NEW BULGARIAN UNIVERSITY

Department of *Informatics*

Third Inter-University Programming Contest

24 March, 2018

A. Subsequence Sum

You are given a sequence of n different integers, $a_1,a_2,\ldots,a_n$. Find a subsequence of m elements $a_{i1},a_{i2},\ldots,a_{im}$, $1 \le i_k < i_{k+1} \le n$, $k = 1,2,\ldots,m$ having the maximum possible sum $S = \sum_{k=1}^{m} a_{i_k}$.

Input. For each test case, on the standard input you will receive the numbers n and m, followed by n additional numbers – the elements of the sequence.

Constraints

$1 \le m \le n \le 200$

$-1000 \le a_i \le 1000$

Output. For each test case, print on a separate line the computed subsequence.

Example:

Input.	*Output.*
5 3 1 8 2 9 3	8 9 3

B. Strings

You are given a string A with length n, which consists of lower letters, and contains more than $\frac{n}{8}$ copies of a string B with length 4. Find the string B.

Input. Each test case is comprised of the string A, provided on a separate line of the standard input.

Constraints

The maximum length of the string is 200000.

Output. For each test case on a separate line, print the string B. If more than one such string is found, print the lexicographically-smallest one.

Example:

Input.	*Output.*
`abcdaakjabcdabc`	`abcd`
`abcjabcjabcj`	`abcj`
`bbbbaxyzbbbbaxyzaaxyz`	`axyz`

C. Government

Several parties have own representatives in the parliament. In order to form a government, it is required to have more than half of the members' votes. Write a program that determines which parties or coalitions can form a government, provided that all members of one party or coalition vote for their party or coalition, and all other members vote against.

Input. For each test case there are: number of parties in the parliament, and for each party – name (string of lower and upper-case letters) and the number of representatives from that party (a positive integer).

Constraints. There will be at most 16 parties in the parliament.

Output. For each test case, print the number of different options for creating a government (total number of parties and coalitions) and the minimum number of parties (standalone or part of a coalition) required to form a government.

Example:

Input.	*Output.*
4 A 100 B 30 C 20 D 10 4 A 10 B 10 C 10 D 10	8 1 5 3

Solution of the first test case: A; A,B; A,C; A,D; A,B,C; A,B,D; A,C,D; A,B,C,D;

D. Arithmetic Expressions

Write a program for computing arithmetic expressions with numbers and addition and subtraction operations.

Input. For each test case, the standard input will contain on one line an arithmetic expression, comprised of integers and the plus (+) and minus (-) signs.

Constraints. The numbers are in the interval $[-10000, 10000]$, and the expressions can contain up to 1000 integers.

Output. For each test case, print the solution on the standard output, in the format given below.

Example:

Input.	*Output.*
`2+2`	`2+2=4`
`18-10+2`	`18-10+2=10`

E. Fence

The shape of a terrain is represented as a closed polygon of N sides, such that no two sides touch (except in the apexes) and no two sides cross. The coordinates of the polygon apexes are integer numbers. To fence the terrain, we need to put pillars in all points of the polygon, which have integer coordinates. Write a program that counts how many pillars are needed.

Input. The first line of the standard input will contain the number of tests. Each test will begin with a line with the number N. Then N lines will follow, each of them having the coordinates of each polygon apex – integer numbers that do not exceed 1 000 000 000.

Constraints. $3 \le N \le 100000$

Output. For each test, print the number of required pillars, on a separate line.

Пример:

Input.	*Output.*
1 4 10 0 10 10 0 10 0 0	40

F. Table

A table is comprised of square cells of equal size, organized in M rows and N columns. Each cell has one integer number, with an absolute value less than 1000. If we draw a line in the table, which connects the centers of two cells, located in the same horizontal, vertical or a diagonal, and passing only through cells that contain positive numbers (the line can be a point), that line's sum is the sum of the values of the cells, through which it is passing. Write a program that finds the maximum sum formed in that manner.

Input. The first line of the standard input will contain the number of test cases. Each test begins with a line containing the values for M and N. Each of the following M lines has N integers separated with spaces – the corresponding row of the table.

Constraints. 1 < M < 1001, 1 < N < 1001

Output. For each test case the program should print on a separate line of the standard output the maximum found sum.

Example:

Input.	*Output.*
1 4 3 1 2 0 -1 5 3 -5 0 1 4 3 0	8

G. School

The School of Competitive Programming at NBU has many new contestants. In order to assure that they are regularly informed for the club meetings, the instructor found out that there are N new students, numbered from 1 to N. M couples of the students know each other, and when one of such a couple learns about an upcoming meeting, he can notify the other. The instructor would like to choose as little students as possible, which he can notify personally, so they can in turn spread out the information using the contacts with each other, given that each one that has been notified about a meeting can connect with and notify all of his acquaintances if somebody else has not already done that. Write a program which uses student acquaintance information to find the minimum number of students, who need to be notified by the instructor, so that the information of the next meeting reaches everyone else.

Input. The first line of the standard input will contain the number of test cases. Each test case will begin with a line, having the numbers N and M, separated with a space. Each of the subsequent M lines will contain two space-separated numbers of students that know each other.

Constraints. $5 \leq N \leq 500$

Output. For each test case the program should print on the standard output the minimum number of students.

Example:

Input.	*Output.*
1 6 4 1 3 1 4 2 5 4 6	2

H. Cards

The young programmers Vladi and Zdravko are playing a game of cards with a deck of 24 cards, containing 4 cards with number 1, 2, 3, 4, 5 and 6. They are placing all cards on the table, in a way that they can see them all, and take turns to pick one card and throw it on the ground. Vladi always plays first. If a player cannot pick a card, such that when thrown on the ground, the sum of numbers of all cards on the ground is less than 49 – he loses. Vladi and Zdravko like the game very much, but they cannot yet answer the question: can it be determined with certainty, who will win a game already started, and with what move. Write a program, which determines that.

Input. The first line of the standard input has the number of tests. Each test case contains a string of digits, showing the number of cards that are on the ground, in the order they were thrown. For example 335 means that first Vladi thew 3, then Zdravko threw 3, and finally Vladi threw 5, with Zdravko having the next turn.

Output. For each test case, on a separate line of the standard output, the program must print which player will win the game if he plays optimally, regardless of what the other player plays and with what move. In a guaranteed win, the program should output V or Z, for Vladi or Zdravko respectively. After that the program should print which card should be played by the player whose turn it is, in order to win, or 0, if there is no chance to win against an optimal play of the opponent. If there are several winning moves, the program should print the one with the highest card played.

Example:

Input.	*Output.*
3 355 44223553556 1221	Z 0 Z 0 V 6

NEW BULGARIAN UNIVERSITY

Department of *Informatics*

Third Inter-University Programming Contest

24 March, 2018

I. Numbers

We are given an unlimited number of decimal digits 2 and 3. Using a part of them, we want to write a decimal number, which can be divided by 2 to the power of n, where n is a natural number. We are looking for the smallest such number. If for example n = 4, one digit will not be enough to write such a number: 2 is even, but cannot be divide by 2 to the power of 4 = 16, and 3 is odd. Using two digits of the possible ones, we can write (in order) 22, 23, 32 and 33. The number 32 already has the specified property (actually it also divides to two to the power of 5 = 32). In fact, 33232 also satisfies the condition, but it is bigger. There are no requirements for the two possible digits to be part of the number, or have the same number of occurrences. Write a program that finds the required number or determines that such number does not exist.

Input. The first line of the standard input contains the number of tests. Each subsequent line contains a natural number n for the test.

Constraints. $1 \le n \le 10000$

Output. For each test, the program should display on the standard output a separate line containing the smallest natural number, comprised only of the digits 2 and 3, and which divides to 2 to the power of n, or a line with the string NO if such a number does not exist.

Example:

Input.	*Output.*
2	23232
6	223232
11	

J. Competition

The competing students from the N-university got tired of solving others' problems. So they organized a competition for creating problems. The main requirement was the definitions to be short, as a protestation against the long and often meaningless definitions of the competitive problems. And a problem could qualify only if it was solved by less than half the participants of the real competition organized with these created problems. Your task is to write a program to solve the following problem, which won the competition:

What is the number of prime numbers in the sequence $ak + b$ for $k = 0, 1, \ldots, n$.

Input. For each test case there will be a separate line with the numbers a, b and n.

Constraints. $1 \le b < a \le 10$ and $1 \le n \le 10000$.

Output. For each test case, print on a separate line the resulting number.

Example:

Input.	*Output.*
2 1 10	7

K. Маска

A mask is a string that contains lowercase letters and a star (*). Given a mask and a sequence of strings, determine the strings that match that mask. A string matches a mask if after a substitution of the star with a chosen (variable number) of lowercase letters, the mask and the string look the same. For example `hello`, `homo` and `hohohoho` match the pattern `h*o`, but `hoh` — not.

Input. The first line of the standard input contains the number of test cases. Each test case begins with a number N – the number of strings that will be tested. The next line contains the mask – a string comprised of lowercase letters and exactly one star (ASCII code 42). Following are N lines containing one test string each, comprised of not more than 100 lowercase letters.

Constraints. $1 \le N \le 100$. The length of the mask is not greater than 100.

Output. For each test string, output `YES` or `NO` on a separate line indicating whether it matches the mask or not.

Example:

Input.	*Output.*
2 4 a*a alabala ananas abracadabra aaa 6 h*n hkjdfjfdshodfhscbajkfnxyemfvsn honijezakon atila je bio hun	YES NO YES YES YES YES NO NO NO YES