

## 問題 1 ワード

(3点)

コンピュータで扱われるデータの最小単位をビット(bit)と呼び、複数のビットをまとめて表した情報量をワード(word)と呼びます。現在、多くのコンピュータでは1ワードを32ビットとして処理しています。

### 課題

1ワードを32ビットで表すコンピュータについて、ワード単位で与えられたデータ量 $W$ をビット単位で出力するプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

W
---

入力は1行からなり、データ量 $W(0 \leq W \leq 100)$ が与えられる。

### 出力

ビット単位の値を1行に出力する。

### 入出力例

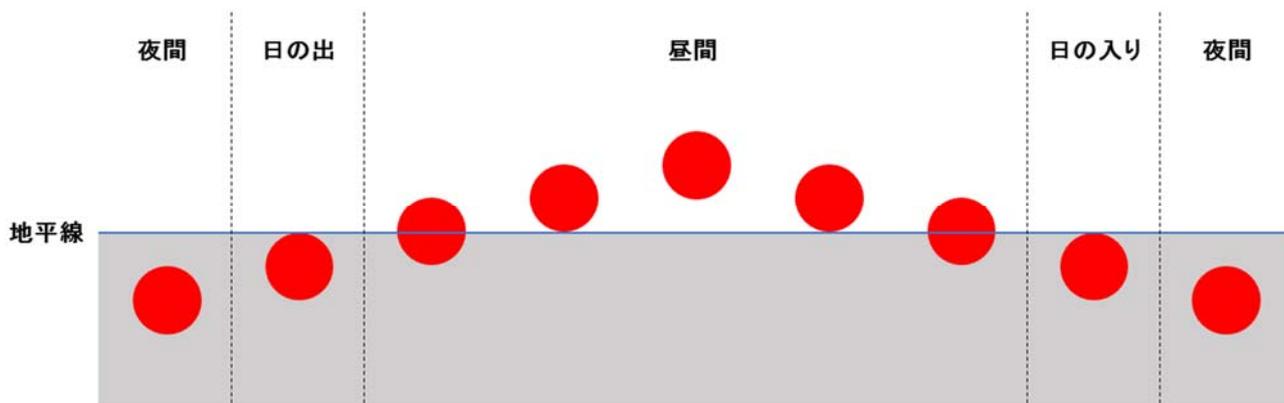
入力例 1	出力例 1
4	128
入力例 2	出力例 2
3	96

## 問題2 日の出と日の入り

(4点)

太陽が現れることを「日の出」、隠れることを「日の入り」と呼びますが、その厳密な時刻は太陽が地平線に対してどのような位置にある時でしょうか。

下の図のように、太陽を円、地平線を直線で表すことにします。このとき、太陽の「日の出」「日の入り」の時刻は、太陽を表す円の上端が地平線を表す直線と一致する瞬間とされています。日の出の時刻を過ぎ、円の上端が直線より上にある時間帯が昼間、円が直線の下へ完全に隠れている時間帯が夜間となります。



### 課題

ある時刻の地平線から太陽の中心までの高さ、太陽の半径を入力とし、その時刻が「昼間」か、「日の出または日の入り」か、「夜間」かを出力するプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

H R
-----

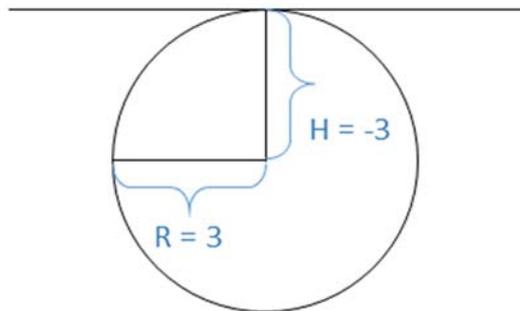
入力は1行からなり、ある時刻の地平線から太陽の中心までの高さを表す整数  $H$  ( $-1000 \leq H \leq 1000$ ) と半径を表す整数  $R$  ( $1 \leq R \leq 1000$ ) が与えられる。ただし、 $H$  は太陽の中心が地平線上にあるときを  $0$  として、それより上にあるときは正、下にあるときは負とする。

### 出力

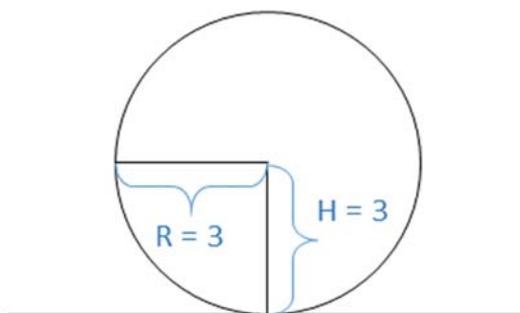
昼間のとき「1」、日の出または日の入りのとき「0」、夜間のとき「-1」を1行に出力する。

## 入出力例

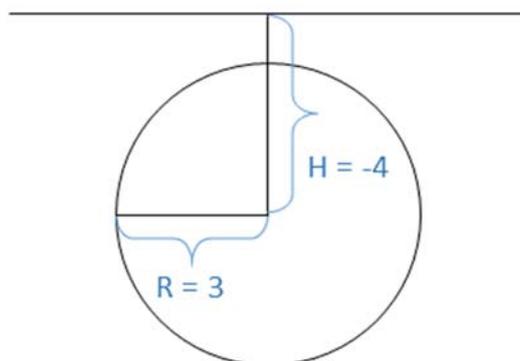
入力例 1	出力例 1
-3 3	0



入力例 2	出力例 2
3 3	1



入力例 3	出力例 3
-4 3	-1



### 問題3 日本の暦

(5点)

「西暦」は西洋から輸入された概念ですが、日本には暦上の年を表す方法として「元号」に年を付けて識別する和暦という概念があります。例えば今年(2016年)は西暦なら2016年ですが、和暦なら平成28年です。どちらもよく使われる年の表現方法ですが、ある年の西暦がわかっても和暦で何年かわからない、またはその逆の状況を経験したことはありませんか？

#### 課題

西暦で年が与えられたとき和暦の年を、和暦で年が与えられたとき西暦の年を出力するプログラムを作成せよ。ただし、西暦と和暦の対応は、簡単のため以下のようにする。

西暦	和暦
1868年から1911年	明治元年から明治44年
1912年から1925年	大正元年から大正14年
1926年から1988年	昭和元年から昭和63年
1989年から2016年	平成元年から平成28年

#### 入力

入力は以下の形式で与えられる。

E Y

入力は1行からなり、E( $0 \leq E \leq 4$ )は与えられる暦の種類、Yはその暦での年であり、Eが0のときは西暦Y( $1868 \leq Y \leq 2016$ )年、1のときは和暦の明治Y( $1 \leq Y \leq 44$ )年、2のときは和暦の大正Y( $1 \leq Y \leq 14$ )年、3のときは和暦の昭和Y( $1 \leq Y \leq 63$ )年、4のときは和暦の平成Y( $1 \leq Y \leq 28$ )年を表す。

#### 出力

西暦ならば和暦に、和暦ならば西暦に変換した結果を、1行に出力する。ただし、西暦を和暦に変換した結果は、明治なら文字「M」、大正なら文字「T」、昭和なら文字「S」、平成なら文字「H」を先頭に付けて出力する。

#### 入出力例

入力例1	出力例1
0 2015	H27
入力例2	出力例2
0 1912	T1
入力例3	出力例3
2 1	1912
入力例4	出力例4
4 28	2016

## 問題4 ニュータウン

(8点)

会津県では人口増加のためにニュータウンを作ることにしました。そのために、新たに長方形の土地を開拓し、この土地を余すところなく、すべて同じ大きさの正方形からなる区画に区切ることを決めました。この土地の整備には、区画数に比例した費用がかかりますが、県としてはこの費用を最小にしたいと考えています。

### 課題

新たに開拓した土地の東西方向と南北方向の長さ、1区画当たりの整備費用が与えられたとき、すべての区画を整備したときにかかる、最小の整備費用を求めるプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

W H C
-------

入力は1行であり、新たに開拓した土地の東西方向の長さ $W$  ( $1 \leq W \leq 1000$ ) と南北方向の長さ $H$  ( $1 \leq H \leq 1000$ )、区画当たりの整備費用 $C$  ( $1 \leq C \leq 1000$ ) が整数で与えられる。

### 出力

土地を整備するために必要な最小の費用を1行に出力する。

### 入出力例

入力例1	出力例1
10 20 5	10

2個の正方形で土地を余すところなく区切ることができ、1区画当たりの整備費用が5なので、10と出力する。

入力例2	出力例2
27 6 1	18

18個の正方形で土地を余すところなく区切ることができ、1区画当たりの整備費用が1なので、18と出力する。

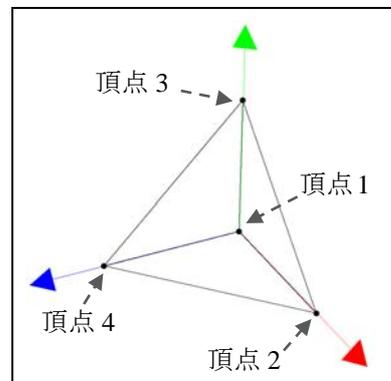
## 問題5 形状データ処理

(8点)

コンピュータグラフィクスでは、三次元の形状を表現する方法として、ポリゴンモデルが使われます。ポリゴンモデルとは、頂点座標と、それらの頂点のつなぎ方を与えて面を作るモデルです。

一般のポリゴンモデルでは、任意の多角形を扱えますが、今回は三角形からなるポリゴンモデルを考えることにします。任意のポリゴンモデルは三角形を表す面情報の集まりとして表すことができます。

一つの面情報は、3つの頂点を並べて表します。ただし、並び方が異なるだけで同じ3点からなる場合は、同じ面情報を表すことにします。例えば、右の図の四面体で、頂点1, 2, 3を繋いでできる面は、頂点2, 3, 1や、頂点3, 2, 1などのように表すこともできます。このように、同じ面情報が複数あると無駄になるので、1つにまとめてしまった方が良いでしょう。



### 課題

面情報が与えられたとき、重複した面を無くすために消さなければならない面情報の個数を求めるプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

```
N
P11 P12 P13
P21 P22 P23
:
PN1 PN2 PN3
```

1行目に、ポリゴンモデルの面情報の数 $N$  ( $1 \leq N \leq 1000$ ) が与えられる。続く $N$ 行に、 $i$ 番目の面を作るために使う頂点の番号 $p_{ij}$  ( $1 \leq p_{ij} \leq 1000$ ) が与えられる。ただし、一つの面について、同じ頂点を2度以上使うことはない ( $p_{i1} \neq p_{i2}$ かつ $p_{i2} \neq p_{i3}$ かつ $p_{i1} \neq p_{i3}$ である)。

### 出力

重複した面を無くすために消さなければならない面情報の個数を1行に出力する。

## 入出力例

入力例 1	出力例 1
4 1 3 2 1 2 4 1 4 3 2 3 4	0

入力例 2	出力例 2
6 1 3 2 1 2 4 1 4 3 2 3 4 3 2 1 2 3 1	2

入出力例 2 では、1 つ目と 5 つ目と 6 つ目の面は頂点 1, 3, 2 を使って三角形を作っていて、点の順番が異なるだけなので重複している。つまり、重複した面のうち 2 つの面を消せば重複した面は無くなる。

## 問題6 パンケーキの焼き上がり

(10点)

あなたが勤めているパンケーキ屋では、細長い鉄板にパンケーキの生地を横1列に並べて焼きます。パンケーキはへらで何回か裏返せば完成します。何回以上裏返せば完成するかはパンケーキごとに異なります。

へらは大きいので、隣り合ったパンケーキは2枚同時に裏返されてしまいます。このとき、これら2枚の位置は入れ替わりません。ただし、両端だけは、隣のパンケーキといっしょに裏返すだけでなく、1枚だけ裏返すこともできます。すべてのパンケーキを必要な回数以上裏返したら、全部いっぺんに鉄板からおろして完成です。

パンケーキを必要な回数より多く裏返すと固くなってしまいますので、あまり多く裏返したくありません。そこであなたは、すべて完成するまでに、各パンケーキが裏返る回数の総和が最小になるような方法を見つけようと考えました。

### 課題

鉄板の上のパンケーキの枚数と、完成するまでに何回以上裏返さなければならないかがパンケーキごとに与えられているとき、すべて完成するまでに各パンケーキが裏返る回数（へらを操作する回数ではない）の総和の最小値を計算するプログラムを作成せよ。

### 入力

入力は複数のデータセットからなる。入力の終わりはゼロ1つの行で示される。各データセットは以下の形式で与えられる。

N
p <sub>1</sub> p <sub>2</sub> ... p <sub>N</sub>

1行目にパンケーキの枚数  $N$  ( $3 \leq N \leq 5000$ ) が与えられる。2行目に各パンケーキが完成するまでに必要な裏返す回数  $p_i$  ( $0 \leq p_i \leq 3$ ) が与えられる。

データセットの数は100を超えない。

### 出力

すべて完成するまでに各パンケーキが裏返る回数の総和の最小値を1行に出力する。

## 入出力例

入力例	出力例
3	4
1 2 1	6
3	
0 3 0	
0	

入力例一番目のデータセットでは、へらを1回操作して左端と真ん中のパンケーキを裏返すと、2個のパンケーキが1回ずつ裏返るので、この操作で裏返る回数は2回。さらに、へらを1回操作して真ん中と右端のパンケーキを裏返すと、2個のパンケーキが1回ずつ裏返るので、この操作で裏返る回数は2回。以上の総和4回が答えになる。

入力例二番目のデータセットでは、へらを1回操作して左端と真ん中のパンケーキを裏返すと、この操作で裏返る回数は2回。これを3回繰り返したときの、総和の6回が答えになる（真ん中のパンケーキは、両隣のどちらかのパンケーキと一緒に裏返すことしかできないことに注意）。

## 問題7 イワシロ・イツァ

(10点)

古代国家イワシロを発見し調査を進めた我々研究者は、ついにイワシロの中心部にある神殿、イワシロ・イツァを発見した。伝承の通り、イワシロ・イツァの最深部にはイワシロの神にささげた石版が保管されていた。全ての石版を調べた結果、必ず文章と呪文が一つずつ、合計2つの文字列が書かれていた。各文字列は左から右へ読む。

我々の説に従えば、イワシロでは呪文が文章の中に、何回現れるかが重要な意味を持つ。ただし、呪文に含まれるすべての文字が順番に、文章の中にとびとびで現れるものも1回現れていると考える。例えば、文章が“abab”で、呪文が“ab”であれば、連続でないものも含めて“ab”は“abab”の中に3回現れている (abab, abab, abababの3通り)。

### 課題

文章と呪文が与えられたとき、呪文が文章の中に何回現れるかを出力するプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

t
b

1行目に、石版に書かれた文章を表す文字列tが与えられる。2行目に、石版に書かれた呪文を表す文字列bが与えられる。どちらの文字列も英小文字だけから成る、長さが1以上1000以下の文字列である。

### 出力

呪文が文章の中に何回現れるかを1行に出力する。ただし、出力すべき値は非常に大きくなりうるので、代わりに1,000,000,007で割った余りを出力する。

### 入出力例

入力例1	出力例1
abab ab	3
入力例2	出力例2
aaaabaaaabaaaabaaaab aaaaa	4368
入力例3	出力例3
data structure	0

## 問題 8 村の道路計画

(14点)

アイツ国のワカマツ平野に、集落が点在していました。いくつかの集落の間はまっすぐで互いに行き来できる道でつながっていて、ワカマツ平野にあるどの集落の間も道を辿って行き来ができます。それぞれの道には長さに応じた維持費がかかりますが、すべての集落が資金を出し合って道を維持していました。

あるとき、すべての集落が一つの村にまとまることになり、村を囲む境界線を引くことになりました。アイツ国の決まりでは、村を構成するどの2つの集落を結んだまっすぐな線も、村の外を通ってはいけません（境界線上を通ることは許されます）。さらに、アイツ国では村を囲む境界線上に道がなければなりません。境界線上に道がない場所には、アイツ国が新たに道を作ってくれます。

しかし、道の維持費は村が支払うので、村人達は境界線をできるだけ短くしたいと考えています。さらに、村人達はすべての集落の間に行き来できる状態を維持しつつ、境界線上にない道を廃止することで、道の長さの合計を最小にすることにしました。

### 課題

集落の位置と元々あった道の情報が与えられる。境界線上に道を置き、かつ、すべての集落が行き来できるようにした場合の、道の長さの合計の最小値を計算するプログラムを作成せよ。ただし、集落は大きさのない点、道は幅のない線分とする。

### 入力

入力は以下の形式で与えられる。

```
V R
X1 Y1
X2 Y2
:
XV YV
S1 t1
S2 t2
:
SR tR
```

1行目に集落の数 $V$  ( $3 \leq V \leq 100$ ) と道の数 $R$  ( $2 \leq R \leq 1000$ ) が与えられる。

続く $V$ 行に集落を表す点の情報が与えられる。各行に与えられる2つの整数 $x_i, y_i$  ( $-1000 \leq x_i, y_i \leq 1000$ ) は、それぞれ $i$ 番目の集落の $x$ 座標、 $y$ 座標を表す。

続く $R$ 行に元々あった道の情報が与えられる。各行に与えられる2つの整数 $s_i, t_i$  ( $1 \leq s_i < t_i \leq V$ ) は、 $i$ 番目の道が $s_i$ 番目の集落と $t_i$ 番目の集落をつないでいることを表す。

入力は以下の条件を満たす。

- $i \neq j$ ならば、 $i$ 番目の集落と $j$ 番目の集落の座標は異なる。
- どの2つの集落についても、それらを直接結ぶ道は高々1つしか現れない。

- 2つの異なる道が端点以外の点を共有することはない。
- 3つ以上の集落が同一直線上に並んでいることはない。

## 出力

条件を満たす道の長さの合計の最小値を1行に実数で出力する。ただし、誤差がプラスマイナス0.001を超えてはならない。この条件を満たせば小数点以下何桁表示してもよい。

## 入出力例

入力例 1	出力例 1
5 5 0 0 1 1 3 0 3 2 0 2 1 2 2 3 2 4 3 4 1 5	11.4142

入力例 2	出力例 2
7 6 0 2 3 0 2 2 1 0 4 1 2 3 3 5 1 3 2 4 2 3 3 5 3 7 6 7	18.2521

## 問題9 プログラミングコンテスト II

(18点)

白虎大学では毎年プログラミングコンテストを開催しています。コンテストは全てのチームの得点が0の状態から開始し、解答状況に応じて得点が加算されていきます。このコンテストでは、得点の高い順に順位付けが行われます。チームの総数をNとしたとき、チームには1からNの番号がそれぞれ割り当てられています。得点と同じ場合は番号がより小さい方が上位になります。

白虎大学では、コンテストを盛り上げるために観戦用のランキングシステムを開発しています。開発チームの一員であるあなたは、このシステムの一部であるプログラムの作成を任されています。

### 課題

与えられた命令に従って、得点の更新と、指定された順位のチームの番号と得点を報告するプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

```
N C
command1
command2
:
commandc
```

1行目にチーム数 $N$  ( $2 \leq N \leq 100000$ ) と命令の数 $C$  ( $1 \leq C \leq 100000$ ) が与えられる。続く $C$ 行に、1行ずつ命令が与えられる。各命令は以下の形式で与えられる。

```
0 t p
```

または

```
1 m
```

最初の数字が0のとき更新命令、1のとき報告命令を表す。更新命令では指定された番号 $t$  ( $1 \leq t \leq N$ ) のチームに、整数で与えられる得点 $p$  ( $1 \leq p \leq 10^9$ ) を加算する。報告命令では指定された順位 $m$  ( $1 \leq m \leq N$ ) のチームの番号と得点を報告する。ただし、報告命令は少なくとも1回現れるものとする。

### 時間制限

入力に対して、実行時間が3秒を超えてはならない。

### 出力

各報告命令に対して、指定された順位のチームの番号と得点を空白区切りで1行に出力する。

## 入出力例

入力例 1	出力例 1
3 11	1 5
0 2 5	2 5
0 1 5	3 4
0 3 4	3 6
1 1	3 6
1 2	1 5
1 3	
0 3 2	
1 1	
0 2 1	
1 2	
1 3	

入力例 2	出力例 2
5 2	1 0
1 1	2 0
1 2	

## 問題 10 ゲームの攻略

(20点)

あなたは所属するプログラミング部の部室から古びたボードゲームを発見しました。面白そうなので遊んでみることにしました。

このゲームは  $M$  個のイベントから成り、時刻  $t_i$  にイベント  $i$  を攻略しなければいけません。ただし、そのときにあなたの強さは  $s_i$  以上である必要があり、 $s_i$  以上にできない場合はゲームオーバーになります。

ゲーム開始時 (時刻 0) のあなたの強さは 0 ですが、アイテムを買うことで強さを増加させることができます。ゲーム開始時のあなたの所持金は 0 ですが、1 単位時間あたり 1 増加します。

ボードにはそれぞれ 1 から  $N$  の番号が付けられた  $N$  個のアイテムが順番に並べられており、アイテム  $i$  の値段は  $v_i$  で、それを購入するとあなたの強さが  $h_i$  増加します。アイテムは所持金が十分であれば好きな時刻に好きな数だけ購入することができますが、残っているアイテムの中で番号が小さいものから順に選ばなければなりません。各アイテムは 1 度購入すると消滅します。

また、同じ時刻に複数のアイテムを連続で買うことができ、このとき隣り合うアイテムの  $h_i$  の差分の和をボーナスとして得ることができます。例えば、ある時刻にアイテム 1, 2, 3 を同時に買った場合、 $h_1 + h_2 + h_3$  に加えて  $|h_1 - h_2| + |h_2 - h_3|$  だけあなたの強さが増加します。

あなたは、全てのイベントを攻略した後の所持金を最大化したいと考えています。

### 課題

アイテムの情報とイベントの情報を入力とし、すべてのイベントを攻略した後の所持金の最大値を出力するプログラムを作成せよ。

### 入力

入力は以下の形式で与えられる。

$N$	$M$
$v_1$	$h_1$
$v_2$	$h_2$
:	
$v_N$	$h_N$
$t_1$	$s_1$
$t_2$	$s_2$
:	
$t_M$	$s_M$

1 行目にアイテムの数  $N$  ( $1 \leq N \leq 3000$ ) とイベントの数  $M$  ( $1 \leq M \leq 1000$ ) が与えられる。続く  $N$  行にアイテム  $i$  の値段  $v_i$  と強さの増加量  $h_i$  ( $1 \leq v_i, h_i \leq 100000$ ) が与えられる。続く  $M$  行にイベント  $i$  の時刻  $t_i$  と条件  $s_i$  ( $1 \leq t_i, s_i \leq 100000$ ) が与えられる。ただし、 $i < j$  のとき、 $t_i < t_j$  とする。入力はすべて整数で与えられる。

### 時間制限

入力に対して、実行時間が 3 秒を超えてはならない。

## 出力

所持金の最大値を1行に出力する。ただし、攻略することができない場合には「-1」を出力する。

## 入出力例

入力例 1	出力例 1
5 4 3 3 2 1 1 5 4 2 2 6 4 1 8 2 10 4 12 17	2

入力例 2	出力例 2
5 4 3 3 2 1 1 5 4 2 2 6 4 1 8 2 10 4 12 30	-1