_2, \dots, V_x という値が記入されているとします。ボードには、0番、1番、 \dots 、 Y 番と番号がふられたマス目があり、順番につながっています。マス目の中には、イベントマスと呼ばれる特別なマスが Z 個あり、そこに到達すると特別な動作を行います。イベントマスのマス目の番号は N で与えられます。イベントマスには 1~3 の種類があり、それぞれ次の動作が行われます。

種類	特別動作	値の範囲
1	指定の数値だけ先へ進む	1~10
2	指定の数値の金額を得る	1~100
3	指定の数値の金額を支払う	1~100

→次ページへつづきます

最初の所持金は0円で、第0マス目からスタートし、第Yマス目に到達するとゴールとなります。ゴールを越えた場合もゴールと見なします。スタートとゴールにイベントは無く、1マスに複数のイベントが重なることはありません。イベントによって進んだ先のマスのイベントは無視します。所持金が0円より少なくなる場合は0円とします。

例えば、ある人生ゲームで得られるお金の期待値は以下のようにして計算できます。



この例では、スタート、イベントマス (100円入手)、ゴールの3つのマスと、1か2が出るルーレットからなる人生ゲームが表されています。まず、1回目にルーレットを回した時、1が出ればイベントマスに到達し、所持金は100円になります。一方、2が出た場合はゴールに到達し、所持金は0円のままです。これらはどちらも2分の1の確率で起こります。

さらに、1回目でイベントマスに到達した場合は2回目のルーレットを回しますが、どの値が出てもゴールに到達し、所持金はどの場合も100円です。

このように、全部で3通りのゴールの仕方があります。ゴールした時点の所持金に着目すると、0円になる場合が1通りでその確率は2分の1、100円になる場合が2通りでその確率が4分の1です。この場合、ゴールでの所持金の期待値は、ゴールの仕方ごとの (所持金 × その確率) を足した値であり、この人生ゲームの期待値は50円になります。

ルーレットの情報とボードの情報を入力とし、ゴール時の所持金の期待値を出力するプログラムを作成してください。Xは1以上4以下、各 V_i は1以上10以下、Y、Nはそれぞれ1以上50以下、Zは0以上Y-1以下、Eは1以上3以下の整数です。なお、所持金の期待値は小数点以下切り捨ての整数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 X Y Z (整数 整数 整数 ; 半角空白区切り)
2 行目 ルーレットの目の値 V1 V2 ...VX (それぞれ整数 : 半角空白区切り)
3 行目 第1のイベントマスの情報 N E A (整数 整数 整数 ; 半角空白区切り)
4 行目 第2のイベントマスの情報
:
Z+2 行目 第Zのイベントマスの情報

出力

入力データセットごとに、最終的な所持金の期待値を出力します。

入力例

```
1 2 0
1
1 2 1
1
1 2 100
1 2 1
2
1 2 100
2 2 1
1 2
1 2 100
4 5 3
1 2 3 4
1 1 2
2 2 100
4 3 60
0 0 0
```

出力例

```
0
100
0
50
20
```

問題

数の表わし方は色々な種類があり、私たちが普段使っている10進数は一つの代表的なものでしかありません。コンピュータ関連の知識がある人には、2進数、16進数あたりはとても身近なものです。

遙か未来、人類が他の星に移住するようになるのが一般的になりました。ここ、会津に住むNさんも、会津の厳しい冬に耐えかね、これを機に他の星に移住することを決意しました。地球から遠く離れた星に移住したNさんですが、文化の違いに度々苦勞させられながらも、温かい原住民たちに迎えられ、雪かきなどない毎日を送っていました。

さて、Nさんは地球での司書資格を活かして司書の仕事に就きました。最初の仕事は図書整理です。この図書館では、地球の本を蔵書することとなりました。その時代の地球の本には一冊ごとに一貫して番号が与えられており、それに従って整理していくことになりました。

この星ではすべての数をマイナス十進数で表すことになっています。

マイナス十進数は、 $a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$ (各 a_i は0~9の数字) と表記する方法であり、この数は以下の数を表します。

$$a_n \times (-10)^n + a_{n-1} \times (-10)^{n-1} + \dots + a_2 \times (-10)^2 + a_1 \times (-10)^1 + a_0 \times (-10)^0$$

たとえば、マイナス十進数の2156は、以下のとおり十進数の-1944に対応しています。

$$\begin{aligned} 2 \times (-10)^3 + 1 \times (-10)^2 + 5 \times (-10)^1 + 6 \times (-10)^0 &= \\ 2 \times (-1000) + 1 \times (100) + 5 \times (-10) + 6 \times (1) &= \\ -2000 + 100 - 50 + 6 &= -1944 \end{aligned}$$

十進数の番号をマイナス十進数に直すのは大変なので、Nさんは大変不自由な思いをしています。困ったNさんは地球の友達であるあなたに十進数で表された本の番号をマイナス十進数に直すプログラムを作ってくれるように頼んできました。

地球での本の番号を入力とし、この番号のマイナス十進数表記を出力するプログラムを作成してください。また、入力されるマイナス十進数は -2^{31} 以上 2^{31} 未満です。

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 本の番号(整数)

出力

入力データセットごとに、この番号のマイナス十進数表記を出力します。

入力例

```
9
10
-10
-1944
-305432133
0
```

出力例

```
9
190
10
2156
1715573947
```

問題

会津には古くから伝わる埋蔵金の伝説があります。一攫千金を目指すじゅん君は、遂に埋蔵金が埋まっている場所を突き止めました。しかし、じゅん君は埋蔵金発掘のための十分な資金は持っておらず、無駄な作業は控えたいところです。幸いにして、埋蔵金が埋まっている深さ、掘り進める地層の状態は分かっているので、綿密に計画を立てれば最小費用で埋蔵金まで到達できます。しかし、じゅん君はこのような計画をするのは苦手なため、相棒のあなたに手助けを求めてきました。あなたはじゅん君を助けるために、地層の状態を読み取って埋蔵金の埋まっている深さまで最小費用で到達するルートを作成しなくてはなりません。

- ▶ 地層の状態は2次元格子状に配置されたセルで表わされ、各セルの位置は座標(x, y)で表されます。左上を(1, 1)とし、x座標は右に行くにつれて大きくなり、y座標は下に深くなるにつれて大きくなるものとします。じゅん君はy座標の一番小さいセルのうち一つを選んでそこから掘り始め、y座標の一番大きいセルの一つまで掘り進めます。セルの状態は2パターンあり、
 1. 土の詰まったセル。掘るのに、セルごとに決められた費用がかかる。
 2. 酸素のたまったセル。掘る必要はなく、セルごとに決まった量の酸素を補給できる。一度酸素を補給したセルの酸素はなくなり、再度の補給はできない。また、このセルに辿りついたら必ず酸素の補給をしなければならない。となっています。
- ▶ あるセルから掘ることができるのは左右と下方向のセルのみです。
- ▶ 発掘にあたっては、酸素ポンプを携帯しなければなりません。酸素ポンプの残量が0になった瞬間、移動も発掘も酸素の補給もできなくなります。残量はセルを移動するたびに1減ります。酸素ポンプの残量が0で埋蔵金の埋まっている深さまで到達しても、到達したとみなされません。また、酸素のたまったセルでは酸素を補給することができますが、容量を超えた分は廃棄されます。

地層のサイズ(横W, 縦H)、じゅん君の発掘費用f、酸素ポンプの容量m、初期状態で持っている酸素の量o、地層の情報を入力とし、一番深いセルまでたどりつくための最小費用を出力するプログラムを作成してください。ただし、Wは3以上10以下の整数、Hは3以上10以下の整数、fは1以上10000以下の整数、mは3以上50以下の整数、oは3以上m以下の整数とします。また、酸素のたまったセルは50筒以内だとします。なお、最小費用が発掘費用を超えてしまう場合や、どのように掘り進めても埋蔵金にたどりつけない場合は“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 横 W 縦 H (整数 整数; 半角空白区切り)

2行目 発掘費用 f 酸素ポンベの容量 m 初期状態の酸素の量 o (整数 整数 整数; 半角空白区切り)

3行目 地層の1行目の情報 $c_1 c_2 \dots c_W$ (それぞれ整数; 半角空白区切り)

各 c_i は、座標 $(i, 1)$ に対するセルの情報で、意味は以下の通り

負の値: 土の詰まったセルで、値は費用を表す

正の値: 酸素のたまったセルで、値は酸素の量を表す

4行目 地層の2行目の情報

:

$H+2$ 行目 地層の H 行目の情報

出力

入力データセットごとに、最少費用または NA を出力します。

入力例

3 3
100 10 10
-100 -20 -100
-100 -20 -100
-100 -20 -100
3 3
100 10 10
-100 -20 -100
-100 -20 -20
-100 -60 -20
3 3
100 10 3
-100 -20 -100
-20 -20 -20
-20 -100 -20
3 3
100 3 3
-100 -20 -30
-100 -20 2
-100 -20 -20
4 5
1500 5 4
-10 -380 -250 -250
-90 2 -80 8
-250 -130 -330 -120
-120 -40 -50 -20
-250 -10 -20 -150
0 0

出力例

60
80
NA
50
390

問題

「ライアン軍曹を救え」という指令のもと、アイツ軍の救出部隊はドイツリーの水上都市で敵軍と激しい戦闘を繰り広げていました。彼らは無事に軍曹と合流しましたが、敵の戦車が多く、救出ヘリを呼べずにいました。そこで、彼らは敵の戦車を混乱させるため、都市にある橋を全て爆破するという作戦を実行することにしました。

この作戦はすぐに司令部に伝えられ、救出ヘリの準備が進められました。救出ヘリを飛ばすためには、いつ橋が全て爆破されるかを予測しなければなりません。そこで軍のプログラマであるあなたは、救出部隊が全ての橋の爆破に必要な時間を計算することになりました。

ドイツリーの水上都市は N 個の島で構成されており、島と島の間には橋がかかっています。すべての島はツリー状に繋がっています（下図参照）。ある島からある島への経路は、一通りだけ存在します。各橋を渡るには、橋ごとに決められた時間がかかり、どちらの方向にもその時間で橋を渡ることが可能です。

救出部隊はボートなど水上を移動する手段を持っていないので島と島の間を移動するには橋を通る他ありません。救出部隊は、その時いる島に隣接している橋のうち、必要なものを一瞬で爆破することができます。救出部隊が全ての橋を爆破するのに必要な最小の時間はいくらでしょうか。ただし、島の中での移動時間は考えません。

島の数、それぞれの橋情報を入力とし、橋を全て爆破するのに必要な最小の時間を出力するプログラムを作成してください。島はそれぞれ $1 \sim N$ の番号で表されます。橋は $N-1$ 本あります。橋情報は、その橋が隣接している二つの島の番号 (a, b) と、その橋を渡るのに必要な時間 t で構成されます。救出部隊は島番号 1 の島からスタートするものとします。また、入力はすべて整数で、 N は 2 以上 20 以下、 t は 1 以上 500 以下であるものとします。

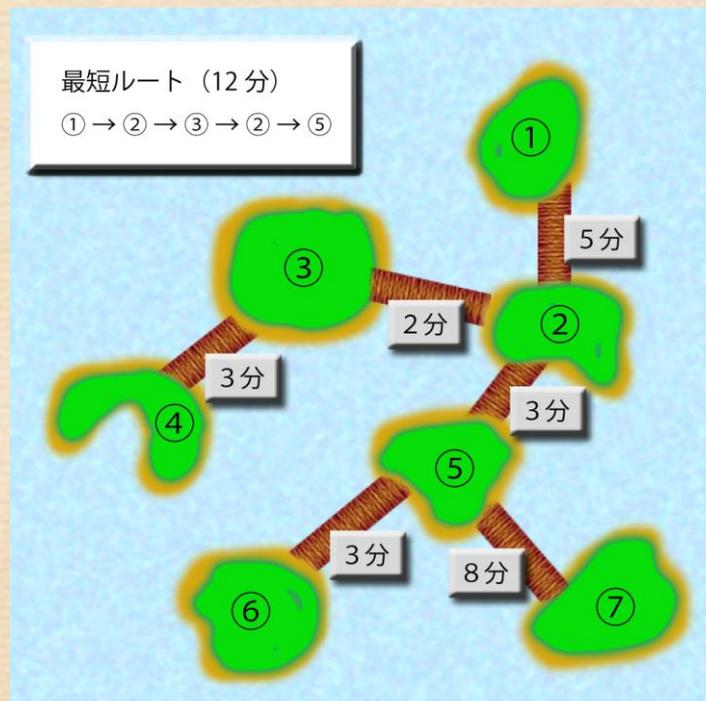


図 都市の例

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 島の数 N (整数)
2 行目 1 本目の橋の情報 $a b t$ (整数 整数 整数; 半角空白区切り)
3 行目 2 本目の橋の情報
:
N 行目 $N-1$ 本目の橋の情報

出力

入力データセットごとに、橋を全て爆破するのに必要な最小の時間を出力します。

入力例

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

出力例

```
12
```


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 W H (整数 整数 ; 半角空白区切り)

2 行目 碁盤の 1 行目の情報 c1 c2 ... cW (整数 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。

c_i : この行における i 番目のマス目の情報 (0, 1 のいずれか)

0 : 何も置かれていない

1 : 石像が置かれている

3 行目 碁盤の 2 行目の情報

:

H+1 行目 碁盤の H 行目の情報

出力

入力データセットごとに、Yes または No を出力します。

入力例

```
5 4
0 0 0 0
0 1 1 0 0
0 0 1 0 1
1 0 0 0 1
5 4
0 0 0 0 0
0 1 1 0 0
0 0 0 0 1
1 0 0 0 1
0 0
```

出力例

```
Yes
No
```

問題

秘境の地“ナラマンダ地方”

この奥深い洞窟に眠る黄金の伝説を求め、探検家ボブとマイクがやってきました。二人は途中、さまざまな困難に立ち向かい、ようやくたどり着いた最後の扉。この先には夢に見た黄金が待っています。

その扉には多数の三角形が描かれていました。試しにマイクがその1つに触れてみると、その三角形は光を放ちました。光は三角形の底辺から垂直に頂点方向へ長方形を描きながら伸びていき、その先にある別の三角形に触れました。すると、光に触れた三角形も同じように光を放ち、それを繰り返していききました。

扉の三角形は全て二等辺三角形（正三角形は含まない）で、どれも同じ大きさで、またどれも光は一定の距離だけ伸びていました。最後の三角形が光り終わると、そこには何事もなかったかのような静けさが戻ってきました。

ボブは自分の足下に奇妙なメッセージを見つけました。

「むやみに触れるべからず。汝正しくこの扉に触れたとき、この扉は放たれん。」

ボブとマイクはこのメッセージの意味を考えながら、何度も三角形に触れてみましたがなかなか扉は開きません。

しかし、それを繰り返していくうちに、二人はメッセージの意味がだんだん分かってきました。

「“むやみに触れるべからず”とは、扉に触れる回数は最小ということ。

できるだけ少ない回数で触れて、全ての三角形を光らせれば、きっと扉は開くはずだ！」と考えたのです。

図で説明しよう。扉には図1のように複数の二等辺三角形が描かれてあります。

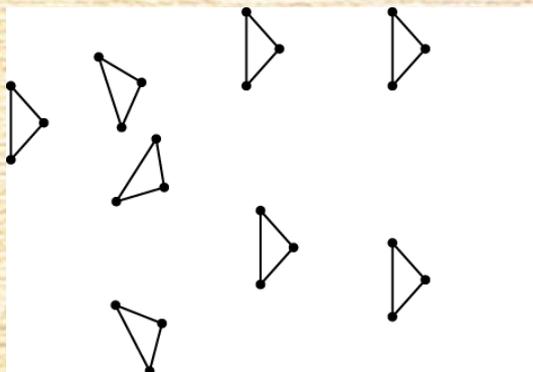


図1

→次ページへつづきます

最初に左下の三角形に触れると、図 2 に示すように、その三角形の底辺から垂直に頂点方向へ光が放たれ、その光が触れる三角形が次々に光ります。

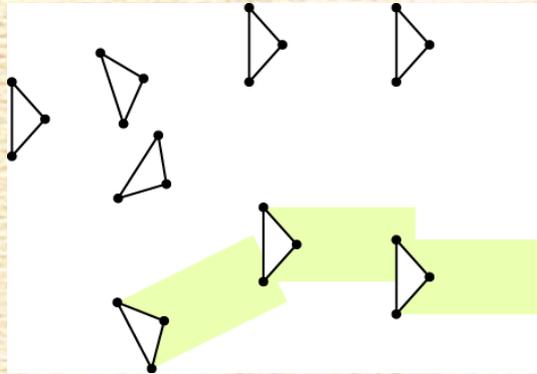


図 2

次に左端の三角形に触れると、図 3 に示すように、全ての三角形が光ります。この手順が全ての三角形を光らせるための最少の手数 (2 手) となります。

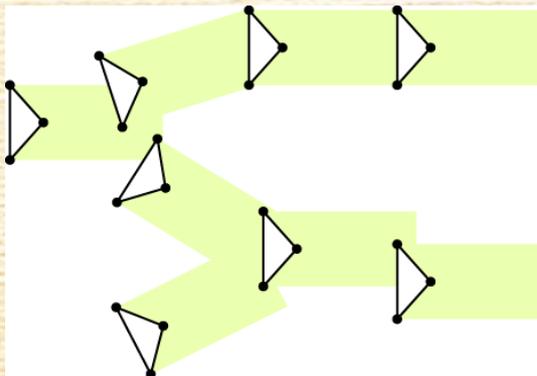


図 3

さあ、ボブとマイクが最後の扉を開ける手助けをしよう。

扉に描かれている三角形の個数 n 、光の伸びる長さ d 、それぞれの三角形の頂点の座標 (x_1, y_1) (x_2, y_2) (x_3, y_3) を入力とし、全ての三角形を光らせるために扉に触れる最少の回数を出力するプログラムを作成して下さい。光の伸びる長さ d は、底辺からの長さであるものとします。ただし、 n は 1 以上 100 以下の整数、 d は 1 以上 1000 以下の整数、三角形の頂点の座標は -1000 以上 1000 以下の実数とします。

なお、この問題を解くにあたっては、2 点の距離が 0.01 以下の場合には、同一の点と処理します。また、1 点とひとつの直線の距離が 0.01 以下の場合には、この点はこの直線上にあるとして処理します。

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 n d (整数 整数 ; 半角空白区切り)

2行目 第1の三角形の3つの頂点の座標 x_1 y_1 x_2 y_2 x_3 y_3 (すべて実数 ; 半角空白区切り)

3行目 第2の三角形の3つの頂点の座標

:

$n+1$ 行目 第 n の三角形の3つの頂点の座標

出力

入力データセットごとに、扉に触れる最少の回数を出力します。

入力例

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

出力例

```
1
3
```