# HKUST Local Programming Contest 2018

Room 4213, HKUST
22, September 2018

Coach: CHEN Qifeng
Problem Setter: ZHANG Haoqian
Judges: ZHANG Haoqian

# A. Calendar

Time Limit: 1 Second
Memory Limit: 128MB

When John was a small kid, he wanted to create a calendar. He found out that one thing he had to know is that, for a given day, what is this day of the week. For example, $22^{nd}$ Sep 2018 is Saturday. However, it was hard for him to calculate on paper. Could you write a program to help him?

John wanted to remind you about the leap year. In each leap year, the month of February has 29 days instead of 28. Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but these centurial years are leap years if they are exactly divisible by 400. For example, the years 1700, 1800, and 1900 were not leap years, but the years 1600 and 2000 were.

***Input:***
The first line contains an integer T(1<=T<=50), the number of test cases.
For each test case:
Each case contains three integer numbers d(1<=d<=31), m(1<=m<=12) and y(1<=y<=10000) in a single line, representing the day, month and year respectively.

***Output:***
For each test case, output a single number that represents day of week in a single line. We assume that 0 will represent Sunday, 1 Monday, 2 Tuesday, …, 6 Saturday.

***Sample Input:***
1
22 9 2018

***Sample Output:***
6

## B. Maze

Time Limit: 1 Second
Memory Limit: 128MB

You are trapped in a maze. Luckily, you have the map of the maze. The maze is a N×M matrix. Character S indicates your position and Character E indicates the exit door of the maze. The exit door of the maze requires k different type of keys to open. You can find all positions of keys on the map, which are labeled with the numbers from 0 to k-1, indicating the type of the key.

Your task is to find a plan with the least time to escape the maze. Each time unit, you can move one step up, down, left or right in the maze. Of course, you can not cross the border of the maze. Once you get to a position with a key, it takes no extra time to collect the key. Also, once you get to the exit door, it takes no extra time to open the door.

There are two more things you need to know. Character # indicates an unsafe area, therefore you can not enter that area. Character $ indicates a magic gate, where you can move to any magic gate without any cost. The number of magic gates will not exceed 10.

***Input:***

The first line contains an integer T(1<=T<=10), the number of test cases.
For each test case:
The first line contains three integer numbers, N, M (1<=N,M<=200) and K (1<=K<=5).
In the next N lines, ith line contains M characters, representing the ith row of the matrix.

***Output:***

For each test case:
Output the minimal time to escape the maze in one line. If there is no way to escape the maze, please output oop! in one line.

***Sample Input:***

```
1
7 8 2
........
..S..#0.
.##..1$.
.0#.....
...1#...
...##E$.
...1....
```

***Sample Output:***

```
8
```

# C. Connection

Time Limit: 1 Second
Memory Limit: 128MB

The city C has N villages. The mayor of the city hopes to establish a network so that any village can communicate directly or indirectly with other villages. There are M plans. Each plan will directly connect two villages with a cost. Some plans have already been implemented, so there is no way to cancel them.

The mayor wants to minimize the cost. Could you help the mayor to calculate the minimal total cost to build the network?

***Input:***
The first line contains two integer numbers N and M (1<=N<=2000, 1<=M<=10000)
For the next M lines, each line contains four integer numbers p, u, v and w, representing each plan. If p=1, it means that the plan has already been implemented, therefore you have to add the cost of that plan. If p=2, it means that the plan has not yet been implemented, so you can decide whether to implement it or not. The plan would connect village u and village v with cost w.
The input data will guarantee the existence of the solution.

***Output:***
Output the minimal total cost to build the network. The answer will not exceed $10^9$.

***Sample Input:***
```
5 6
1 1 2 1
1 2 3 1
1 3 4 1
1 4 1 1
2 2 5 10
2 2 5 5
```

***Sample Output:***
```
9
```

# D. Signal

Time Limit: 1 Second
Memory Limit: 128MB

Tom received an important signal from outer space. The signal was presented by an array of N integer numbers, a[1], a[2] … a[N]. To prevent someone from secretly modifying the signal, the smart Tom thought that a signature of the signal could be generated.

The signature is generated by the following:

Generating two integers l and r randomly between 1 to N. If r is greater than l, then exchanging l and r. The signature contains three real numbers. They are the expectations of the XOR, AND and OR summation of the numbers from the lth to the rth in the signal. XOR (AND, OR) summation refers to successive XOR (AND, OR) operations on integers.

However, Tom only knows a $O(N^2)$ algorithm to calculate the signature, which is not fast enough. Could you help Tom to generate the signature faster?

***Input:***

The first line contains a single integer N (1<=N<=100000)
The second line contains N integer numbers, a[1], a[2] … a[N] ($1<=a[i]<=10^9$)

***Output:***

Output three real numbers in one line with a space between two numbers. Your answer should be rounded to three digits after the decimal point.

***Sample Input:***

2
4 5

***Sample output:***

2.750 4.250 4.750

***Explanation:***

There are 4 different pairs of l and r.

| (l, r) Before Exchanging | XOR Summation | AND Summation | OR Summation |
|---|---|---|---|
| (1,1) | 4 | 4 | 4 |
| (1,2) | 1 | 4 | 5 |
| (2,1) | 1 | 4 | 5 |
| (2,2) | 5 | 5 | 5 |

Expectation of XOR Summation: (4+1+1+5)/4=2.75
Expectation of AND Summation: (4+4+4+5)/4=4.25
Expectation of OR Summation: (4+5+5+5)/4=4.75

# E. Random

Time Limit: 1 Second
Memory Limit: 128MB

Have you ever wondered how computer generates random numbers? Here we are going to introduce you the Linear congruential generator algorithm.

The algorithm is defined by recurrence relation:

$$X_{n+1} = (aX_n + c)\ mod\ m$$

Your task is to calculate the nth random number by given $X_0$, a, c and m. Since we only need a random number between 0 and p-1, please give us the $X_n$ mod p.

***Input:***
Only one line contains 6 integer numbers. They are m($1<=m<=10^9$), a, c, $X_0$($0<=a, c, X_0<=10^9$), n ($1<=n<=10^{18}$) and p ($1<=p<=10^9$), respectively.

***Output:***
Only one number, the $X_n$ mod p

***Sample Input:***
11 8 7 1 5 3

***Sample Output:***
2

***Explanation:***
$X_0 = 1$
$X_1 = (8*1+7) \bmod 11=4$
$X_2 = (8*4+7) \bmod 11=6$
$X_3 = (8*6+7) \bmod 11=0$
$X_4 = (8*0+7) \bmod 11=7$
$X_5 = (8*7+7) \bmod 11=8$
Answer = $X_5 \bmod 3=2$

## F. Card

Time Limit: 1 Second
Memory Limit: 128MB

A typical deck of playing cards consists of 54 cards. It has 4 suits and two special cards, called jokers. The 4 suits are Spades, Hearts, Diamonds and Clubs. Each suit contains 13 cards: Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King.

Tom wants to collect p1 Spades cards, p2 Hearts cards, p3 Diamonds cards and p4 Clubs cards. A joker card can be considered to be any suit card. Each time unit, tom will randomly pick up one card from the desk and collect that card without putting back to the desk. If the card is a joker card, Tom will consider it as one suit card and put it into that suit collection, so that the expectation of collecting time will be minimized. Could you tell Tom what is the expectation of time needed?

***Input:***
Only one line contains 4 integer numbers, p1, p2, p3 and p4 (0<=p1, p2, p3, p4<=15).

***Output:***
Output the expectation of time needed in one line. Your answer should be rounded to three digits after the decimal point.
If it is impossible to complete the collection task, please output -1.000 in one line.

***Sample Input1:***
13 14 14 13

***Sample Output1:***
54.000

***Sample Input2:***
15 15 15 15

***Sample Output2:***
-1.000

***Sample Input3:***
1 2 3 4

***Sample Output3:***
16.393

## G. Prime

Time Limit: 1 Second
Memory Limit: 128MB

The easiest problem is at the end! Hoping all of you can pass this problem. The problem is very easy. Please output how many prime numbers in a given interval [l,r].

In case you forget what is a prime number, here is its definition:

A prime number (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers. A natural number greater than 1 that is not prime is called a composite number. For example, 5 is prime because the only ways of writing it as a product, 1 × 5 or 5 × 1, involve 5 itself. However, 6 is composite because it is the product of two numbers (2 × 3) that are both smaller than 6.

If you still have no idea, l and r are small, which means you can manually compute them on paper and then write them in your code!

Enjoy!

***Input:***
Only one line contains 2 integer numbers, l and r (1<=l<=r<=500).

***Output:***
output how many prime numbers in one line.

***Sample Input:***
2 5

***Sample Output:***
3