Sapientia
Hungarian University of Transylvania

Sapientia ECN International Programming Contest – 2010

PROBLEMS

Târgu Mureş / Marosvásárhely, 7–8 May 2010

PROBLEM A

Binary Tree

Input File:	**A.IN**
Output File:	**STANDARD OUTPUT**

It is very easy to produce the various traversals of a binary tree, however this problem requires you to produce the tree from the traversals. Specifically, given the pre-order and in-order traversals of a binary tree (not necessarily a binary search tree), reconstruct, if possible, the original tree.

(Source: NEW ZEALAND PROGRAMMING CONTEST 2008)

For instance, the following tree will produce the traversals shown

```
          E
        /   \
       D     F
      /       \
     B         G
    / \         \
   A   C         I
                / \
               H   K
                  /
                 J
```

```
Preorder : EDBACFGIHKJ
Inorder  : ABCDEFGHIJK
Postorder: ACBDHJKIGFE
```

Input

Input will consist of representations of several trees, each on a single line, and terminated by a #. Each line will consist of two strings of up to 26 unique uppercase letters in the form

```
<pre-order> <space> <in-order>
```

Output

Output will consist of the post-order traversal of the reconstructed tree if possible, otherwise the words 'Invalid tree'.

Sample Input: A.IN
EDBACFGIHKJ ABCDEFGHIJK
#

Sample Output: Standard output
ACBDHJKIGFE

PROBLEM B

Timetable inconsistencies

Input File:	B.IN
Output File:	STANDARD OUTPUT

An international programming contest is hosted by Landia, a beautiful country. The organizers detected some inconsistencies when they consulted the official website for information about train timetable. More precisely, the distance between two arbitrary linked stations is not always the same. This inconsistency leads to different prices when two travelers take distinct trains, even though the trains are of same rank and cover the same route.

The organizers want to verify each pair of linked stations; two stations are linked if there is no other station between them. They must display every pair for which distinct distances are detected.

Input

The timetable for each train is given as follows:

- the first line shows information about train (rank, identifier etc); the first character of this line is always letter;
- the second line, the fourth line etc, show an information about a particular station: distance covered by train until this time, station identifier, station name;
- the station identifier is an unsigned integer number; the station name is at most 55 characters length;
- for some particular trains, only partial route is depicted (see Express train 500);
- the third line, the fifth line etc, show the arrival and departure time; you have to ignore these lines;
- the input file size is about 50000 lines. For each pair of stations, we don't detect more than two distinct distances.

Output

- the output pairs must be ordered alphabetically;
- the answer time must be at most two seconds.

Sample Input: B.IN

```
Slow train 2000
0 51000 Klaus
- 7:25
5 71020 Canda
7:35 7:37
12 61030 Bank
8:10 -
Fast train 1000
0 51000 Klaus
- 10:0
6 71020 Canda
10:15 10:18
15 81040 San Dinka
10:45 -
Express train 500
10 81040 San Dinka
- 9:0
20 71020 Canda
9:45 9:50
27 61030 Bank
10:12 -
```

Sample Output: Standard output

```
71020 Canda - 51000 Klaus
71020 Canda - 81040 San Dinka
```

PROBLEM C

Ellipsoid

Input File:	`C.IN`
Output File:	`STANDARD OUTPUT`

As it is now well known, the real world is multi-dimensional: the physicists talk about ten, eleven or even more dimensional spaces. Also it is known that the matter is built up from elementary particles.

A couple of days ago[1] a leading physicist made a crucial discovery: the mass of an elementary particle is quantized, and it is equal to the number of lattice points inside an s-dimensional ellipsoid, having only integer coordinates! The shape of this ellipsoid in the s-dimensional space is described by the positive integers $a_1, a_2, \ldots, a_s$, which simply express the extent the elementary particle penetrates the space in the directions $x_1, x_2, \ldots, x_s$ respectively.

So for example, if a particle is essentially 2-dimensional and its shape is described by the constants 5 and 3, in the direction x and y respectively, then it's mass is the number of lattice points inside the ellipse in the Fgure 1.

The mass of this particle is therefore 45 units.

A simple computer algorithm to count these points may looks like Algorithm 1.

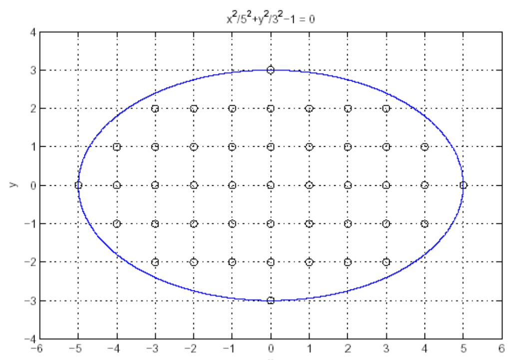


Figure 1.

```
read(a,b)
N=0
for(x=-a,x<=a,x=x+1)
  for(y=-b,y<=b,y=y+1)
    if x^2/a^2+y^2/b^2<=1
      N=N+1
    end
   end
end
print(N)
```

Algorithm 1.

The task is to quickly write a program which counts the mass of a given particle in s-dimensional space! So the input of this program is first of all a number s (the dimension of the space) and then a sequence of positive integers $a_1, a_2, \ldots, a_s$ (the shape of the ellipsoid). It is obvious that a lattice point having integer coordinates $x_1, x_2, \ldots, x_s$ is inside the ellipsoid if and only if:

$$\frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \ldots + \frac{x_s^2}{a_s^2} \leq 1.$$

Input

C.IN contains in the first line a positive integer, the number of test cases, then for each test the first line contains a positive integer s, and a second line containing s positive integers, separated by commas.

Output

In the output you will write a single number representing the mass of the corresponding particle.

Sample Input: C.IN
1
2
5,3

Sample Output: Standard output
45

[1] the story is fictional

PROBLEM D

Dance

Input File:	D.IN
Output File:	STANDARD OUTPUT

We consider a ballroom of rectangular shape divided in squares of size 1×1, having *a* rows and *b* columns. Here *n* girls and *n* boys will dance. They are numbered from 1 to *n*. At first, each of them is staying on a given square. Every girl has a colored cordon, long enough to be catched by a boy along a straight line.

The organizer wants to put the dancers in pairs (a girl with a boy) such as when the cordons will be stretched by them, these do not intersect each other. It is known, that three dancers will not stay along the same line.

If there are no three dancers, placed along a same straight line, determine the pairs such as the cordons do not intersect each other.

Input

The first line will contain the number *t* of the test cases. The format of each test case is the following. The first line contains the number *n* of the girls (equal with the number of boys). The next *n* lines contain exactly two positive numbers, identifying the positions of the girls on the dancing place (in the i^{th} line the first number notes the column, the second represents the row, where the i^{th} girl is staying). The next *n* lines will contain the positions of the boys in a similar way. Two test cases are separated by an empty line.

Output

Your output should contain exactly *t* packages of answers. Each of them will consist of up to *n* lines (where *n* is the corresponding value from the input). On each line there will be two positive numbers, representing the number of the girl and the number of the boy from a pair. These numbers represent the order from the input file. If there are at least three dancers which are staying along the same straight line, the output file should contain '0 0'. All the test cases will be followed by an empty line.

Constraints

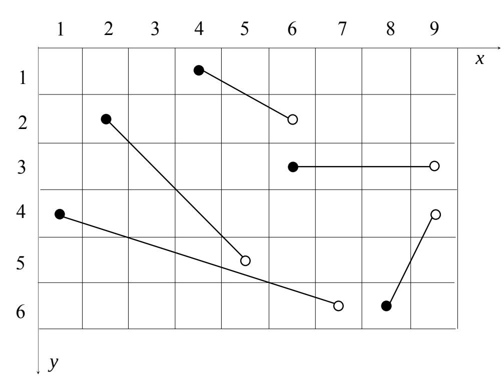
- $5 \le n \le 50$;
- $1 \le a \le 600$;
- $1 \le b \le 300$;
- The numbers of the girls will be written in the output file in increasing order;
- The distance between a girl and a boy can be 0 (they are staying on neighboring squares);
- If there is more than one solution, in the output file will be written only one.
- Time limit: 1 sec/test

Sample Input: D.IN

```
5
5 5
6 2
7 6
9 4
9 3
8 6
6 3
4 1
2 2
1 4
```

Sample Output: Standard output

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



PROBLEM E

Cut – Join – Mirror

Input File:	`E.IN`
Output File:	`STANDARD OUTPUT`

Given n ($0 < n < 10$) a non-negative integer. For all integer-pairs (x, y) ($0 \le x < y < 2^n$), containing on their n-bit binary representation the same number of 0-s, determine the minimal number of steps to transform x into y, applying the following operations on the binary representation of x:

- **cut**: cut the current bit-segment in two non-empty sub-segments;
- **join**: join two neighbour sub-segments;
- **mirror**: mirror the current bit-segment.

For example, if $n = 4$, $x = 3$ and $y = 10$ than the minimal-step solution is the following (6 step):

0011 cut 0|011 mirror 0|110 join 0110 cut 01|10 mirror 10|10 join 1010

Remarks:

- 0011 is the binary representation on 4 bit of 3
- 1010 is the binary representation on 4 bit of 10

Input

The first line contains the number of the test cases. The second line contains the values of n for each test-case separated with one space-character.

Output

Your program (for each test case) has to print to the standard output for all valid (x, y) pairs (according to their lexicographic order) the corresponding minimal-step. Between test-cases an empty line has to be inserted. Each non-empty line of the output has to contain three numbers separated with one space-character: the corresponding values of x, y and the minimal-step.

Sample Input: E.IN

```
2
3 2
```

Sample Output: Standard output

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

1 2 1
```

PROBLEM F

Knuth's Permutation

Input File:	`F.IN`
Output File:	`STANDARD OUTPUT`

There are some permutation generation techniques in Knuth's book *“The Art of Computer Programming – Volume 1”*. One of the processes is as follows:

> For each permutation $A_1, A_2, \ldots, A_{n-1}$ form n others by inserting a character n in all possible places obtaining
>
> $$\begin{gathered} n\,A_1\,A_2\,\ldots\,A_{n-1} \\ A_1\,n\,A_2\,\ldots\,A_{n-1} \\ \ldots \\ A_1\,A_2\,\ldots\,n\,A_{n-1} \\ A_1\,A_2\,\ldots\,A_{n-1}\,n \end{gathered}$$
>
> For example, from the permutation 231 inserting 4 in all possible places, we get 4231, 2431, 2341, 2314.

Following this rule, you have to generate all the permutations for a given set of characters. All the given characters will be different, their number will be less than 10, and they all will be alphanumeric characters. This process is recursive, and you will have to start the recursive calls with the first character and keep inserting the other characters in order. The sample input and output will make this clear. Your output should exactly match the sample output for the sample input.

Input

The input contains several lines of test cases. Each line will be a sequence of characters. There will be less than ten alphanumeric characters in each line. The input will be terminated by End of File.

Output

For each test case, generate the permutations of those characters. The order of the input characters is very important for the output, i.e. the permutation sequence for **abc** and **bca** will not be the same.

Separate each set of permutation output with a blank line.

Sample Input: F.IN
abc bca dcba

Sample Output: Standard output
cba bca bac cab acb abc acb cab cba abc bac bca abcd bacd bcad bcda acbd cabd cbad cbda acdb cadb cdab cdba abdc badc bdac bdca adbc dabc dbac dbca adcb dacb dcab dcba

PROBLEM G

The Garden

Input File:	`G.IN`
Output File:	`STANDARD OUTPUT`

Once, a gardener ordered different kind of flowers which were delivered one after the other. These flowers were planted line by line by the gardener on the basis of their arriving in an $n \times m$ matrix shape garden.

Write a program that will calculate in which column of the garden will be the most flowers with the same color. If there will be more solutions you have to write down the first solution (the smallest column index).

Input

The first line contains the number of the test cases. In the corresponding line of each case a string and two positive integers (separated by spaces) are written. The characters of the string represent the colors of the flowers, the first integer is the value of n (number of the lines) and the second one is m (number of the columns).

Output

Your program has to print (for each test case) two numbers and one character separated by spaces: the number of column (the number of start column is 1) in which the most unicolered flowers are, and the number of most flowers with the same color, and what is this color.

Restrictions

- $n < 100$
- $m < 100$

Sample Input: G.IN

```
2
ABCBBADCBDBA 4 3
KFPSSKPFFKPSSPFKPSPK 5 4
```

Sample Output: Standard output

```
2 3 B
3 4 P
```

PROBLEM H

Divisibility

Input File:	**H.IN**
Output File:	**STANDARD OUTPUT**

We define the base b expansion, of a given number N as: $(a_0, a_1, \ldots, a_k)_b$, where $N = b^0 \cdot a_0 + b^1 \cdot a_1 + \ldots + b^k \cdot a_k$. We assume that $2 < b \leq 500$, and $0 \leq k < 100$. Write a program for checking the divisibility of N by factors of $b - 1$.

Input

The H.IN input file contains several test cases. One test case consists of three lines, the first stores a number that indicates the base b, the second the length of the number's base expansion, and the third line the base b expansion of the given number N. After a blank line, continue to appear more test cases.

You have to check the divisibility of number N, by factors of $b - 1$ for each test case.

Output

The output must be printed to the standard output and consists in corresponding divisors in decreasing order, in different line for each test case. Write –1 in the corresponding line if the divisibility test fails.

Sample Input: H.IN

```
256
15
7 15 122 250 8 210 126 7 87 52 0 123 67 56 10

3
4
2 0 1 0

35
14
12 4 19 16 0 6 33 7 3 14 15 6 5 13
```

Sample Output: Standard output

```
15 5 3
-1
17
```

PROBLEM I

Student

Input File:	`I.IN`
Output File:	`STANDARD OUTPUT`

Our hero is a student in city C and just got his driving license. He is driving from his natal city O to city C and transports hitchhikers in order to earn some money. His car has a capacity of $k + 1$ persons, so in a certain moment he can transport k persons (and himself). From city O to city C there are l localities (cities, towns, villages), city O being the first and city C being the l^{th}. There are n groups of hitchhikers along the road. The i^{th} group consists of p_i persons, wants to travel from locality s_i to locality d_i and will pay an amount of m_i money. A group must be taken into the car as a whole.

Restrictions

- $1 \leq k \leq 7$
- $1 \leq l \leq 1.000.000$
- $1 \leq n \leq 50$
- $1 \leq p_i \leq k$
- $1 \leq s_i \leq l - 1$
- $2 \leq d_i \leq l$
- $s_i < d_i$
- $1 \leq m_i \leq 1.000.000$

Input

The first line of the input contains the number of test cases. The first line of each test case contains the numbers n, l and k separated by a single space. n lines follow, the i^{th} containing p_i, s_i, d_i and m_i separated by a single space.

Output

For each test case output a single line containing the maximal amount of money the student can earn.

Sample Input: I.IN

```
1
5 5 4
2 1 5 50
1 2 4 20
2 3 4 40
2 4 5 50
3 4 5 80
```

Sample Output: Standard output

```
140
```

PROBLEM J

Squares

Input File:	`J.IN`
Output File:	`STANDARD OUTPUT`

I have a rectangular paper. Its size is $m \times n$. I want to cut this paper in a minimal number of squares. To cut the paper means a full cutting in longitudinal or transversal direction.

For example, a rectangle which size is 6×8, we can cut into squares in many ways:

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

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

1	1	1	1	1	1	2	2
1	1	1	1	1	1	2	2
1	1	1	1	1	1	3	3
1	1	1	1	1	1	3	3
1	1	1	1	1	1	4	4
1	1	1	1	1	1	4	4

The minimal number of squares is 4.

Input

The input file has only one line. This line will contain 2 integer numbers m and n representing the rectangle's sizes, separated by a space.

Output

Your task is to write to the standard output, an $m \times n$ matrix with minimal number of squares. The matrix will be filled by the number of each particular square (1, 2, 3, ...).

Restrictions

- $3 \leq m, n \leq 100$

Sample Input: J.IN

```
6 8
```

Sample Output: Standard output

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

or

Sample Input: J.IN

```
6 8
```

Sample Output: Standard output

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