2008 National Collegiate Programming Contest

- Problems: There are 7 problems (15 pages in all, not count- ing this cover page) in this packet.
- Problem Input: Input to the problems are through the input files. Input filenames are given in the table below. Each input file may contain one or more test cases. Test cases may be separated by any delimiter as specified in the problem statements.
- Problem Output: All output should be directed to standard output (screen output).
- Time Limit: The judges will run each submitted program with certain time limit (given in the table below).

Table 1: Problem Information Sheet

	Problem Name	Input File	Time Limit
Problem A	Summation Formula	pa.in	3 sec.
Problem B	Divide Game	pb.in	5 sec.
Problem C	Sierpinski's Triangle	pc.in	5 sec.
Problem D	Maximum Sum of Non-overlapping Rectangles	pd.in	3 sec.
Problem E	Generators of A Cyclic Group	pe.in	3 secs.
Problem F	Triangle and Diamond	pf.in	5 secs.
Problem G	Postage stamp problem	py.in	120 secs.

Problem A
Summation Formula
Input File: *pa.in*
Time Limit: *3 sec.*

Most of you remember the following formula:
$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$

$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + n$

How about $\sum_{i=1}^{n} i^3, \sum_{i=1}^{n} i^4, ..., \sum_{i=1}^{n} i^k$?

There is a simple way to derive the closed form formula of these summations. Let's consider the case for $k = 2$.

$$\begin{array}{rrcrcrcrcr}
 & n^3 & - & (n-1)^3 & = & 3n^2 & - & 3n & + & 1 \\
 & (n-1)^3 & - & (n-2)^3 & = & 3(n-1)^2 & - & 3(n-1) & + & 1 \\
 & ... & & & & & & & & \\
 & 2^3 & - & 1^3 & = & 3(2)^2 & - & 3(2) & + & 1 \\
+ & 1^3 & - & 0^3 & = & 3(1)^2 & - & 3(1) & + & 1 \\
\hline
 & n^3 & & & = & 3\sum_{i=1}^{n} i^2 & - & 3\sum_{i=1}^{n} i & + & n
\end{array}$$

Note on the righthand side, most terms are canceled, we get $n^3 = 3\sum_{i=1}^{n} i^2 - 3\sum_{i=1}^{n} i + n$. Substitute $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. We get $3\sum_{i=1}^{n} i^2 = n^3 + 3\frac{n(n+1)}{2} - n$, i.e., $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + n$. You are to write a program to compute the formula with a given k $(1 \leq k \leq 20)$. The input for each test case contains two numbers, k, d, and the output should be the coefficient of the n^d terms of the polynomial for $\sum_{i=1}^{n} i^k$. Since the coefficients are all rational numbers, therefore the final output should be a fractional number in its irreducible form. You might need Pascal's triangle to get the coefficients of $(n-1)^k$.

Input File Format

The first line of the input file contains an integer indicating the number of test cases to follow. Each test case contains two integers k, d in a single line.

Output Format

For each test case output an integer if the coefficient is an integer otherwise output the numerator and denominator of the coefficient in irreducible form in one line.

Sample Input

```
5
3 3
3 0
7 6
17 2
17 15
```

Output for the Sample Input

```
1 2
0
7 12
-3617 60
0
```

Problem B
Divide Game
Input File: *pb.in*
Time Limit: *5 sec.*

The rule of playing a divide game is as follows: The game begins with M ($M \leq 12$ and M is even) non-zero integers. In each round, you can choose arbitrary way to arrange these integers into pairs from left to right. In a pair, the bigger integer is divided by the smaller one to get a remainder. The sum of the remainders is the points you get in each round.
The game continues using the remainders as input for the next round. if zero is produced in a round, it is discarded. After zeros are discarded, If the number of non-zero integers is odd, the rightmost integer is discarded so that an even number of non-zero integers will be ready for next round. This game repeats until no more pairs can be formed to perform any division. The final score of a play is the sum of the scores in each round.
For example, let (1 2 3 4 5 6) be the input. In one play, integers are paired as (1 2) (4 6) (3 5). The divide game can be played as follows:

```
(1 2) (4 6) (3 5)  -> First round
  0     2     2    -> you get 4 points.
 (2     2)         -> Second round
    0              -> you get 0 points.
Final score = 4 + 0 = 4
```

In another play, the integers are paired as (1 2) (3 4) (5 6), the game can be played as follows:

```
 (1 2) (3 4) (5 6)  -> First round
   0     1     1    -> you got 2 points.
        (1     1)   -> Second round after right most number is discarded
         0          -> you get 0 points.
Final score = 2 + 0 = 2
```

This pairing generates less final score than the previous one.

Given a list of integers, please write a program to compute the maximum final score that can be reached in a given divide game.

Input File Format

The first integer N ($N \leq 9$) is the number of test cases. Each test case begins with M the number of integers in the first line. The second line of a test case contains M integers separated by space.

Output Format

For each test case, please output the maximum final score that can be reached.

Sample Input

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

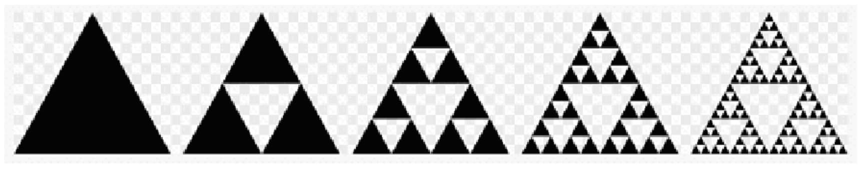
Output for the Sample Input

```
2
4
```

Problem C

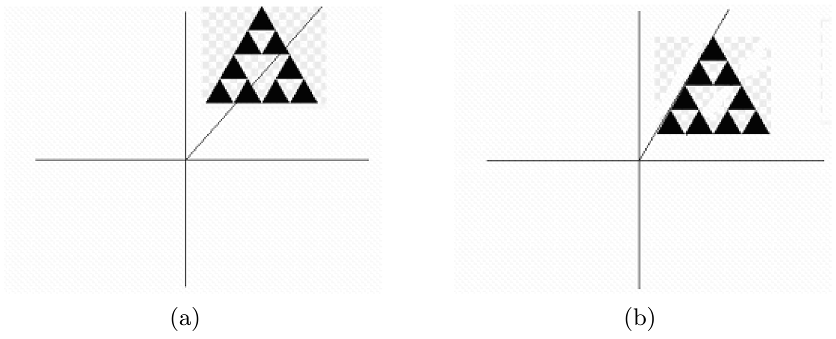
Input File: *pc.in*
Time Limit: *5 sec.*

The interesting shapes in the following are called Sierpinski's Triangle (or gasket), after the Polish mathematician Waclaw Sierpinski who described some of its interesting properties in 1916. Among these is its fractal or self-similar character. The shapes from the left to the right are a large black triangle, a large black triangle consists of three smaller black triangles, each of which itself consists of three smaller black triangles, each of which ..., a process of subdivision which could, with adequate screen resolution, be seen to continue indefinitely. The following shapes list from level 0 to level 4



```
Level 0     level 1     level 2     level3     level 4
```

You are given a Sierpinski's triangle in the plane of a certain level and a ray from the origin and passing through a given point. You are asked to write a program to determine how many black triangles the ray intersects with a Sierpinski's triangle. Note that when the ray intersects with a triangle it means that the ray must intersect two edges of the triangle. For example, in Figure 1a the Sierpinski's triangle starts from an equilateral triangles with level 2 and a ray intersects with two smaller black triangles. In Figure 1b the ray (the angle between the ray and x axis is 60 degree) intersects with four triangles, due to it intersects with two vertices of each triangle



(a) (b)

Figure 1: A ray Intersects with Level 2 Sierpinski's triangle

Input File Format

The input will consist of a number of test cases and their data. The first line contains that a positive number n ($n \leq 9$) indicates how many test cases available. The second line to $n+1^{st}$ line are the information of three vertices and level of Sierpinski's triangle (The number of levels is less than eight) and a given point the ray passes. Each line contains the Sierpinski's triangle given by its three vertices, each is a point in the plane, and a level, which is a positive integer, and the given point the ray passes through. Calculation value that is less than 0.00001 is regarded as zero.

Output Format

The output contains one line for each test case. Each line contains an integer to indicate how many triangles the ray intersects.

Sample Input

```
4
1.0  1.0  1.0  6.0  6.0  11.0  1.0  3 -1.0  1.0
1.0  1.0  1.0  6.0  6.0  11.0  1.0  2  1.0  1.0
1.0  1.0  2.0  6.0  7.0  11.0  2.0  4  1.0  1.5
1.0  1.0  2.0  6.0  7.0  11.0  2.0  4  2.0  1.0
```

Output for the Sample Input

```
0
4
4
5
```

Problem D
Maximum Sum of Non-overlapping Rectangles

Input File: *pd.in*

Time Limit: *3 sec.*

Given a 2-dimensional array of positive and negative integers, you are to write a program to find two non-overlapping sub-rectangles with the largest sum. The sum of two rectangles is the sum of all the elements in the two rectangles. A sub-rectangle is any contiguous sub-array of size 1*1 or greater located within the whole array. Two rectangles are non-overlapping if they have no common element. For example, in the following array,

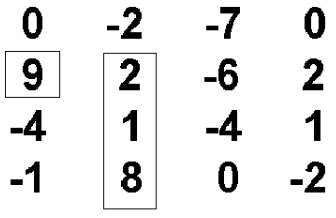
$$\begin{array}{rrrr} 0 & -2 & -7 & 0 \\ 9 & 2 & -6 & 2 \\ -4 & 1 & -4 & 1 \\ -1 & 8 & 0 & -2 \end{array}$$

The rectangles $\begin{array}{|c|c|}\hline 0 & -2 \\ \hline 9 & 2 \\ \hline\end{array}$ and $\begin{array}{|c|}\hline -4 \\ \hline 0 \\ \hline\end{array}$ are non-overlapping, and

$\begin{array}{|c|c|c|}\hline 9 & 2 & -6 \\ \hline\end{array}$ and $\begin{array}{|c|c|c|}\hline -4 & 1 & -4 \\ \hline\end{array}$ are also non-overlapping. But

$\begin{array}{|c|c|c|}\hline 9 & 2 & -6 \\ \hline\end{array}$ and $\begin{array}{|c|c|}\hline -6 & 2 \\ \hline\end{array}$ `are overlapping since they have a common element -6.`

As shown in the next figure, the sum of the two rectangles is 20, which is the maximum of any two non-overlapping rectangles in the array.



Input File Format

The input consists of a number of test cases. Each test case begins with a positive integer n ($n \leq 100$) on a line by itself indicating the size of the square array. This is followed by n^2 integers separated by white-space (newlines and spaces). These n^2 integers make up the array in row-major order (i.e., all numbers on the first row, left-to-right, then all numbers on the second row, left-to-right, etc.). The numbers in the array will be in the range [-127, 127]. The case with n=0 indicate the end of the input.

Output Format

For each test case, your program has to output the maximum sum in a line. You don't need to process the case with n=0.

Sample Input

```
 4
 0 -2  -7  0
 9  2  -6  2
-4  1  -4  1
-1  8   0 -2
 2
-1 -2
-3 -4
 0
```

Output for the Sample Input

```
20
-3
```

Problem E

Generators of A Cyclic Group

Input File: pe.in

Time Limit: 3 seconds

A group is said to be cyclic if it can be generated by a single element. For each of the positive integer, n, consider a cyclic group $Z_n = \{0, 1, 2, \cdots, n-1\}$ under the binary operation $\oplus$ defined as

$$c = a \oplus b \;=\; (a+b)\ mod\ n \;\; \forall\ a, b \in Z_n$$

where c is the remainder of $a + b$ divides n.

In particular, for $g \in Z_n$ and any positive integer k, kg is defined as

$$kg = \underbrace{(g \oplus g \oplus \cdots \oplus g)}_{k}\ mod\ n$$

A generator for Z_n is $g \in Z_n$ such that $\{g, 2g, \cdots, ng\} = Z_n$. This problem is to ask you to write a program to compute $\psi(n)$, the number of elements which can generate the cyclic group for any given positive integer n, where $3 \leq n \leq 500$.

Input File Format

The first line of the input file always contains one integer, K ($K \leq 6$), indicating the number of test cases to come. Each of the following lines is the integer n, based on which, you need to compute $\psi(n)$.

Output Format

For each n, output $\psi(n)$, the number of generators for the group Z_n.

Sample Input

```
6
3
7
10
202
257
451
```

Output for the Sample Input

```
   3     2
   7     6
  10     4
 202   100
 257   256
 451   400
```

Problem F

Triangle and Diamond

Input File: pf.in

Time Limit: 5 seconds

We have an equilateral triangular land on which we like to build diamond-shaped houses. The equilateral triangular land consists of many equilateral triangles that have sides of unit length, and they will be refereed to as "unit equilateral triangles". Two unit equilateral triangles are *adjacent* if and only if they share a common side, and any two adjacent unit equilateral triangles can be used to build a diamond-shaped house. There are unit equilateral triangles that are not suitable for construction, which will be referred to as "bad" triangles. Now given the locations of these bad triangles, please determine the maximum number of houses that can be built in this equilateral triangular land. Houses cannot overlap each other.

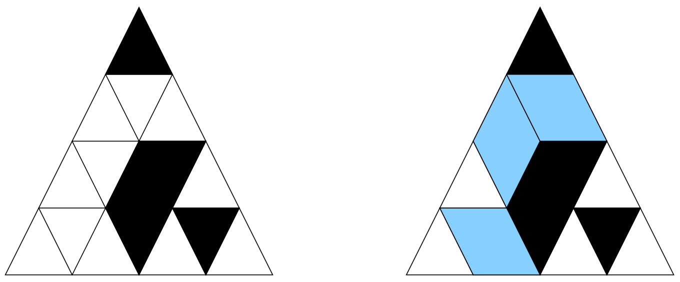


Figure 2: An equilateral triangular land that has 5 bad triangles (left) and 3 houses (right).

Figure 2 illustrates a problem instance. The five bad triangles are illustrated in black and the three houses are illustrated in gray.

Input File Format

The first line of the input file has the number of test cases, there are at most 10 test cases. The first line of a test case contains S $(1 \leq S \leq 100)$, the number of unit equilateral triangles along each side of the equilateral triangular land. The next S lines indicate the status of unit equilateral triangles in each row. The unit equilateral triangles are listed from top to bottom, and from left to right. An 1 indicates that the unit equilateral triangle can be used for house construction, and a 0 indicates a bad triangle.

Output Format

For each test case, output the maximum number of houses that can be built in this equilateral triangular land.

Sample Input

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

Output for the Sample Input

```
3
1
```

Problem G
Postage stamp problem
Input File: *pg.in*
Time Limit: *120 sec.*

This problem is called postage stamp problem. Envision a country that issues n different denominations of stamps but allows no more than m stamps on a single letter. For given values of n, m and p, write a program that computes the greatest consecutive range of postage values, from one up to p, and output one possible set of denominations that realize that range. For example, for $n = 5$, $m = 3$ and $p = 36$, the stamps with values (1,4,6,14,15) is a set that satisfies the requirement.

Input File Format

The first line of the input file contains an integer,k ($k \leq 5$), indicating the number of test cases. Each test case consists of 3 integers, n, m, p ($n + m \leq 8, p \leq 100$) in one line separated by spaces, where n is the number of different denominations of stamps, m is the maximum number of stamps allowed on a single letter, and p is the maximum postage value desired.

Output Format

List the values on n stamp denominations separated by spaces, or x if you cannot find a solution. Please note you only have to output one set of denominations.

Sample Input

```
1
5 3 36
```

Output Sample

```
1 4 6 14 15
```