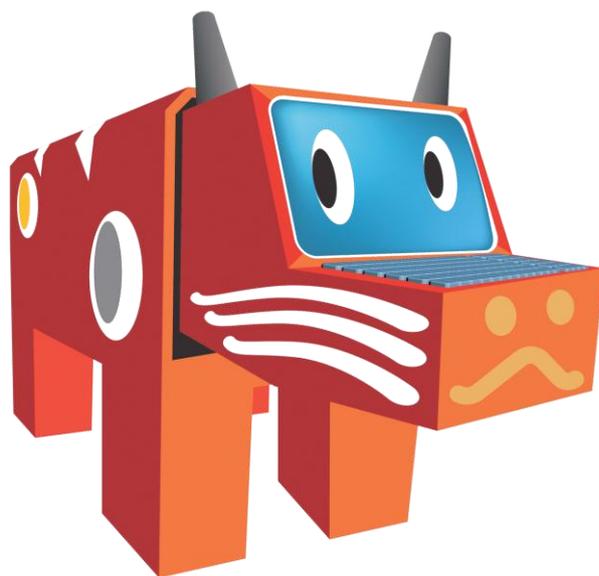


に・ゼロ・いち・に パソコン甲子園2012

全国高等学校パソコンコンクール

プログラミング部門 予選問題

平成24年 9月 8日(土) 午後1時30分～午後4時30分



10th
Anniversary

全国高等学校パソコンコンクール実行委員会

問題 1 10 問解いたら何点取れる？

(4 点)

選手の皆さん、パソコン甲子園によろこそ。パソコン甲子園は今年で 10 周年になりますが、出題される問題数や合計得点は年によって異なります。各問題には難易度に応じて得点が決まっています。問題数が 10 問で、それぞれの問題の得点が与えられるとき、それらの合計を出力するプログラムを作成して下さい。

入力

入力は以下の形式で与えられる。

s_1
s_2
·
·
s_{10}

入力は10行からなり、 i 行目に問題 i の得点を表す整数 s_i ($0 \leq s_i \leq 100$) が与えられる。

出力

合計得点を 1 行に出力する。

入出力例

入力例 1	出力例 1
1	55
2	
3	
4	
5	
6	
7	
8	
9	
10	

入力例 2	出力例 2
4	100
4	
8	
8	
8	
10	
10	
12	
16	
20	

問題 2 乗車券

(4 点)

新幹線に乗るためには、「乗車券」「特急券」の2枚の切符が必要です。経路の一部で新幹線を利用しない場合があるため、これらは別々の切符となっていますが、新幹線のみを利用する経路では、1枚で乗車券と特急券を兼ねる「乗車・特急券」が発行されることもあります。

自動改札機では、これらの切符を読み込んで、正しい切符が投入されたときだけゲートを開けなければなりません。「乗車券」と「特急券」それぞれ1枚、または、その両方、または、「乗車・特急券」が1枚投入されたかどうかを判定し、自動改札機の扉の開閉を判断するプログラムを作成して下さい。

入力

入力は以下の形式で与えられる。

b_1 b_2 b_3

入力は1行からなり、1つの空白で区切られた3つの整数を含む。 b_1 は「乗車券」の投入状態、 b_2 は「特急券」の投入状態、 b_3 は「乗車・特急券」の投入状態を示す。投入状態は0または1で表され、0の場合は投入されていない状態、1の場合は投入されている状態を表す。ただし、想定される投入状態の組み合わせは以下の場合のみとする。

入力	投入状態	投入に対する扉の動作
1 0 0	「乗車券」のみ投入	Close
0 1 0	「特急券」のみ投入	Close
1 1 0	「乗車券」と「特急券」投入	Open
0 0 1	「乗車・特急券」投入	Open
0 0 0	投入なし	Close

出力

自動改札機の開閉を表す Open または Close を1行に出力する。

入出力例

入力例 1	出力例 1
0 0 1	Open
入力例 2	出力例 2
0 1 0	Close

問題3 家庭菜園

(8点)

家庭菜園に野菜を植えることにしました。n粒の種があったので1日に1粒ずつ、n日かけてn粒の種をまきました。どの種からも芽が出て、すくすくと育っています。収穫の時期が待ち遠しいものです。

ある日、いつものように苗に水やりをしていると、おかしなことに気づきました。n本の野菜の苗があるはずなのに、1本多いのです。雑草が生えてきてしまいました。直ちに引っこ抜きたいのですが、困ったことにどの苗もよく似ていて、野菜と雑草の見分けがつきません。

手がかりになるのは、野菜の成長速度です。この野菜は、種をまいてからずっと、1日に決まった長さだけ伸び続けるのです。しかし、この「決まった長さ」というのが何センチメートルなのかはわかりません。また、最初の種を何日前にまいたのかも忘れてしまいました。苗は一列に並んでいますが、唯一覚えているのは、種をまくとき毎日一粒ずつ右から順にまいていったことだけです。

n + 1本の苗の長さを入力し、雑草の長さを出力するプログラムを作成して下さい。

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
h1 h2 h3 ... hn+1
```

1行目のn ($4 \leq n \leq 100$)は野菜の苗の数を表す整数である。2行目は1つの空白で区切られたn+1個の整数を含み、 h_i ($1 \leq h_i \leq 10^9$)は左からi番目の苗の長さを示す。

データセットの数は500を超えない。

出力

各データセットごとに雑草の長さを1行に出力する。

入出力例

入力例	出力例
5	6
1 2 3 6 4 5	1
6	12
1 3 6 9 12 15 18	
4	
5 7 9 11 12	
0	

※ 「パソコン甲子園2012予選プログラミング部門解説」の“問3 家庭菜園 お知らせ”にて、上記の問題文を補足する情報を掲載していますので、あわせてご確認ください。

問題 4 すべての数は 6 1 7 4 に通ず

(8 点)

0 から 9 の数字からなる四桁の数 N に対して以下の操作を行う。

1. N の桁それぞれの数値を大きい順に並べた結果得た数を L とする
2. N の桁それぞれの数値を小さい順に並べた結果得た数を S とする
3. 差 $L-S$ を新しい N とする (1 回分の操作終了)
4. 新しい N に対して 1. から繰り返す

このとき、全桁が同じ数字 (0000, 1111 など) である場合を除き、あらゆる四桁の数はいつかは 6174 になることが知られている。例えば $N = 2012$ の場合

1 回目 ($N=2012$): $L = 2210$, $S = 0122$, $L-S = 2088$

2 回目 ($N=2088$): $L = 8820$, $S = 0288$, $L-S = 8532$

3 回目 ($N=8532$): $L = 8532$, $S = 2358$, $L-S = 6174$

となり、3 回の操作で 6174 に到達する。

0 から 9 の数字からなる四桁の数が与えられたとき、何回の操作で 6174 に到達するか計算するプログラムを作成して下さい。

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 0000 が 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

データセットは 1 行であり、 $N(1 \leq N \leq 9999)$ は四桁の数を示す。 $N < 1000$ の場合は上の桁は 0 で埋められている。

データセットの数は 10000 を超えない。

出力

各データセットごとに何回の操作で 6174 に到達したかを 1 行に出力する。ただし全桁が同じ数字である数が入力として与えられた場合は NA と出力する。

入出力例

入力例	出力例
6174	0
2012	3
3333	NA
0000	

問題5 パイプつなぎ職人の給料

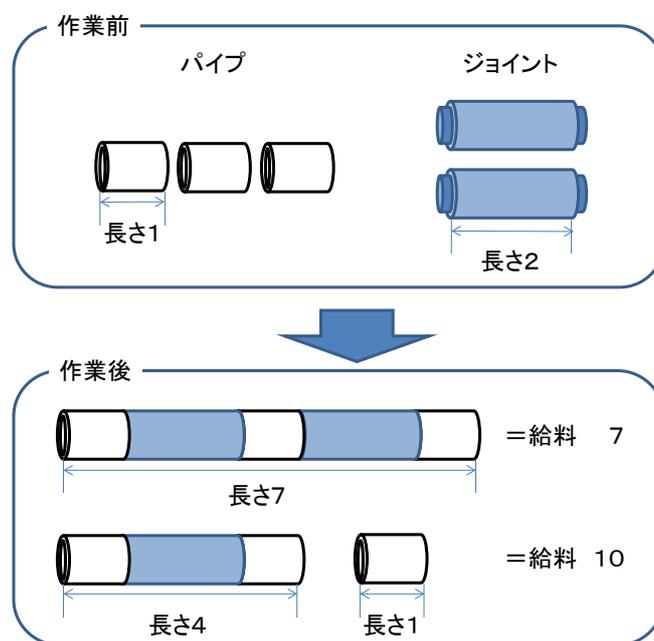
(8点)

ぼくはパイプつなぎ職人です。パイプをつなぐジョイントとパイプさえもらえれば、どんなパイプだっ
てつないでみせます。ぼくは毎日、親方からパイプとジョイントを渡されて、それをつないで親方に渡
します。でも、パイプの数が多すぎるときは、1日でそれを全部つなげることはできません。そんなと
きでも親方はにっこり笑って、ぼくに給料を渡してくれます。

ところで、あるとき変なことに気がつきました。全部のパイプをつなげられたときより、つなげられな
かったときの方が給料が多いことがしょっちゅうあるんです。あんまり変なので、ある日、親方が来た
ときに、給料の計算方法が書いてあるメモをこっそり見ちゃいました。そしたら、なんと

給料は「パイプの本数×パイプの長さの総和」で支払う。
ただし、ジョイントでつなげて、ひとつながりになったものは、それ
を1本のパイプとみなす。

って書いてありました。これで全部つなげた方が給料が安くなることがある理由がわかりました。たと
えば下図のように、長さ1のパイプ3本と長さ2のジョイント2本を全部つなげると長さ $1+2+1+2+1=7$
のパイプが1本できるので、 $1\times(7)=7$ です。でも、ジョイントを一つだけ使って長さ $1+2+1=4$ のパイ
プと長さ1のパイプの2本にすると $2\times(4+1)=10$ なので、全部つなげるよりいっぱい給料がもらえます。



親方がなんでこんな方法で給料を決めてるかわからないけど、これでぼくもどうすればもらえる給料を
多くできるかわかりました！

それでは、パイプの本数が与えられたとき、もらえる給料の最大の金額を計算するプログラムを作成し
てください。

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
p1 ... pn
j1 ... jn-1
```

1行目にパイプの本数 n ($2 \leq n \leq 65000$)が与えられる。2行目は1つの空白で区切られた n 個の整数からなる。 p_i ($1 \leq p_i \leq 1000$)は i 番目のパイプの長さを示す。3行目は1つの空白で区切られた $n-1$ 個の整数からなる。 j_i ($1 \leq j_i \leq 1000$)は i 番目のジョイントの長さを示す。

i 番目のジョイントは、 i 番目と $i+1$ 番目のパイプだけをつなげることができる。つなげたパイプの長さは、 $p_i + j_i + p_{i+1}$ になる。

データセットの数は100を超えない。

出力

各データセットごとに、得られる給料の最大の金額を1行に出力する。入力として与えられるデータセットでは、出力される値は必ず32ビット符号無し整数の範囲に収まるものとする。

入出力例

入力例	出力例
3	12
1 1 1	48
3 3	76
4	
3 3 3 3	
1 1 1	
5	
1 2 3 4 5	
4 3 2 1	
0	

問題6 マヤの大予言

(10点)

真也君はテレビで「マヤの大予言！2012年で世界が終る？」という番組を見ました。結局、世界が終るかどうかはよくわかりませんが、番組で紹介されていたマヤの「長期暦」という暦に興味を持ちました。その番組では以下のような説明をしていました。

マヤ長期暦は、右の表のような単位からなる、全部で13バクトゥン(187万2000日)で構成される非常に長い暦である。ある計算法では、この暦は紀元前3114年8月11日に始まり2012年12月21日に終わると考えられていて、このため今年の12月21日で世界が終るとい説が唱えられている。しかし、13バクトゥンで1サイクルとなり、今の暦が終わったら新しいサイクルに入るだけという考えもある。

1 キン = 1 日
1 ウィナル = 20 キン
1 トゥン = 18 ウィナル
1 カトゥン = 20 トゥン
1 バクトゥン = 20 カトゥン

「ぼくの二十歳の誕生日はマヤ暦だと何日になるのかな？」真也君はいろいろな日をマヤ長期暦で表してみたくまりました。

では、真也君の代わりに、西暦とマヤ長期暦とを相互変換するプログラムを作成してください。

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#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。

b.ka.t.w.ki

または

y.m.d

データセットは1つの文字列を含む1行からなる。b.ka.t.w.ki はマヤ長期暦の日付、y.m.d は西暦の日付である。与えられる単位は以下の通りである。

マヤ長期暦

b	バクトゥン	$(0 \leq b < 13)$
ka	カトゥン	$(0 \leq ka < 20)$
t	トゥン	$(0 \leq t < 20)$
w	ウィナル	$(0 \leq w < 18)$
ki	キン	$(0 \leq ki < 20)$

西暦

y	年	$(2012 \leq y \leq 10,000,000)$
m	月	$(1 \leq m \leq 12)$
d	日	$(1 \leq d \leq 31)$

西暦の日の最大値は、大の月、小の月、うるう年かどうかで変わる(うるう年は4の倍数の年のうち、100で割り切れない年か、400で割り切れる年である)。マヤ長期暦の日付の範囲は0.0.0.0.0から12.19.19.17.19までとする。ただし、マヤ長期暦の0.0.0.0.0は西暦の2012.12.21に対応する。また、西暦の日付の範囲は2012.12.21から10000000.12.31までとする。

データセットの数は500を超えない。

出力

入力が西暦のときはマヤ長期暦を、マヤ長期暦のときは西暦を、それぞれ入力と同じ形式で出力する。入力された西暦を換算した結果、マヤ長期暦の次のサイクルに入った場合でも b.ka.t.w.ki の形式で出力してよい。

入出力例

入力例	出力例
2012.12.31	0.0.0.0.10
2.12.16.14.14	3054.8.15
7138.5.13	0.0.0.0.10
10.5.2.1.5	6056.2.29
10000000.12.31	8.19.3.13.2
#	

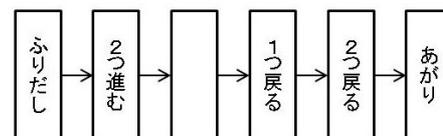
問題7 すごろくを作る

(10点)

太郎君は、子供会の催しでみんなで遊べるようにすごろくを作りました。ゲームをおもしろくするために、「ふりだし」と「あがり」以外のすごろくのマスいくつかに「6つ進む」、「5つ戻る」のように指示を書き込んでいきました。ルーレットを回して出た数だけ進み、止まったマスに指示が書き込んであれば、その指示に従って移動します。ただし、指示に従って進んだ先のマスの指示には従いません。

ルーレットは1からある数までの間の数を等確率で出すことができますものとしします。また、「あがり」に達するより大きな数が出たときや、指示に従うと「あがり」より先に進んでしまうときは「あがり」に移動します。指示に従って戻るときに「ふりだし」より前に戻ってしまうときは「ふりだし」に戻ることにします。

ところが、ルーレットとマスの指示によっては「あがり」にたどりつけない場合が出てきてしまいます。たとえば、図のようなすごろくを作ったとしましょう。1と2しか出ないルーレットを使うと、1, 2の順に出れば「あがり」に行けますが、はじめに2が出たらその後は何が出ても永久に「あがり」にはたどり着けません。太郎君は、そうとは知らずに調子に乗ってあちこちのマスに指示を書き込んでしまいました。



そこで、太郎君に代わって、ルーレットとマスの指示によっては、「あがり」にたどり着けない場合が生じるかどうか判定するプログラムを作成してください。

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。

```
max
n
d1
d2
.
.
.
dn
```

1行目にルーレットが出す数の最大値 \max ($2 \leq \max \leq 250$) が与えられ、2行目に「ふりだし」と「あがり」以外のマスの数 n ($2 \leq n \leq 250$) が与えられる。続く n 行に各マスの指示を表す数 d_i ($-n \leq d_i \leq n$) が与えられる。 d_i がゼロのときは指示が書いていないことを、正の数ときは $|d_i|$ 進む指示を、負の数ときは $|d_i|$ 戻る指示を表す（ここで、 $|x|$ は x の絶対値を表す）。入力される値はすべて整数である。

データセットの数は100を超えない。

出力

各データセットごとに判定結果を1行に出力する。ルーレットとマスの指示によっては、「あがり」にたどり着けない場合が生じるときは「NG」、そうでなければ「OK」を出力する。

入出力例

入力例	出力例
3	OK
3	NG
-2	NG
1	
0	
2	
4	
2	
0	
-1	
-2	
2	
2	
-2	
-2	
0	

問題 8 ビートパネル

(12 点)

遊太君は、近所のゲームセンターで人気のゲーム「ビートパネル」にはまっています。このゲームは図のようなグリッド状に並べられた 4×4 の計 16 個のパネル型のボタンからなります。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

図のように、左上から右下にボタン 1, ボタン 2, …, ボタン 16 の順にボタンが並んでいます。ゲームでは一定間隔でビート音が鳴り最後に終了音が鳴ります。ビート音が鳴ると同時に複数のボタンが光ります。1 個も光らない場合もあります。プレイヤーは、ビート音が鳴った直後から次の音が鳴るまでの間に両手の指を使って複数のボタンを 1 度だけ同時に押すことができます。何も押さないということも可能です。終了音が鳴ったと同時にゲームは終了します。

遊太君は c 通りの押し方を習得しており、ビート音が鳴る度に、それらの押し方の中から 1 つ決めてボタンを押します。押したボタンが光っていれば、それらのボタンの光は消え、消えたボタンの数がプレイヤーのスコアに加算されます。また、一度光ったボタンはそのボタンが押されるまで光は消えることはありません。

ビート音が n 回鳴るときのボタンの光り方と遊太君が習得している c 通りのボタンの押し方を入力とし、遊太君が獲得することのできる得点の最大値を出力するプログラムを作成してください。

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 2 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 c
a1,1 a1,2 … a1,16
a2,1 a2,2 … a2,16
…
an,1 an,2 … an,16
b1,1 b1,2 … b1,16
b2,1 b2,2 … b2,16
…
bc,1 bc,2 … bc,16
```

1 行目は 1 つの空白で区切られた 2 つの整数からなる。 n ($1 \leq n \leq 30$) はビート音の数、 c ($1 \leq c \leq 30$) は習得しているボタンの押し方の数を示す。続く $n+c$ 行にボタンの光り方とボタンの押し方が与えられる。行中の入力項目の区切りは 1 つの空白である。 $a_{k,i}$ は k 回目のビート音が鳴ったときのボタン i の光り方を、 $b_{j,i}$ は c 通り中 j 番目の押し方でボタン i が押されるかどうかを表す。 $a_{k,i}$ の値は 0 が「光らない」、1 が「光る」を表す。 $b_{j,i}$ の値は 0 が「押さない」、1 が「押す」を表す。

データセットの数は 20 を超えない。

出力

各データセットごとに、得点の最大値を1行に出力する。

入出力例

入力例	出力例
2 2	8
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	7
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	16
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	
2 2	
0 0 1 0 1 1 1 1 0 0 1 0 0 0 1 0	
0 1 0 1 0 1 1 1 0 1 0 1 0 0 0 0	
0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0	
0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0	
5 3	
0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0	
1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1	
0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0	
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0	
0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0	
0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0	
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	
0 0	

問題 9 有限体電卓

(16 点)

0 以外のすべての実数の逆数は実数になりますが、整数の逆数が整数になるとは限りません。C 言語などで $3.0/2.0*2.0 = 3.0$ なのに $3/2*2 = 2$ になってしまうのはこのためです。ですが、素数で割った余りが等しい整数を同じものとみなすと、0 以外のすべての整数が逆数をもつようになります。

整数 x と y を p で割った余りが等しいとき、 $x \equiv y \pmod{p}$ と書きます。 p が素数のとき、このような x と y を同じものとみなすなら、すべての整数 n は 0 から $p-1$ までのどれかの整数 x に対して $x \equiv n \pmod{p}$ となるので、けっきょく $\{0, 1, 2, \dots, p-1\}$ という集合だけからなる世界を考えればよいことがわかります。

この世界での加減乗除は以下のようになります。

- ・ 加算 $x+y$ の値は、 $x+y \equiv z \pmod{p}$ となる 0 から $p-1$ までの数 z になります。
- ・ 減算 $x-y$ の値は、 $y+m \equiv 0 \pmod{p}$ となる 0 から $p-1$ までの数 m (これが $-y$ に相当します) について、 $x-y \equiv x+m \pmod{p}$ となることから得られます。たとえば、 $p=5$ のとき、 $4+1 \equiv 0 \pmod{5}$ となります。このとき、 $2-4 \equiv 2+1 \equiv 3 \pmod{5}$ より、 $2-4$ の値は 3 になります。
- ・ 乗算 $x*y$ の値は、 $x*y \equiv z \pmod{p}$ となる 0 から $p-1$ までの数 z になります。
- ・ 除算 x/y の値は、 $y*d \equiv 1 \pmod{p}$ となる 0 から $p-1$ までの数 d (これが y の逆数になります) について、 $x/y \equiv x*d \pmod{p}$ となることから得られます。たとえば、 $p=5$ のとき、 $2*3 \equiv 1 \pmod{5}$ より 3 は 2 の逆数になります。このとき、 $1/2 \equiv 1*3 \equiv 3 \pmod{5}$ より、 $1/2$ の値は 3 になります。

このようにして加減乗除の四則演算がすべて 0 から $p-1$ の範囲に収まります。このとき、集合 $\{0, 1, \dots, p-1\}$ を p の有限体といいます。この有限体の中で、加減乗除、0 から $p-1$ までの数、カッコを使って 算術式を構成することができます。

p の有限体の 0 以外のすべての要素が逆数をもつことは、フェルマーの小定理と呼ばれる有名な定理 $a^{p-1} \equiv 1 \pmod{p}$ からわかります (ただし、 p は素数、 a と p は互いに素)。 p の有限体のすべての要素 x は p と互いに素なので、この定理から $x*x^{p-2} \equiv 1 \pmod{p}$ となり x^{p-2} が x の逆数になるからです。

では、素数と式が与えられたときに、その素数の有限体で式を計算する電卓プログラムを作成してください。

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 0 の後にコロンが 1 つ続くだけの行 0: で示される。各データセットは以下の形式で与えられる。

```
p:exp
```

データセットは 1 行であり、 p ($2 \leq p \leq 46000$) は素数、 exp は算術式を示す。 exp は加減乗除 (それぞれ $+$, $-$, $*$, $/$) とカッコ、0 から $p-1$ までの数から構成される。演算子、数、カッコなどの前後には 1 つ以上の空白が現れる可能性がある。 exp の長さは 100,000 を超えない。

データセットの数は 1000 を超えない。

出力

データセットごとに、計算結果を出力する。出力形式は以下の通りである。

$$\text{exp} = \text{val} \pmod{p}$$

val は、p の有限体での算術式 exp の計算結果である。ただし、0 での除算が含まれるときは NG と出力する。また、exp に現れる加減乗除、カッコ、数の前後には空白を空けないこと。ただし、= の前後には空白を一つずつ空ける。また、val と開きカッコの間、mod と p の間にも空白を一つずつ空ける。

入出力例

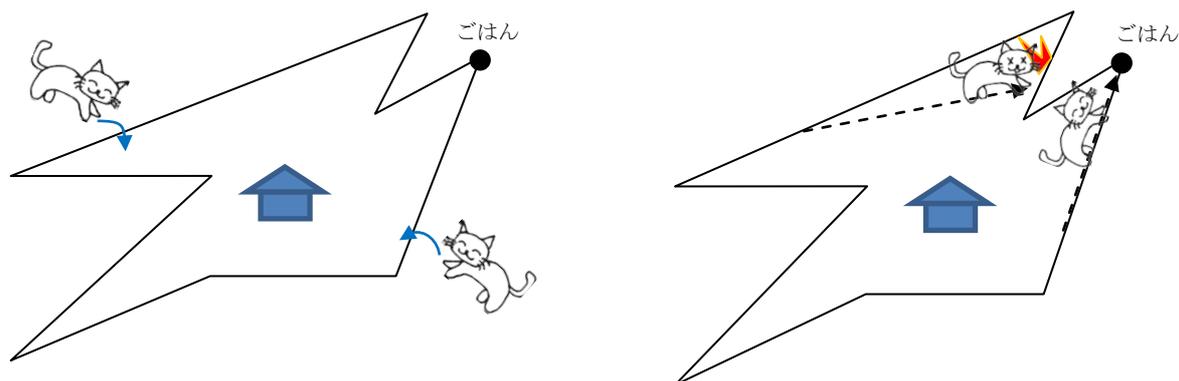
入力例	出力例
5: 2 - 3	2-3 = 4 (mod 5)
17: 1 + 3 * (2 + 3 / 5 * 2) + 7	1+3*(2+3/5*2)+7 = 4 (mod 17)
11: 1 / 8 - 5 - 8 * 2	1/8-5-8*2 = 8 (mod 11)
19: 8 / (2 - 3 * 7)	NG
1153: 10 * 3 / 7 + (50 + 81 / 22) + 11	10*3/7+(50+81/22)+11 = 915 (mod 1153)
0:	

問題 10 ねこまっしぐら

(20 点)

あるところに、高い塀に囲まれた大きなお屋敷がありました。そのお屋敷の主人は猫がとても大好きで、時折やってくる猫たちのために、いつもおいしいごはんを用意していました。お腹を空かせた猫たちは、高い塀をひょいと飛び越え、お屋敷の至る所に置いてあるごはんにまっしぐらに駆け寄るのでした。

ある日主人は、屋敷の中で何匹かの猫が倒れているのを見つけました。猫たちはごはんを探して屋敷の中を縦横無尽に走り回ったので、ぶつかったり転んだりしてしまったのです。主人は猫たちの安全を考えて、ごはんの置き場所を工夫することにしました。



上空から見ると、このお屋敷の塀は多角形をなしています。主人は猫たちがごはんを見つけやすいように、敷地内の中でも、多角形の頂点にあたる場所だけにごはんを置くことにしました。また猫たちは気まぐれなので、多角形の周上のどの点からお屋敷に入ってくるかは予想できません。そこで主人は、猫がどの点から入ったとしても、その点から直進していずれかのごはんにたどり着けるように、ごはんを配置することも決めました。

すべての頂点にごはんを配置すれば、この条件を満たすことができます。しかし、ごはんを補充して回るのは大変なので、主人はできるだけ少ない数の頂点にごはんを配置したいと思いました。さて、主人は最低何箇所にごはんを配置する必要がありますでしょうか。

お屋敷の塀を表す多角形を入力とし、ごはんを配置する頂点の数の最小値を求めるプログラムを作成して下さい。ただし、猫は多角形の内部のみを直進できるものとします（辺は多角形の内部に含めるものとします）。

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
x1 y1
x2 y2
...
xn yn
```

1 行目に多角形の頂点の数 n ($3 \leq n \leq 16$) が与えられる。続く n 行に多角形の頂点の座標が与えられる。 n

行のそれぞれは、1つの空白で区切られた2つの整数からなる。 x_i ($-1000 \leq x_i \leq 1000$)は*i*番目の頂点の*x*座標、 y_i ($-1000 \leq y_i \leq 1000$) は*i*番目の頂点の*y*座標を示す。多角形の頂点は、隣り合った頂点を反時計回りに訪問するような順番で与えられる。

データセットの数は20を超えない。

出力

各データセットごとに、ご飯を配置する頂点の数を1行に出力する。

入出力例

入力例	出力例
8	1
0 0	2
3 2	
6 2	
8 6	
6 5	
7 7	
0 4	
3 4	
8	
0 0	
5 3	
5 2	
4 1	
6 1	
8 6	
6 4	
2 4	
0	