

# 2024 ICPC Texas State – Mexico Invitational Programming Contest

Apr. 5, 2024

---

## Problem 1: Stone Game

### Description

Tom and Jerry are good friends, and they like to play a game with a pile of stones that initially contains  $S$  stones ( $1 \leq S < 10^{10000}$ ). They alternate turns, with Tom going first. When it is a person's turn, he must remove  $x$  stones from the pile, where  $x$  is any positive integer palindrome of the person's choosing. If the pile is empty when a person's turn starts, that person loses.

Definition: A positive integer is a palindrome if it reads the same forward and backward; examples of palindromes include 1, 121, and 9009. Leading zeros are not allowed; e.g., 990 is not a palindrome.

There are  $T$  ( $1 \leq T \leq 10$ ) independent test cases. For each test case, print who wins the game if both Tom and Jerry play optimally.

### Input Specification

- The first line contains  $T$ , the number of test cases.
- The next  $T$  lines describe the test cases, one line per test case. Each test case is specified by a single integer  $S$ .

### Output Specification

- For each test case, output  $T$  if Tom wins the game under optimal play starting with a pile of stones of size  $S$ , or  $J$  if Jerry wins, on a new line.

### Sample Input:

```
3
8
10
12
```

### Sample Output:

T  
J  
T

## Problem 2: Food Exchange

### Description

Alice is a kind-hearted girl who has adopted many stray dogs and feeds them with bones. She noticed that when feeding them, these dogs enjoy forming a circle and passing bones to each other.

Assume Alice's  $N$  ( $1 \leq N \leq 2 \times 10^5$ ) dogs are lined up in a circle such that for each  $i$  in  $1, 2, \dots, N-1$ , the dog to the right of dog  $i$  is dog  $i+1$ , and the dog to the right of dog  $N$  is dog  $1$ . The  $i^{\text{th}}$  dog has a bucket that can hold up to  $a_i$  ( $1 \leq a_i \leq 10^9$ ) bones. All buckets are initially full.

Every minute, the dogs exchange bones according to a string  $S_1S_2\dots S_N$  consisting solely of the characters 'L' and 'R'. If the  $i^{\text{th}}$  dog has at least 1 bone, she will pass 1 bone to the dog to her left if  $S_i='L'$ , or to the right if  $S_i='R'$ . All exchanges happen simultaneously (i.e., if a dog has a full bucket but gives away a bone but also receives a bone, its bones remain the same). If a dog's total bones ever end up exceeding  $a_i$ , then the excess bones will be lost and wasted.

After  $M$  minutes ( $1 \leq M \leq 10^9$ ), what is the total amount of bones left among all dogs?

### Input Specification

- The first line contains  $N$  and  $M$ .
- The second line contains a string  $S_1S_2\dots S_N$  consisting solely of the characters 'L' or 'R', denoting the direction each dog will pass their bones in.
- The third line contains integers  $a_1, a_2, \dots, a_N$ , the capacities of each bucket.

### Output Specification

Output an integer, the sum of bones among all dogs after  $M$  minutes.

#### Sample Input:

```
9 5  
RRLLRLLR  
5 8 4 9 3 4 9 5 4
```

#### Sample Output:

```
38
```

### Problem 3: Disneyland Visit

Mary is a student at Texas State University, and she is excited to visit Disneyland in Orlando this summer for the first time. Orlando Disneyland has many attractions that are numbered from 1 to  $N$ . Each attraction  $i$  closes at time  $c_i$ . Mary wakes up at time  $S$  and she wants to maximize the productivity of her day by visiting as many attractions as possible before they close. She plans to visit attraction  $i$  at time  $t_i + S$ . However, she must arrive at an attraction strictly before its closing time. Otherwise, she cannot visit it.

Mary has  $Q$  questions. For each question, she provides two integers  $S$  and  $V$ . She wants to know if she can visit at least  $V$  attractions if she wakes up at time  $S$ .

#### Input Specification

- The first line consists of two integers  $N$  and  $Q$ .
- The second line contains the closing times of the attractions:  $c_1, c_2, c_3, \dots, c_N$ .
- The third line contains the visit times of the attractions:  $t_1, t_2, t_3, \dots, t_N$ .
- The next  $Q$  lines each contain two integers  $V$  (the minimum number of attractions to visit) and  $S$  (the time Mary wakes up).

#### Output Specification

- The output format requires printing YES or NO for each question.

#### Sample Input:

```
5 5
3 5 7 9 12
4 2 3 3 8
1 5
1 6
3 3
4 2
5 1
```

#### Sample Output:

```
YES
NO
YES
YES
NO
```

### Problem 4: Sun and Moon

You recently missed an eclipse and are waiting for the next one! To see any eclipse from your home, the

sun and the moon must be in alignment at specific positions. You know how many years ago the sun was in the right position, and how many years it takes for it to get back to that position. You know the same for the moon. When will you see the next eclipse?

**Input Specification:**

The input consists of two lines.

The first line contains two integers,  $d_s$  and  $y_s$  ( $0 \leq d_s < y_s \leq 50$ ), where  $d_s$  is how many years ago the sun was in the right position, and  $y_s$  is how many years it takes for the sun to be back in that position. The second line contains two integers,  $d_m$  and  $y_m$  ( $0 \leq d_m < y_m \leq 50$ ), where  $d_m$  is how many years ago the moon was in the right position, and  $y_m$  is how many years it takes for the moon to be back in that position.

**Output Specification:**

Output a single integer, the number of years until the next eclipse. The data will be set in such a way that there is not an eclipse happening right now and there will be an eclipse within the next 5000 years.

**Sample Input:**

3 10

1 2

**Sample Output:**

7

**Notes:**

The number of test cases  $t$  will be  $1 \leq t \leq 100$

## Problem 5: Training

Ashley is training for a programming contest on Brandon's Online Judge. Brandon's Online Judge has a new feature which allows Ashley's coach, Tom, to load a list of problems for Ashley.

Tom has selected some problems for Ashley to work on. Each problem has two integers representing a lower skill bound and an upper skill bound that can work on the problem. Each programmer has an integer skill level. If someone with a skill level between the lower and upper bounds of a problem (inclusive) solves a problem, then his/her skill level goes up by 1. A programmer never works on problems that are outside their skill range.

Ashley will train on Tom's curated list of problems as follows – she will look at the first problem on the list and either solve it or skip it. She will repeat this for every problem on the list in the order Tom loaded the problems. Once she has skipped a problem, she can never go back to it.

Compute the maximum skill level Ashley can have if she chooses to solve or skip problems optimally.

### Input Specification:

The first line contains two integers  $n$  and  $s$  separated by one space ( $1 \leq n \leq 10^5$ ,  $0 \leq s \leq 10^9$ ), where  $n$  is the number of problems Tom has curated for Ashley, and  $s$  is Ashley's current skill level.

Each of the next  $n$  lines contains two integers  $l$  and  $r$  ( $0 \leq l \leq r \leq 2 \cdot 10^9$ ). These are the lower ( $l$ ) and upper ( $r$ ) skill bounds on each of Tom's problems, in the order that Tom loaded them.

### Output Specification:

Output a single integer, which is the maximum skill level Ashley can attain.

### Sample Input:

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

### Sample Output:

```
2
```

## Problem 6: Open A Lock



You are part of an elite red team trying to break into a client's building. In front of you is a lock with four circular wheels. Each wheel has 10 slots: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'. Each wheel can rotate freely and wrap around: for example, we can turn from a 9 to a 0 or a 0 to a 9. Each move consists of turning one wheel one slot.

This is a very sophisticated lock, and you know there are certain combinations that will trigger an alarm, freeze the lock, or worse, so you never enter those. If you have a target combination you think will work, you need to know what the minimum number of tries will be so you can plan your time.

The length of deadends codes will be in the range [1, 500].

### Input Specification:

The first line is the number of independent test cases integer  $t$  ( $1 \leq t \leq 100$ )

Each test case will be given as follows:

A line containing a single integer  $n$  – the number of deadends ( $0 \leq n \leq 1000$ )

$n$  lines, each containing a four-digit deadend code

And lastly a final line containing the four-digit target code

### Output Specification:

You are to print a single integer showing either the number of turns to open the lock or  $-1$  if impossible.

**Sample Input:**

3  
5  
0201  
0101  
0102  
1212  
2002  
0202  
1  
8888  
0009  
8  
8887  
8889  
8878  
8898  
8788  
8988  
7888  
9888  
8888

**Sample Output:**

6  
1  
-1

**Explanation:**

6 - A sequence of valid moves would be "0000" -> "1000" -> "1100" -> "1200" -> "1201" -> "1202" -> "0202"

1 - We can turn the last wheel in reverse to move from "0000" -> "0009"

-1 - Can't reach target without getting stuck

**Notes:**

1. Target code will not be in the list of deadend codes
2. Every string in deadends and the string target will be a string of 4 digits from the 10,000 possibilities '0000' to '9999'.
3. '0000' will not be a deadend code
4. The number of test cases will be  $1 \leq t \leq 100$