netsoc

UCD Internet and Computer Science Society

Programming Competition

Questions Sheet

April 22, 2013

Quick guide

Welcome to the second netsoc Programming Competition!

Prizes

One **Google Nexus 7** per winner. There are two tablets to be won.

NOTE: The winner is selected by the jury, and might not reflect the scores on the DOMJudge scoreboard! *This is due to different question sheets per year!*

Rules

- You are required to bring your own computer for the competition
- The submissions open at **6PM** on the 24th of April and close at **8:30PM**
- You are allowed to use **any** text editor of your choice (This includes IDEs, such as eclipse)
- You are **NOT** allowed to use the **Internet**! Exceptions:
  1. Submit solutions to DOMJudge
  2. Consult the documentation of the language of your choice
- The allowed languages and the installed compilers/interpreters are:
  - **C** *gcc v4.6.3*
  - **C++** *g++ v4.6.3*
  - **Java** *OpenJDK v1.7.0_15*
  - **Ruby** *ruby v1.8.7*
  - **Python** *python v2.7*
  - **JavaScript** *node v0.10.4*
  - **LOLCODE** *lci v0.10.4*
  - **Malbolge** *malbolge v0.1.1*

Example I/O code for Netsoc programming contest.

Each of the simple examples below shows how to read and write from stdin and stdout in the majority of the languages used for the competition. With the exception of some of the more esoteric languages such as LOLCODE & Malbolge.

Example C code

```
#include <stdio.h>
int main(void)
{
    char buffer[14];
    int c;
    int i = 0;
    while ((c = getchar()) != EOF) {
        buffer[i] = c;
        i++;
    }
    printf("Hello %s",buffer);
}
```

Example C++ code

```
#include <iostream>
using namespace std;

int main ()
{
    string i;
    cin >> i;
    cout << "Hello " << i << "\n";
    return 0;
}
```

Example Java Code

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;


public class HelloWorld {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader  (new InputStreamReader(System.in))
        String s;
        while((s = in.readLine()) != null){
            System.out.println("Hello, "+s+"!");
        }
    }
}
```

Example Ruby Code

```
s = gets
puts "Hello #{s}"
```

Example Python Code

```
s = raw_input()
print "Hello "+s
```

Java script code (Uses Node.js)

```
process.stdin.resume();
process.stdin.setEncoding('utf8');

process.stdin.on('data', function (text) {
    process.stdout.write('Hello '+text);
    process.exit();
});
```

1 Hello World! - NC2P0

To test the submission system, write a program which greets a person.

Input

The name of the person

Output

”Hello” followed by the name and an exclamation mark.

Example

Input

Dave

Output

Hello Dave!

2 Happy Days - NC2P3

Johnny has a pool in his garden. There are several islands in the pool. Some islands are connected by bridges. Any bridge can be removed. Every day Johnny removes some bridges so that there is only one way from any island to any other. In the evening he returns removed bridges to their places. Also he has some favorite bridges which he never removes. Johnny will be happy if he is able to make a configuration of bridges on the given day which he has never made before. You have to count the amount of days he will be happy. Of course, if the favorite bridges themselves don't satisfy the happiness condition Johnny will not be happy for even single day.

Input

The first line of input file contains number t the number of test cases. Then the description of each test case follows. The first line of each test case contains number n the number of islands. Islands are numbered with integers from 1 to n. Then n lines follow each containing n characters defining the connectivity matrix of those islands. Character in column x of line y will be 1 if the islands with numbers x and y are connected and 0 otherwise. The next line is number p the number of favorite bridges. The next p lines contain the pairs of islands that are connected by favorite bridges.

Output

For each test case print the number of days Johnny will be happy in this situation. *There is an additional new line at the end of the input.*

Constraints

```
1 <= t <= 5
2 <= n <= 30
1 <= p <= min(6, n-1)
```

Example

Input

```
1
4
0111
1011
1101
1110
2
1 2
3 4
```

Output

```
4
```

3 Mixtures - NC2P5

Harry Potter has n mixtures in front of him, arranged in a row.Each mixture has one of 100 different colors (colors have numbers from 0 to 99). He wants to mix all these mixtures together. At each step, he is going to take two mixtures that stand next to each other and mix them together, and put the resulting mixture in their place. When mixing two mixtures of colors a and b, the resulting mixture will have the color (a+b) mod 100. Also, there will be some smoke in the process. The amount of smoke generated when mixing two mixtures of colors a and b is a*b. Find out what is the minimum amount of smoke that Harry can get when mixing all the mixtures together.

Input

There will be a number of test cases in the input. The first line of each test case will contain n, the number of mixtures, $1 \leq n \leq 100$. The second line will contain n integers between 0 and 99 - the initial colors of the mixtures.

Output

For each test case, output the minimum amount of smoke. *You will need to include an additional trailing newline character.*

Example

Input

```
2
18 19
3
40 60 20
```

Output

```
342
2400
```

4 Tautology - NC2P10

Write a program that checks if the given logical expression is a tautology. The logical expression is a tautology if it is always true, regardless of logical value of its variables.

Input

On the first line there is the number of expressions to check (at most 35). The expression is in a prefix notation, that means that operator precedes its arguments. The following logical operators will be used: C - and D - or I - implies E - if, and only if N - not The variables will be lowercase letters (a-z). There will be no more than 16 different letters in the expression. The length of the expression will not exceed 111 characters.

Output

For each expression write one word: YES if it is a tautology, NO in other case. (Add an additional trailing newline to your output)

Example

Input

```
7
IIpqDpNp
NCNpp
Iaz
NNNNNNNp
IIqrIIpqIpr
Ipp
Ezz
```

Output

```
YES
YES
NO
NO
YES
YES
YES
```

5 Logging Game - NC2P11

Logging can be a very mundane job, but Alice and Bob have devised a game to help them pass the time. They take turns choosing a log, and cutting it into 2 smaller logs. The sum of the lengths of the 2 logs equals the length of the original log. The only restriction is that neither of the resulting logs may be shorter than 1 meter in length (but exactly 1 meter is fine). In other words, non-integer lengths are allowed. Alice makes the first cut, and the first logger who cannot make a legal cut loses.

Input

Input begins with an integer T, the number of test cases (less than 450). T test cases follow, each on its own line. Each test case begins with an integer N, the number of logs at the start of the game. N integers follow, giving the initial lengths of the logs. There are at most 7 logs, and the total length of the logs will not exceed 250 meters. Note that the initial lengths of the logs are integers, but logs may be cut to non-integer lengths.

Output

For each test case, output a single line containing the name of the winner of the game, assuming both loggers choose their cuts optimally. *Add a trailing newline character to your output*

Example

Input

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

Output

```
Alice
Alice
Bob
```

6 Squash the Bugs - NC2P12

Bugs have gotten into The Chefs kitchen! Help him trap them all and hell make you a batch of his famous chocolate chip cookies. You are given a given a square map of the kitchen divided into tiles, and in each tile sits some known number of bugs. You also have a square trap, which can be dropped to cover a certain number of tiles (the trap may only cover tiles from within the map, and must be aligned to the borders of the kitchen). However, the trap only catches bugs from one of the tiles which it has covered, having a minimum number of bugs on it. For all possible positions at which the trap can be dropped, determine number of bugs that will be caught.

Input

Two numbers, $0 < n \leq 1000$ (size of the map), and $0 < k \leq n$ (size of the trap), followed by n rows with n numbers, determining the number of bugs on each tile. The number of bugs on each tile will fit in an signed 32-bit integer. *Add a trailing newline character to your output*

Output

You should output n-k+1 rows with n-k+1 numbers in each row.

Example

Input

```
4 2
0 1 2 3
4 5 6 7
8 9 0 1
2 3 4 0
```

Output

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

7 Quadratic Equations - NC2P13

Knowing Johnny’s mathematical talent, our teacher has prepared a new interesting problem for him, hoping he will enjoy solving it. The problem description is given below.

”There is a rectangular room of length l and width w (l and w are integers). The length and width of the room fulfill the relation l=Aw+B, where A and B are given integer constants. The room is divided into square cells of unit dimensions. You have observed that, after adding an integer C to the number of cells in the room, the number of cells becomes divisible by the prime number P. Find all the possible values of the width of the room.”

Input

The first line contains t, the number of test cases (about 10000). Then t test cases follow. Each test case is given in one line containing 4 integers A, B, C and P $(2 \leq P < 106, 0 < A < P, 0 \leq B, C < P)$. P is always a prime number.

Output

For each test case, write the result in one line. The first number K is the number of solutions. Then K numbers X1, X2, ...,XK follow $(0 \leq Xi < P)$ in increasing order, which are the solutions to the corresponding problems. *Add a trailing newline character to your output*

Example

Input

```
2
1 1 0 2
1 2 2 3
```

Input

```
2 0 1
0
```