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 Maximum Random Walk - NC2P2

Consider the classic random walk: at each step, you have a $\frac{1}{2}$ chance of taking a step to the left and a $\frac{1}{2}$ chance of taking a step to the right. Your expected position after a period of time is zero; that is the average over many such random walks is that you end up where you started. A more interesting question is what is the expected rightmost position you will attain during the walk.

Input

The input consists of an integer n, which is the number of steps to take ($1 \leq n \leq 1000$). The final two are double precision floating-point values L and R which are the probabilities of taking a step left or right respectively at each step ($0 \leq L \leq 1, 0 \leq R \leq 1, 0 \leq L + R \leq 1$). Note: the probability of not taking a step would be 1-L-R.

Output Description

A single double precision floating-point value which is the expected rightmost position you will obtain during the walk (to, at least, four decimal places).

Sample Input

```
walk(1,.5,.5)
walk(4,.5,.5)
walk(10,.5,.4)
```

Sample Output

```
0.5000
1.1875
1.4965
```

3 Some More Homework - NC2P14

In the computer science class at the school in Byteland, the teacher handed out the following assignment as homework:

"For an integer **n**, let **bn** denote the bit parity of the binary representation of **n**, i.e. **bn**=0 if **n** has an even number of ones when written down in the binary system, and **bn**=1 otherwise. The numbers **bn**, for **n** $\geq$0, form an infinite sequence of bits (zeros and ones). Given a sequence **c**=(c0, ..., cp-1) of **p** bits, find the first occurrence of sequence **c** as a subsequence of **b** (i.e., the smallest value of index **k** such that for all **i** between 0 and p-1, we have ci = bi+k)."

And the teacher gave his students several short sequences c, asking them to provide the answers next day. Most, as expected, wrote programs to solve the task. Only Johnny computed the results by hand, claiming (quite correctly) that it was quicker that way. The teacher, slightly exasperated, decided to teach Johnny a lesson, and prepared a harder assignment, just for him.

"Given a sequence **c**=(c0, ..., cp-1) of **p** bits, for each s between 0 and p-1, compute the first occurrence of the prefix (c0, ..., cs) of sequence **c** as a subsequence of **b**."

And to be doubly sure that Johnny does his homework using a computer, the teacher gave him some really long sequences to deal with. Now, Johnny is in a bit of a spot, because he has never bothered to learn to program. Please help him out!

Input

The first line of input contains a positive integer $t < 10$, describing the number of tests. Exactly t test cases follow. Each test case is given in two lines. The first line contains integer p ($1 \leq p \leq 106$, the length of sequence k). The next line contains exactly p space-separated numbers (0 or 1), denoting successive elements of sequence c.

Output

For each test case, print a line containing exactly p space-separated numbers, corresponding to the indexes of the first occurrence of successive prefixes of c as subsequences of b. All indexes are numbered starting from zero. If there

is no such occurrence, output -1. (Add a trailing newline character to your output)

Example

Input

```
1
9
1 0 0 1 0 1 1 1 0
```

Output

```
1 2 4 4 8 8 8 -1 -1
```

4 Traffic jam - NC2P15

Little Johnny has a selection of boxes. Each box has a number on its side. The boxes are placed in a sequence, and Johnny wants to sort them (in ascending order). He has a device to manipulate the boxes, which performs the following operation. Johnny can select a subset of boxes, and the machine will lift the selected subset, shift the selected subset to the right (keeping the order in the subset), shift the not-selected subset to the left, filling up empty spaces (keeping the order in the subset), then finally move the raised boxes to be in one level again.

For example: if Johnny has the sequence: 1,2,**3**,**4**,5,**6**, and selects the subset in bold: 1,2,3,4,5,6, then the result is: 1,2,5,**3**,**4**,**6**.

Help Johnny to write a program that will calculate the minimal number of moves required to sort the given sequence of boxes in ascending order of numbers.

Input

First, $1 \leq t \leq 10$, the number of test cases. Then, t test cases follow. Each starts with $1 \leq n \leq 105$, the number of boxes. Then, n integer values describing the sizes of boxes in the sequence.

Output

For each test case, in a separate line, print the answer to that test case. *Add an additional newline character to the end of your output*

Example

Input

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

Output

```
2
```

5 Se7en - NC2P16

Tomek and his numerous friends are standing in a circle. They are all numbered with consecutive identifiers, from 1 to 1337 in the clockwise direction. Starting from person 1, who says "1", successive people read out successive positive integers. The starting direction is clockwise, and there is rule, that whenever integer is divisible by 7 or contains digit 7, the direction is reversed.

So, the identifiers of the persons who read-out successive numbers, are: 1, 2, 3, 4, 5, 6, 7 (person 7 has just read "7" and reversed the direction), 6, 5, 4, 3, 2, 1, 1337 (person 1337 has just read "14" and reversed the direction), 1, 2, 3 (person 3 has just read "17" and reversed the direction), 2, 1, 1337, 1336, and so on. Tomek has his favorite number, and he wants to calculate where in the circle he should stand to read that number out loud. Even though he is skilled programmer, he is a bit little lazy and would like you to help him out.

Input

First, $1 \leq t \leq 1000$, the number of test cases. Each test case is in a separate line, and contains a positive integer smaller than 10100, representing Tomek's favorite number.

Output

For each testcase, output one integer, the identifier which Tomek should choose in the circle to read his favorite integer loud. *Add an additional newline character to the end of your output*

Example

Input

```
3
10
100
1000
```

Output

```
4
2
1311
```

6 Short - NC2P17

Given n and k, find the number of pairs of integers (a, b) such that $n < a < k, n < b < k$ and ab-n is divisible by (a-n)(b-n).

Input

The first line contains the number of test cases t ($1 \leq t \leq 5$). Then t test cases follow, each test case consists of a line containing two integers n and k ($0 \leq n \leq 100000, n < k \leq 1018$).

Output

For each test case output one line containing the required number. *Add an additional newline character to the end of your output.*

Example

Input

```
2
1 5
2 5
```

Output

```
2
3
```

7 A Game of Thrones - NC2P18

Bran and Tyrion are the last two high lords fighting for the Iron Throne. With a mutual agreement that included all knights of the realm, it was decided to settle the issue with a game of thrones, of course not a game of swords but a game of numbers, after all one of them is a cripple and other is a dwarf. Seven wisest men of the realm came forward and forged rules of this game which are as follows :

Initially N numbers were written down on a notebook (possibly with multiple copies of every number).

Players alternate their turns with Bran playing first.

In the first turn Bran gets to choose a number of his choice from the numbers written on the notebook and declare it as the current number of the game.

After that in every move this is what they do : let's say the current number of the game is u. They erase u from the notebook (if u was written multiple times, they erase it only once) and declare one of the numbers still written on the notebook v as current number of the game. v can be chosen iff prime factorization of u and v differ by exactly 1 prime factor. (Read notes for a more formal definition)

The player who can't make a move loses the game.

You're one of Varys' spider and he has asked you to predict the outcome of this game beforehand so that he can devise future strategy. So you've to find out who has a winning strategy assuming both players play optimally.

Notes

1) v can be chosen after u iff either of the following conditions hold :

$v > u$ and u — v and (v / u) is prime

$u > v$ and v — u and (u / v) is prime

2) A natural number if prime if it has exactly two distinct positive factors. 1 is not a prime number.

Input

First line of the input contains a single integer N denoting number of different numbers written in the notebook. Then follow N lines. Each of the following lines contain two space separated integers : ui and ci where ui is the ith

distinct integer written on notebook and it has been repeated ci number of times.

Output

Your program should print Bran if he has a a winning strategy else it should print Tyrion. Also in case Bran could win, your program must output the smallest number Bran could choose in the first turn to ensure a win. See sample output for details.

Add an additional newline character to the end of your output

Example

Input

```
3
2 3
14 3
21 2
```

Output

```
Bran 21
```

7.1 Constraints

```
1 <= N <= 500
1 <= ui <= 10^18
1 <= ci <= 10^9
All ui are distinct.
```