**Birla Institute of Technology & Science, Pilani**
**Summer Semester 2017-2018, DSA (CS F211)**
Lab Assignment #1

**Problem-1:** A sequence of N positive integers (10 < N < 100 000), each of them less than or equal 10000, and a positive integer S (S < 100 000 000) are given. Write a program to find the minimal length of the subsequence of consecutive elements of the sequence, the sum of which is greater than or equal to S.

**Input**

Many test cases will be given. For each test case the program has to read the numbers N and S, separated by an interval, from the first line. The numbers of the sequence are given in the second line of the test case, separated by intervals. The input will finish with the end of file.

**Output** For each the case the program has to print the result on separate line of the output file.

**Sample Input**

10 15

5 1 3 5 10 7 4 9 2 8

5 11

1 2 3 4 5

**Sample Output**

2

3

**Problems-2:** Given a 2-dimensional array of positive and negative integers, find the sub-rectangle with the largest sum. The sum of a rectangle is the sum of all the elements in that rectangle. In this problem the subrectangle with the largest sum is referred to as the *maximal sub-rectangle*.

A sub-rectangle is any contiguous sub-array of size 1 *X* 1 or greater located within the whole array.
As an example, the maximal sub-rectangle of the array:

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

is in the lower-left-hand corner:

$$\begin{matrix} 9 & 2 \\ -4 & 1 \\ -1 & 8 \end{matrix}$$

and has the sum of 15.

**Input**

The input consists of an *N X N* array of integers.
The input begins with a single positive integer *N* on a line by itself indicating the size of the square two dimensional 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.). *N* may be as large as 100. The numbers in the array will be in the range [-127*;* 127].

**Output**

The output is the sum of the maximal sub-rectangle.

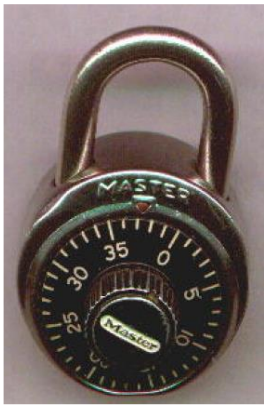
**Sample Input**

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

**Sample Output**

15

**Problem 3:**




In US most graduate student uses a number combination lock. The lock has a dial with 40 calibration marks numbered 0 to 39. A combination consists of 3 of these numbers; for example: 15-25-8. To open the lock, the following steps are taken:

- turn the dial clockwise 2 full turns
- stop at the first number of the combination
- turn the dial counter-clockwise 1 full turn
- continue turning counter-clockwise until the 2nd number is reached
- turn the dial clockwise again until the 3rd number is reached
- pull the shank and the lock will open.

Given the initial position of the dial and the combination for the lock, how many degrees is the dial rotated in total (clockwise plus counter-clockwise) in opening the lock?

**Input**

For each test case there is a line of input containing 4 numbers between 0 and 39. The first number is the position of the dial. The next three numbers are the combination. Consecutive numbers in the combination will be distinct. A line containing `0 0 0 0' follows the last case.

**Output**

For each case, print a line with a single integer: the number of degrees that the dial must be turned to open the lock.

**Sample Input**

```
0 30 0 30
5 35 5 35
0 20 0 20
0 0 0 0
```

**Sample Output**

```
1350
1350
1620
```

**Problem 4:**

As you didn't show up to the yearly general **local craft collectors** meeting of the Collectors Club, you were unanimously elected to organize this years excursion to Pink City. You are free to choose from a number of weekends this autumn, and have to find a suitable hotel to stay at, preferably as cheap as possible.

You have some constraints: The total cost of the trip must be within budget, of course. All participants must stay at the same hotel, to avoid last year's catastrophe, where some members got lost in the city, never being seen again.

**Input** The input file contains several test cases, each of them as described below. The first line of input consists of four integers: $1 \leq N \leq 200$, the number of participants, $1 \leq B \leq 500000$, the budget, $1 \leq H \leq 18$, the number of hotels to consider, and $1 \leq W \leq 13$, the number of weeks you can choose between. Then follow two lines for each of the H hotels. The first gives $1 \leq p \leq 10000$, the price for one person staying the weekend at the hotel. The second contains W integers, $0 \leq a \leq 1000$, giving the number of available beds for each weekend at the hotel.

**Output** For each test case, write to the output the minimum cost of the stay for your group, or "stay home" if nothing can be found within the budget, on a line by itself.

**Sample Input**

```
3 1000 2 3
200
0 2 2
300
27 3 20
5 2000 2 4
300
4 3 0 4
450
7 8 0 13
```

**Sample Output**

900
stay home

**Problem 5:**
The **"*reverse and add*"** method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

For example:

```
          195   Initial number
          591
          ---
          786
          687
          ---
For example: 1473
          3741
          ---
          5214
          4125
          ---
          9339   Resulting palindrome
```

In this particular case the palindrome `9339' appeared after the 4th addition. This method leads to palindromes in a few step for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.
You must write a program that give the resulting palindrome and the number of iterations (additions) to compute the palindrome.
You might assume that all tests data on this problem:
_ will have an answer ,
_ will be computable with less than 1000 iterations (additions),
_ will yield a palindrome that is not greater than 4,294,967,295.

**Input**
The first line will have a number *N* (0 < *N* _ 100) with the number of test cases, the next *N* lines will have a number *P* to compute its palindrome.

**Output**
For each of the *N* tests you will have to write a line with the following data :
***Minimum number of iteration (additions) to get the palindrome*** and **the resulting palindrome itself separated by one space.**

**Sample Input**
3
195
265
750

**Sample Output**

```
4 9339
5 45254
3 6666
```