

問題01 キャンディーとクラス旗(3点)

3年C組では、平成19年11月10日の体育祭で使用する「クラス旗」を、将来のクラス会の時にも使うことにしました。そこで「クラス旗」を保管する生徒を決めるために、先生が先日差し入れてくれた大量のキャンディーを使って次のようなゲームを行うことにしました。

1. 各生徒は生徒番号の順に1個ずつキャンディーを取ります。
2. 一巡してもキャンディーが残っていたら、最初の生徒番号の人から順々にキャンディーを取り続けます。
3. 最後のキャンディーを取った人が「クラス旗」を保管する生徒になります。



3年C組のクラスの人数は39人です。彼らの生徒番号は3C01から3C39です。例えば、キャンディーの数が50個の場合、クラス全員が1個目のキャンディーを取り終わると、キャンディーの残りは11個となります。それを再び生徒番号順にとると、最後の1個は、3C11の生徒が取ることとなります。すなわち3C11の生徒が「クラス旗」を保管する生徒となります。キャンディーの個数を入力とし、「クラス旗」を保管する生徒の生徒番号を出力して終了するプログラムを作成してください。

入力

キャンディーの個数 (1 以上 10,000 以下の整数)

出力

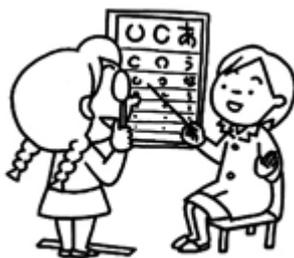
「クラス旗」を保管する生徒の生徒番号 (半角英数字)

入力例1	出力例1
50	3C11
入力例2	出力例2
5576	3C38
入力例3	出力例3
5577	3C39
入力例4	出力例4
5578	3C01

問題02 視力検査(3点)

視力検査がありました。検査結果データを入力とし、下記の視力判定表に基づいて各判定に当てはまる人数を、左右の視力別に出力して終了するプログラムを作成してください。ただし、視力は0.1以上2.0以下の0.1刻みとし、生徒の数は40人以下とします。

判定	視力
A	1.1以上
B	0.6以上1.1未満
C	0.2以上0.6未満
D	0.2未満



入力

1行目 1人目の左の視力 右の視力 (実数 実数; 半角空白区切り)
2行目 2人目の左の視力 右の視力 (実数 実数; 半角空白区切り)
:
:

出力

1行目 左の視力がAの人数 右の視力がAの人数 (整数 整数; 半角空白区切り)
2行目 左の視力がBの人数 右の視力がBの人数 (整数 整数; 半角空白区切り)
3行目 左の視力がCの人数 右の視力がCの人数 (整数 整数; 半角空白区切り)
4行目 左の視力がDの人数 右の視力がDの人数 (整数 整数; 半角空白区切り)

入力例	出力例
1.0 1.2	2 3
0.8 1.5	2 1
1.2 0.7	0 0
2.0 2.0	0 0

問題03 双子素数(8点)

素数は暗号、符号、通信、計測などに広く利用されています。

素数の中でも「5と7」「11と13」のような差が2となる素数の組を双子素数と言います。

双子素数を構成する素数のうち大きい素数をその双子素数の大きさと呼ぶことにします。

入力された整数 n 以下の双子素数で大きさが最大のものを出力し終了するプログラムを作成してください。 n は 5 以上 10,000 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。

n (整数)

出力

入力データセットごとに以下の形式で出力します。

n に対する双子素数 $p1$ $q1$ (整数 整数; 半角空白区切り)

$p1$: 双子素数を構成する小さい素数

$q1$: 双子素数を構成する大きい素数

入力例	出力例
12	5 7
100	71 73
200	197 199
300	281 283
0	

問題04 マス目(8点)

$n \times n$ のマス目にそれぞれ1か0が書いてあります。

この情報を入力とし、上下、左右、または、対角線方向に並んだ1の列のうち、最大のものの長さを出力して終了するプログラムを作成してください。例えば、下の図で一番長いのは点線で囲まれた列です。

0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
1	0	1	0	1
0	0	0	1	0

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 マス目の大きさ n (2以上255以下の整数)

2行目 マス目の1行目の情報 (1と0だけを含む n 文字の文字列)

3行目 マス目の2行目の情報 (1と0だけを含む n 文字の文字列)

:

$n+1$ 行目 マス目の n 行目の情報 (1と0だけを含む n 文字の文字列)

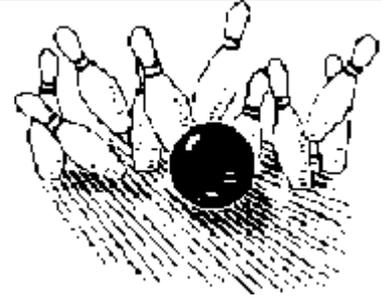
出力

入力データセットごとに列の長さの最大値 (整数) を出力します。

入力例	出力例
5	4
00011	8
00101	1
01000	0
10101	
00010	
8	
11000001	
10110111	
01100111	
01111010	
11111111	
01011010	
10100010	
10000001	
2	
01	
00	
3	
000	
000	
000	
0	

問題05 ボウリングのスコア計算(8点)

クラスのレクリエーションとしてボウリングを行うことになりました。参加者ごとの投球情報を入力とし、得点の高い順に成績情報を出し終了するプログラムを作成してください。なお、同点の場合は学籍番号の若い順に出力してください。ただし参加者は3名以上40名以下とし、1人当たり1ゲームずつ投球しているものとします。



ボウリングとはご存じのとおり、プレイヤーに対して頂点を向けて正三角形に並べられた、10本のピンをめがけてボールを転がし、ピンを倒すスポーツです。1回目の投球ですべて転倒した場合をストライクと言い、その投球のみで次のフレームに進みます。ストライクでない場合は、残ったピンをそのままにして2回目の投球を行います。2回目の投球ですべて転倒した場合をスペアと言います。2回目の投球終了後、次のフレームに進みます。

1ゲームは10のフレームから構成され、第1～9の各フレームは2回投球できます。各フレームの開始時点では、10本のピンがすべて立った状態で用意されます。第10フレームは、ストライクもしくはスペアが出た場合には計3回の投球を、それ以外の場合は2回の投球を行い、ゲーム終了となります。

スコア例1

フレーム	1	2	3	4	5	6	7	8	9	10
投球内容	8	▲▲	9	7	-	▲▲	▲▲	-	8	▲▲
フレーム得点	20	20	17	7	30	20	18	8	30	30
合計得点	20	40	57	64	94	114	132	140	170	200

スコア例2 (最高得点 300点の場合)

フレーム	1	2	3	4	5	6	7	8	9	10
投球内容	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
フレーム得点	30	30	30	30	30	30	30	30	30	30
合計得点	30	60	90	120	150	180	210	240	270	300

スコア計算の方法

- 各フレームにおいてスペア、ストライクがない場合は、2回の投球で倒したピンの本数がそのフレームの得点となります。(スコア例1の第4フレームと第8フレーム)
- スペアを出した場合、倒した本数である10点に加え、次の投球で倒したピンの本数がこのフレームの得点に加算されます。(スコア例1の第1フレームと第2フレームの関係など)
スコア例1の第1フレームでは第2フレームの1投で倒した10本(点)を加えた20点が得点となります。第3フレームも同様の計算方法です。
- ストライクを出した場合、倒した本数である10点に加え、続く2回の投球で倒したピンの本数が加算されます。(スコア例1の第2フレームと第3フレームの関係など) もちろん続く2投中にストライクの場合があります。(スコア例1の第5フレームと第6、7フレームの関係など)
- 第10フレームのみ、スペア、ストライクを出した場合、3投して倒したピンの総数が第10フレームの得点として加算されます。
- 各フレームの得点の合計が1ゲームの得点となり、最高得点は300点となります。

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 参加者数m (3以上40以下の整数)
 2行目 1人目の参加者情報
 3行目 2人目の参加者情報
 :
 m+1行目 m人目の参加者情報

参加者情報は1行ずつ次の形式で与えられます。

学籍番号 第1投の転倒ピン数 第2投の転倒ピン数... 第n投の転倒ピン数 (整数 整数 整数...
 整数; 半角空白区切り)

学籍番号は0以上9999以下の整数、転倒ピン数は0以上10以下の整数、総投球数nは、12以上21以下の整数で、得点計算に必要なピン数が過不足なく与えられるものとします。

出力

入力データセットごとに得点の高い順 (同点の場合は学籍番号の若い順)に次の形式で出力します。

1行目 成績情報 (学籍番号 得点) (整数 整数; 半角空白区切り)
 2行目 成績情報 (学籍番号 得点) (整数 整数; 半角空白区切り)
 :
 m行目 成績情報 (学籍番号 得点) (整数 整数; 半角空白区切り)

入力例	出力例
3	1200 127
1010 6 3 10 7 1 0 7 9 1 10 6 2 4 3 9 1 9 0	1010 123
1200 5 3 9 1 7 1 0 0 8 1 10 10 4 3 9 1 8 2 9	1101 60
1101 3	3335 300
4	3321 200
3321 8 2 10 9 1 7 0 10 10 10 0 8 10 10 10 10	3340 175
3332 5 0 10 9 1 4 1 9 0 10 10 7 1 5 2 8 1	3332 122
3335 10 10 10 10 10 10 10 10 10 10 10 10	
3340 8 2 7 3 6 4 8 2 8 2 9 1 7 3 6 4 8 2 9 1 7	
0	

問題06 三角形と円(8点)

平面上にある三角形と円の位置関係を判定し、出力して終了するプログラムを作成してください。対象となる図形はいずれも境界を含むものとします。

三角形は3頂点の位置が与えられ、円は中心の位置と半径が与えられます。

位置は直交座標系による2つの整数の組によって与えられます。半径も整数で与えられます。与えられる整数はすべて、1以上10,000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 三角形の第1の頂点の座標 x_1 y_1 (整数 整数; 半角空白区切り)

2行目 三角形の第2の頂点の座標 x_2 y_2 (整数 整数; 半角空白区切り)

3行目 三角形の第3の頂点の座標 x_3 y_3 (整数 整数; 半角空白区切り)

4行目 円の中心の座標 x_4 y_4 (整数 整数; 半角空白区切り)

5行目 円の半径 r (整数)

出力

入力データセットごとに以下の判定結果を出力します。

円が三角形に含まれる場合 a

三角形が円に含まれる場合 b

それ以外の場合で、共通部分がある場合には c

共通部分がない場合には d

入力例	出力例
1 1	b
3 1	c
3 3	d
3 2	a
3	
3 12	
9 3	
11 12	
8 7	
5	
15 3	
17 7	
22 5	
7 6	
4	
6 11	
8 2	
16 9	
10 8	
2	
0 0	

問題07 カードの総和(8点)

整数が書いてあるカードが何枚か入っている袋を使ってゲームをしましょう。各回のゲームで参加者はまず、好きな数 n を一つ宣言します。そして、袋の中から適当な枚数だけカードを一度に取り出して、それらのカードに書かれた数の総和が n に等しければ豪華賞品がもらえます。なお、それぞれのゲーム終了後カードは袋に戻されます。

袋の中の m 種類のカードの情報および、 g 回のゲームで参加者が宣言した数を入力とし、それぞれのゲームで豪華商品をもたらえるカードの組み合わせが何通りあるかを出力して終了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 カードの種類 m (1以上7以下の整数)
2行目 第1種類目のカード情報 a_1 b_1 (整数 整数; 半角空白区切り)
 a_1 : カードに書かれた数字 (1以上100以下の整数)
 b_1 : 枚数 (1以上10以下の整数)
3行目 第2種類目のカード情報 a_2 b_2 (整数 整数; 半角空白区切り)
 :
 $m+1$ 行目 第 m 種類目のカード情報 a_m b_m (整数 整数; 半角空白区切り)
 $m+2$ 行目 ゲームの回数 g (1以上10以下の整数)
 $m+3$ 行目 ゲーム1で宣言された数 n_1 (1以上1,000以下の整数)
 $m+4$ 行目 ゲーム2で宣言された数 n_2 (1以上1,000以下の整数)
 :
 $m+g+2$ 行目 ゲーム g で宣言された数 n_g (1以上1,000以下の整数)

出力

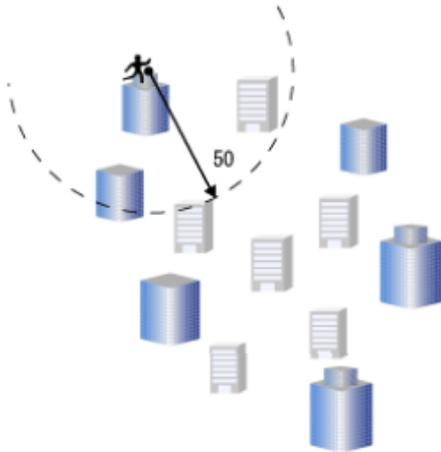
入力データセットごとに次の形式で出力します。

1行目 ゲーム1で豪華賞品がもらえるカード組み合わせ数 (整数)
2行目 ゲーム2で豪華賞品がもらえるカード組み合わせ数 (整数)
 :
 g 行目 ゲーム g で豪華賞品がもらえるカード組み合わせ数 (整数)

入力例	出力例
5	16
1 10	0
5 3	12
10 3	7
25 2	9789
50 2	13658
4	17466
120	
500	
100	
168	
7	
1 10	
3 10	
5 10	
10 10	
25 10	
50 10	
100 10	
3	
452	
574	
787	
0	

問題08 スパイダー人(18点)

正義のヒーロー「スパイダー人 (スパイダージン)」は、腕からロープを出してビルからビルへ飛び移ることができます。しかし、ロープが短いので自分からの距離が50以下のビルにしか移動できません。それより遠くのビルに移動するには、一旦別のビルに飛び移らなくてはなりません。



ビルの数 n 、及び n 個のビルの情報、スパイダー人の移動開始位置及び目的地を入力とし、その移動の最短経路を出力して終了するプログラムを作成してください。

どのようにビルを経由しても目標のビルに移動できない場合はNAと出力してください。

各ビルは点として扱い、最短距離で移動するビルの経由方法が2つ以上存在することはないものとします。

入力として与えられる整数はすべて1以上1,000以下の値とします。

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 ビルの数 n (整数)
2行目 第1のビル情報 $b1\ x1\ y1$ (それぞれ整数; 半角空白区切り)
 $b1$: ビル番号
 $x1$: ビルの x 座標
 $y1$: ビルの y 座標
3行目 第2のビル情報 $b2\ x2\ y2$ (それぞれ整数; 半角空白区切り)
 :
 $n+1$ 行目 第 n のビル情報 $b_n\ x_n\ y_n$ (それぞれ整数; 半角空白区切り)
 $n+2$ 行目 移動情報の個数 m (整数)
 $n+3$ 行目 第1の移動情報 $s1\ g1$ (それぞれ整数; 半角空白区切り)
 $s1$ スパイダー人の移動開始ビル番号
 $g1$ スパイダー人の目的地ビル番号
 $n+4$ 行目 第2の移動情報 $s2\ g2$ (それぞれ整数; 半角空白区切り)
 :
 :

n+m+2行目 第m の移動情報 sm gm (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次の形式で出力します。

1行目 第1の移動情報に対する経路 s1 br11 br12 . . . g1 (それぞれ整数 ; 半角空白区切り) または
NA

s1 : スパイダー人の移動開始ビル番号

br11 : 第1の経由ビル

br12 : 第2の経由ビル

:

g1 : スパイダー人の目的地ビル番号

2行目 第2の移動情報に対する経路 s2 br21 br22 . . . g2 (それぞれ整数 ; 半角空白区切り) または
NA

:

m行目 第mの移動情報に対する経路 sm brm1 brm2 . . . gm (それぞれ整数 ; 半角空白区切り) または
NA

入力例	出力例
4	1 2 3
1 0 0	NA
2 30 0	1 3 9 20 11 6 22
3 60 40	
4 0 60	
2	
1 3	
1 4	
22	
1 0 0	
2 150 40	
3 30 20	
4 180 150	
5 40 80	
6 130 130	
7 72 28	
8 172 118	
9 50 50	
10 160 82	
11 90 105	
12 144 131	
13 130 64	
14 80 140	
15 38 117	
16 190 90	
17 60 100	
18 100 70	
19 130 100	
20 71 69	
21 200 110	
22 120 150	
1	
1 22	
0	

問題09 お城の堀(18点)

いま、忍者が城外から天守閣に忍び入ろうと計画を立てています。この忍者は地面を走ったり、堀の中を泳いだりすることは難なくできますが、堀から這い上がることはとても苦手なので、忍者は堀に入る回数をできるだけ少なくしたいと考えています。

お城の見取図を入力とし、城外から天守閣に至るまでに、堀から這い上がらなくてはならない最小回数を出力して終了するプログラムを作成して下さい。

お城の見取図は二次元格子として与えられます。見取り図に描かれた記号には、天守閣の位置を示す「&」と堀の位置を示す「#」の位置が記され、そのほかの地点には「.」（半角ピリオド）が記されています。

なお、お城には天守閣は一つだけあるものとし、忍者は走ったり泳いだりして移動するときに東西南北方向に1マスずつ移動するものとし、斜めには移動しません。

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2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 見取図の東西の幅n 南北の幅m (整数 整数; 半角空白区切り)
(n, m の値はそれぞれ1以上100以下とする)
- 2行目 見取図の1行目の情報
(記号「&」、「#」、「.」（半角ピリオド）からなる長さn文字の文字列)
- 3行目 見取図の2行目の情報
- ⋮
- m+1行目 見取図のm行目の情報

出力

入力データセットごとに堀から這い上がらなくてはならない最小回数 (整数) を出力します。

入力例	出力例
<pre> 5 5 .###. #...# #.&.# #...# .###. 18 15 ..####....####.... #####.#####...#### #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## #.#####.## ##### ##### 9 10 ###### #####.# #...#.# #.#.#.# #.#&#.#.# #.#...#.# #.#.#.# #.#.#.# #####.# #.#.#.# ##### 9 3 ###...### #.#.&.#.# ###...### 0 0 </pre>	<pre> 1 2 0 0 </pre>

問題10 マトリョーシカ(18点)

マトリョーシカとは女性像をかたどった木製の人形で、ロシアの代表的な民芸品です。マトリョーシカは、大きな人形の中にそれより小さな人形が入っている入れ子構造になっており、大きさの異なる複数の人形で構成されています。このような入れ子構造にするため、各人形の胴体は上下で分割できる筒状の構造になっています。マトリョーシカは職人の手で手作りされるため、一つ一つの人形は世界に一つしかない非常に貴重なものになります。



幼稚園に通う1歳離れた兄弟である一郎君と次郎君は、マトリョーシカで遊ぶのが大好きで、各自がそれぞれ1組のマトリョーシカを持っていました。一郎君のマトリョーシカは n 個の人形から構成されており、次郎君のマトリョーシカは m 個の人形から構成されています。

ある日、好奇心が旺盛な一郎君は、これら2組のマトリョーシカに含まれる人形たちを組み合わせ、より多くの人形を含む新たなマトリョーシカを作れないかと考えました。つまり、 $(n + m)$ 個の人形を使い、 k 個の人形からなる1組のマトリョーシカを作を試みたのです。 n と m の大きい方よりも k を大きくすることができれば、一郎君の目的は達成されます。

兄弟は2人仲良く、どのように人形を組み合わせれば k の値を最大にできるかを考えました。しかし、幼い2人にとってこの問題はあまりにも難しいので、年上のあなたはプログラムを作成して弟たちを助けることにしました。

一郎君と次郎君のマトリョーシカの人形の情報を入力とし、新たなマトリョーシカが含む人形の数 k を出力して終了するプログラムを作成して下さい。

入力される人形に大きさが同じものは存在しません。また、人形を高さ h 半径 r の円柱とみなした場合、高さ h 、半径 r の人形が含むことのできる人形は $x < h$ かつ $y < r$ を満たす高さ x 半径 y の人形です。

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 一郎君のマトリョーシカの人形の数 n (100以下の整数)

2行目 一郎君の第一の人形の情報 $h_1 r_1$ (整数 整数; 半角空白区切り)

h_1 : 一郎君の第一の人形の高さ (1,000未満の整数)

r_1 : 一郎君の第一の人形の半径 (1,000未満の整数)

3行目 一郎君の第二の人形の情報 $h_2 r_2$ (整数 整数; 半角空白区切り)

:

$n+1$ 行目 一郎君の第 n の人形の情報 $h_n r_n$ (整数 整数; 半角空白区切り)

$n+2$ 行目 二郎君のマトリョーシカの人形の数 m (100以下の整数)

$n+3$ 行目 二郎君の第一の人形の情報 $h_1 r_1$ (整数 整数; 半角空白区切り)

h_1 : 二郎君の第一の人形の高さ (1,000未満の整数)

r_1 : 二郎君の第一の人形の半径 (1,000未満の整数)

$n+4$ 行目 二郎君の第二の人形の情報 $h_2 r_2$ (整数 整数; 半角空白区切り)

:

$n+m+2$ 行目 二郎君の第 m の人形の情報 $h_m r_m$ (整数 整数; 半角空白区切り)

出力

入力データセットごとに新たなマトリョーシカが含む人形の数k（整数）を出力します。

入力例	出力例
6	9
1 1	6
4 3	8
6 5	
8 6	
10 10	
14 14	
5	
2 2	
5 4	
6 6	
9 8	
15 10	
4	
1 1	
4 3	
6 5	
8 6	
3	
2 2	
5 4	
6 6	
4	
1 1	
4 3	
6 5	
8 6	
4	
10 10	
12 11	
18 15	
24 20	
0	