Problem : Greedy Hostel Owner

You know that summers are at peak this year and every day is hot and due to this everyone is using coolers and ACs and a lot of electricity is consumed by the people.

You are living in a hostel and your hostel owner decided to charge extra for electricity consumption. To achieve this he put one separate electricity meter for every room and connected all those meters to central meter.

But the hostel owner is a bit greedy and wants to manipulate the meters to show a reading that is more than the actual consumption of electricity. He also encrypted all the meters with alphabets. The technique he used for encrypting is as follows:

Every meter has 6 Alphabets i.e. 6 digits.

Every alphabet is in upper case.

Allowed alphabets are A, B, C, D, E, F, G, H, I, J.

A corresponds to 0, B = 1 and similarly C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8, J = 9

The interpretation rules change as follows:

If the alphabet next to J is A, then J represents 0. Similarly, if the alphabet after I is B, then I counts as 1 (and not 8), the alphabet after H is C, then H represents 2. The same is true if D follows G and if E follows F. Note that A, B, C, D and E will always retain their respective values.

When J is not followed by A, J will represent 9 and similar rules for I, H, G and F

You are given central meter reading and encrypted readings of all the meters in the hostel. Your task is to find out whether the owner is Greedy or Innocent. If he is greedy then print the unit difference otherwise print innocent.

Owner is greedy if and only if
(units of all meters in the hostel except central meter < central meter units)

Input Format:

First line contains an integer N, giving the number of rooms in the hostel.

The next line contains N strings each of length 6 characters giving the readings of the meters in the rooms

The next line contains an integer that gives the reading in the central meter

Output Format:

First line containing either GREEDY or INNOCENT

If the first line is GREEDY, the next line should contain the difference (as a decimal number) between the central meter reading and the consumption shown in the rooms.

Constraints:

Number of rooms <= 100

Example 1

Input
3
JAABHF JAACJA JAACDA
500

Output
GREEDY
105

Explanation
In the reading JAABHF, J represents 0 since it is followed by A, and hence the reading is 000175. Similarly, the other readings are 000200 and 000230. The total of the readings in the rooms is 605 and the central meter reading is 500. Thus the owner is GREEDY and he stole 605-500 = 105 units.

Example 2

Input
8
JAACJA JAABCH JAABHD JAACAF JAJAJJ JAABEJ JAACJJ JAACDI
1500

Output
INNOCENT

Explanation
The readings are,

000200, 000127, 000173, 000205, 0000099, 000149, 000299, 000238

The sum of these readings is 1490 < 1500, the central meter reading. Hence the owner is INNOCENT.

Problem : Numbers with non-decreasing digits

Some numbers such as 7, 234, 12378 have the digits that are non-decreasing when we read them from left to right. In this problem, we want to find the largest such number less than or equal to a given number N.

Input Format:

Integer N

Output Format:

Largest integer M ≤ N that has its digits non-decreasing. The output should not contain leading zeros.

Constraints:

N <= 10^18

Example 1

Input
89

Output
89

Explanation
89 itself has non-decreasing digits.

Example 2

Input
549

Output
499

Explanation
From 500 to 549, the integers have 5 as the leading digit and the second digit must be less than or equal to 4. But then, such a number cannot have its digits non decreasing.

Note:

Please do not use package and namespace in your code. For object oriented languages your code should be written in one class.

Problem : Air in the balloons

You have been given 'N' number of spherical Balloons of different radius when filled. You have to fill one balloon per day and the last balloon will be filled on Nth day. There is some rate of air reduction 'K' per day from each balloon. Fill the balloons in such an order so that the sum of the volume of all the balloons is maximum on the day when all the balloons are filled.

Input Format:

First Line is an integer N giving the number of balloons.

Second line gives space separated N positive real numbers with up to 1 decimal place giving the radii of the balloons.

Third line gives K, the rate of reduction in the volume of air as a percentage.

Output Format:

Maximum sum of volumes of all the balloons on the Nth day when all the balloons are filled. Take 3.14 as the value of PI and give the answer to two decimal places (truncated by ignoring all the decimals from third onwards). Note that the truncation should happen only after computing the volume of all the balloons on the final day to maximum precision.

Constraints:

Number of balloons <= 10
Radius of balloons <=200

Example 1

Input
5
8 4 6 10 3
10

Output
7117.88

Explanation
If we fill the balloons in the order 3, 4, 6, 8, 10, their volumes on the fifth day are respectively
74.165544
195.33312
732.4992
1929.216
4186.666667
And their sum is 7117.880531. Truncating the value two decimal places, we obtain 7117.88

Example 2

Input
7
3.5 9 4 6.6 7 11 9.1
12.5

Output
12555.35

Explanation
If we inflate the balloons in the order 3.5 4 6.6 7 9 9.1 11, their volumes on the seventh day would be respectively
80.56025567
137.4322396
705.5574848
962.0256771
2336.74875
2760.581763
5572.453333 The sum of these volumes is 12555.3595 and truncating to two decimal places, we obtain 12555.35

Problem : One Egg

One Egg is an egg supply company which supplies eggs to retailers. They have M classes of eggs. Each class can have N number of eggs (N can be same or can vary class to class).

They accept an order via mail for X eggs. In response, they confirm if they can supply the eggs with a Thank you note and the number of eggs or with a Sorry note and the numbers of eggs they can supply. They also mention the breakdown of eggs by class they will supply. The ordered eggs are adjusted against the different classes with the most number of eggs adjusted first then the balance is adjusted against the second highest and so on.

The company is a bit superstitious as well. If the number of eggs ordered is greater than or equal to the total number of eggs in stock then they retain one egg and responds back with the "Sorry" note with total number of eggs in stock minus one and breakdown of eggs by class.

Note: If the classes have same number of eggs then class entered first should be selected to adjust.

Input Format:

First line contains two space-separated integers denoting the respective values of M (the number of classes of eggs) and X, the number of eggs ordered

The following M lines contain an integer each indicating the number of eggs available in each class

Output Format:

First line should be, if X is less than total number of Eggs then Print
Thank you, your order for X eggs is accepted
Else if X is greater than or equal to total number of Eggs then print "
Sorry, we can only supply (total number of Eggs in stock -1) eggs
Then M lines with 3 columns:
First column - Number of eggs available in each class
Second column - Eggs allocated against each class for that order
Third column - Balance Eggs against each class

Constraints:

$1 \leq M \leq 20$
$N \geq 1$
$X \geq 1$

Example 1

Input
5 150
50
15
80
10
5

Output
Thank you, your order for 150 eggs are accepted
50 50 0

15 15 0
80 80 0
10 5 5
5 0 5

Explanation
Total order of 150 eggs is less than the total number of Eggs 50+15+80+10+5 = 160. Hence the Thank you message.

150 was first adjusted against Class with first highest number of eggs 80. Balance of 150-80 = 70 was adjusted against second highest class of 50. Balance of 70-50 = 20 then adjusted against 15. Balance of 20-15 = 5 then adjusted against 10 leaving behind 5 eggs in that class.

Example 2

Input
4 250
80
50
70
20

Output
Sorry, we can only supply 219 eggs
80 80 0
50 50 0
70 70 0
20 19 1

Explanation
Total order of 250 eggs was greater than the total number of eggs 80+50+70+20 = 220. Hence the sorry message.

250 was first adjusted against Class with first highest number of eggs 80. Balance of 250-80 = 170 was adjusted against second highest class of 70.

Balance of 170-70 = 100 was then adjusted against 50. Balance of 100-50 = 50 then adjusted against 20. Since Balance is greater than last class of egg all but one egg is left in that last class.

Note:

Please do not use package and namespace in your code. For object oriented languages your code should be written in one class.

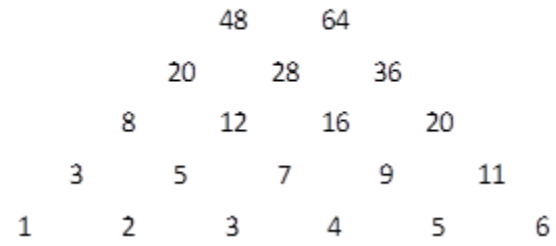
Note:

Participants submitting solutions in C language should not use functions from <conio.h> / <process.h> as these files do not exist in gcc

Problem : Pascal Pyramid

Pascal's triangle giving binomial coefficients is well known. In this triangle, elements in each row can be obtained by adding two adjacent elements in the previous row. The pyramid of numbers we construct in this problem is similar to the Pascal triangle. We start with six numbers at the base of the pyramid, and construct the pyramid one layer at a time by adding the two adjacent numbers in the previous layer. For Example, starting with the numbers 1 2 3 4 5 6 in the base, we get the following pyramid.

The apex of the pyramid is filled with the product of the numbers in the row below instead of the sum.



In the above pyramid, the apex is filled with the number 48 x 64 =3072. The aim is to get the largest number possible at the apex of the pyramid.

The input will be a set of N positive integers. Six need to be chosen from these and arranged at the base to get the largest possible number at the top.

Input Format:

The first line of the input is N, the total number of integers that will be given.

The second line is a set of N (not necessarily distinct) comma separated positive integers from which the six numbers at the base need to be selected.

Output Format:

The output is one line with an integer representing the maximum value of the apex of the pyramid when six integers are selected and arranged suitably at the base.

Constraints:

N < 13
Integers provided for selection ≤ 100

Example 1

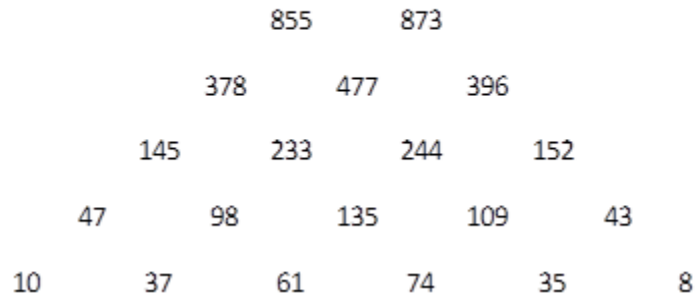
Input
8
10,4,74,61,8,37,2,35

Output
746415

Explanation

There are 8 numbers given, from which the 6 numbers in the base are to be selected and arranged so as to maximize the apex number. One way of doing this is in the figure below.

The product of the two numbers below the apex is 746415, which is the output.

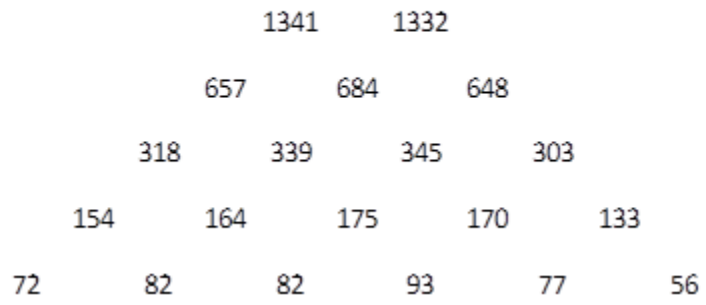


Example 2

Input
10
37,93,56,10,77,82,72,82,39,7

Output
1786212

Explanation
There are 10 numbers (N=10). One arrangement that will give a high product at the apex is



The product of the numbers at the top is 1786212, which is the output.

Note:

Problem : Prime Counters

Given a number N, let CP(N) denote the number of primes between 1 and N (inclusive of N). We call N a prime counter if CP(N) is a prime (N need not be a prime).

For example, CP(3) = 2, CP(4) = 2, CP(5) = 2, CP(7) = 4.

Input Format:

An integer T, number of test cases

T lines each containing two positive integers L, R separated by space

Output Format:

T lines containing the number of prime counters between L and R (both inclusive) in the ith test case (or NONE is no prime counter exists in that range)

Constraints:

$L \leq R \leq 10^6$

Example 1

Input
1
1 10

Output
4

Explanation
CP(1) = 0, CP(2) = 1, CP(3) = 2, CP(4) = 2, CP(5) = 3, CP(6) = 3, CP(7) = 4= CP(8) = CP(9) = CP(10)

Hence there are 4 prime counters, 3, 4, 5, 6 in the range 1 to 10.

Example 2

Input
2
2 20
3 30

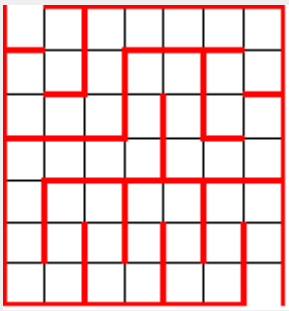
Output
8
8

Explanation
Up to 10, we have 4 prime counters. Between 11 and 20 the prime counters are 11, 12, 17, 18 and hence the count is 8. Between 21 and 30, we have no prime counters.

Note:

Problem : Shortest Path in a Maze

In the fairground of Dizzyville there is an interesting maze. It is square, and its walls are made of wood, with identical locks on them. You are given one key, so that one wall can be unlocked, but the key gets stuck in the lock, and so cannot be removed. Hence only one wall can be eliminated, but you can choose which wall.



You need to determine the length of the shortest path out of the maze (number of squares visited), with perhaps unlocking one door. The positions of the doors will be given with the coordinates of the start and end points.

The start will always be the top left corner of the maze, and the end, the bottom right corner of the maze.

The wall position will be based on the starting and ending points, which is based on the origin of the point coordinate system. This is defined as the lowest left corner of the maze, with the positive X axis along the horizontal line to the right.

Input Format:

The first line consists of two space separated positive integers, giving the size of the maze (N) and the number of interior walls (k).
There will be k lines with four space separated numbers being the starting x and y coordinates, and the ending x and y coordinates of the door. Note that this will be in the coordinates using the point coordinate system defined above.

Output Format:

The output will be single line with the number of squares visited in the minimum path, after perhaps unlocking one wall.

Constraints:

$N \leq 20$

$K \leq 50$

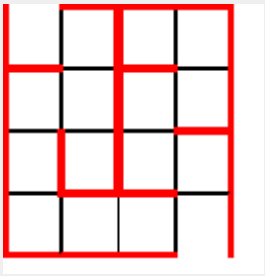
Example 1

Input
4,6
0,3,1,3
2,4,2,1

1,1,2,1
1,2,1,1
2,1,3,1
2,3,3,3

Output
7

Explanation
There are 4 squares in each row of the maze and 6 walls. A picture of the maze is given below. The walls are shown in red lines



The shorted path to the destination from the starting point is 9 squares. If the wall (0,3), (1,3) is opened, the distance will be 7 squares. Hence the output is 7.

Example 2

Input
7,18
0,6,1,6
1,5,2,5
2,7,2,5
0,4,3,4
3,6,3,4
3,6,6,6
6,5,7,5
5,6,5,4
5,4,6,4
4,5,4,3
1,3,7,3
1,3,1,2
1,2,1,1
2,2,2,0
3,3,3,1
4,2,4,0
5,3,5,1
6,2,6,0

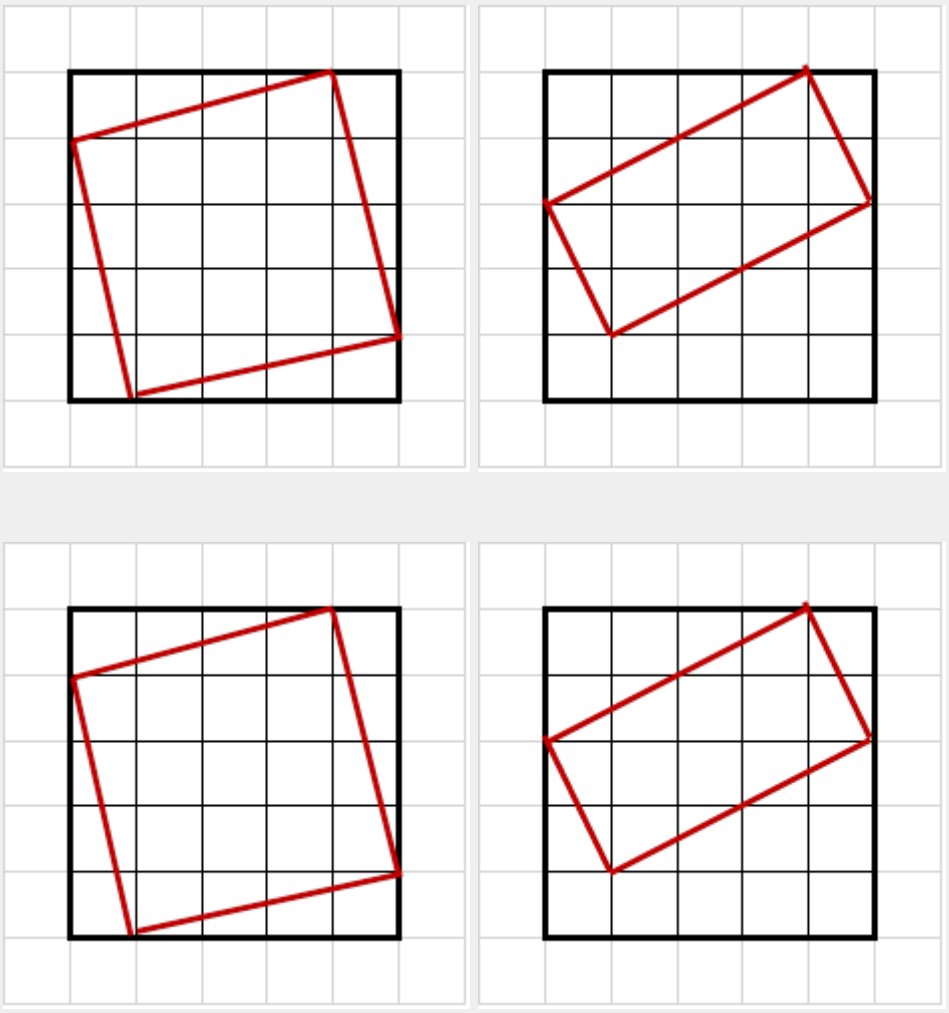
Output
21

Explanation
There are 7 squares a side in the maze. There 18 walls, which are listed. The picture of the maze is the 7 x 7 maze listed at the top. The length of the path without unlocking any walls is 49 squares. If the wall segment between corners (3,6) and (3,7) is unlocked, the length of the shortest path reduces to 21 squares.

Problem : Counting Rectangles

Consider a grid of N x N squares. There are (N+1) x (N+1) grid points (corners of the squares). We need to count the number of rectangles (including squares) that can be formed with vertices at these grid points.

Note that the rectangles may not be aligned to the sides of the squares in the grid. For example, the figure below gives an example of two of these "oblique" rectangles in a 5x5 (or larger) grid of squares.



Input Format:

The input has one line with a positive integer, N, which is the number of squares per side of the grid.

Output Format:

The output is one line with the total number of rectangles that can fit into the grid.

Constraints:

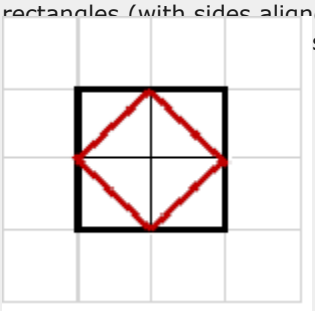
$L \leq R \leq 10^6$

Example 1

Input
2

Output
10

Explanation
The input says that N=2, and we are considering a 2 square X 2 square grid. Consider the "straight" rectangles (with sides aligned to the sides of the grid). There are 4 1x1 squares, 4 2x1 rectangles (including square, a total of 9 rectangles. There is one "oblique" square pictured below.



Example 2

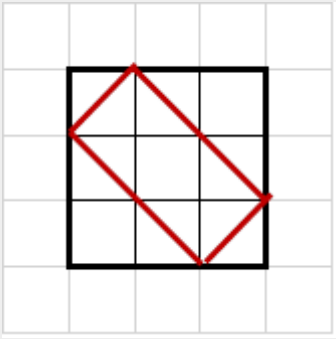
Input
3

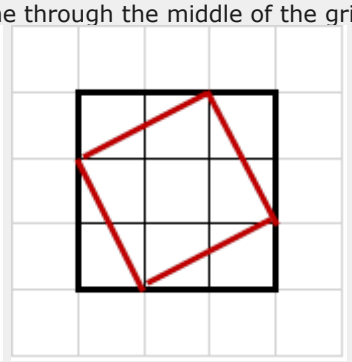
Output
44

Explanation
The input says N=3, and we are considering a 3 square by 3 square grid.

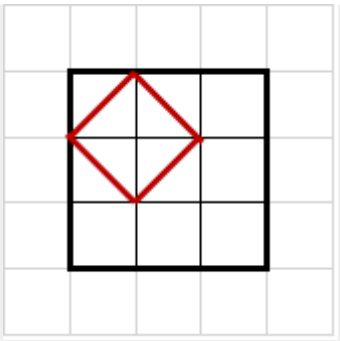
As before, if we consider the "straight" rectangles first. There are 9 1x1 squares, 4 2x2 squares and 1 3x3 square. If we consider 2x1 rectangles, there are 6 with the longer side parallel to the x axis, and 6 with the shorter side parallel to the x axis, If we consider 3x1 rectangles, there are 3 with the longer side parallel to the x axis and 3 with the shorter side parallel to the x axis. If we consider 3x2 rectangles, there are 2 with the longer side parallel to the x axis, and 2 with the longer side parallel to the y axis.

If we look at oblique rectangles, there are two types which have vertices on the 3x3 square. Each may be reflected around a vertical line through the middle of the grid to get a different rectangle, a total of 4.





We also have the oblique square embedded in a 2x2 grid that we looked at in Example 1. This can be embedded in a 3x3 square (see figure below). There are 4 ways to do this (embedding a 2x2 square in a 3x3 square).



The total number of rectangles is 9+4+1 (squares) + 12 + 6 + 4 (straight rectangles) + 4 (oblique with vertices on outside square) + 4 (from embedded 2x2 square) = 44 rectangles. Hence the output is 44.