

TakeOff Programming Contest

Preliminary Round

Fall - 2022, Slot: C

Judging Director: Abu Saleh

Associate Judging Director: Masum Bin Hossain

Slot C Judging Director: Anisur Rahman Sunny

Slot C Associate Judging Director: Nesar Ahmed

Serial	Problem Name	Problem Setter	Reviewer
A	Inception	Md. Mehedi Hasan Naim	Farhin Khaled
B	Equability	Farhin Khaled	Nesar Ahmed
C	A Cricket Match	MD Nabid Anzum Akash	Md. Minhajul Islam
D	Line Segment	Md. Minhajul Islam	MD Nabid Anzum Akash
E	Lucky Numbers V1	Md. Minhajul Islam	Nesar Ahmed
F	Palindrome Revelation	Nesar Ahmed	Md. Minhajul Islam
G	Milky Way Galaxy	Nesar Ahmed	Anisur Rahman Sunny
H	Lucky Pairs V2	Anisur Rahman Sunny	Nesar Ahmed

A. Inception

Category: Giveaway

Problem Analysis:

- Observation: This problem has no input. You have to print a single line: "I believe, I can be a good programmer." without quotes.

- Solution Idea: You just need to print the particular line by using printf function.

Special Thanks: Nesar Ahmed

Code:

```
#include<stdio.h>
int main(){
    printf("I believe, I can be a good programmer.");
    return 0;
}
```

B. Equability

Category: Simple If else

Problem Analysis:

- If the result of adding the values of A and B is equal to K then the output will be "Equal" without quotes.
- otherwise if the result of adding A and B is greater than K then the output will be "Greater" without quotes.
- and if the result of adding A and B is less than K then the output will be "Less" without quotes.

Code:

```
#include <stdio.h>
int main(){
    int a, b, k;
    scanf("%d %d %d", &a, &b, &k);
    if (a + b == k) printf("Equal");
    else if (a + b > k) printf("Greater");
    else printf("Less");
    return 0;
}
```

C. A Cricket Match

Category: Simple Math

Problem Analysis:

- Observation: In an ODI cricket match the total over is 50, so the total ball is 300.
- Remaining ball is: 300 - (played ball which is given as input).

- Now you have to convert the remaining ball: over and ball count.
- For over: remaining ball divided by 6.
- For ball: remaining ball mod by 6.

Special Thanks: Md. Minhajul Islam

Code:

```
#include<stdio.h>
int main() {
    int played_ball;
    scanf("%d", &played_ball);
    int remaining_ball = 300 - played_ball;
    int over_cout = remaining_ball / 6;
    int ball_count = remaining_ball % 6;
    printf("%d Over and %d Ball remaining\n", over_cout, ball_count);
    return 0;
}
```

D.Line Segment

Category: Nested If - Else

Problem Analysis:

- 1. Firstly we will handle the equal conditions. If x1 is equal to x2 and y1 is equal to y2 then the segments are **Equal**.
- 2. If it is not equal then we have to check if the x2 and y2 segment is inside the x1 and y1 segment. So we need to check if $x1 \leq x2$ and $y2 \leq y1$, (we have already checked $x1 = x2$ and $y1 = y2$ in the previous condition) if this condition is true then x2 and y2 segments are inside the x1 and y1 segment. So we will print **Inside**
- 3. If it is not Equal and Inside then we will print **Nothing**

Special Thanks: MD Nabid Anzum Akash

Code:

```
#include<stdio.h>

int main()
{
    int x1,y1,x2,y2;
    scanf("%d %d %d %d",&x1,&y1,&x2,&y2);

    if(x1==x2 && y1==y2) printf("Equal\n");
    else if(x1<x2 && y1>y2) printf("Inside\n");
```

```
else if(x1==x2 && y1>y2) printf("Inside\n");
else if(x1<x2 && y1==y2) printf("Inside\n");
else printf("Nothing\n");

return 0;
}
```

E.Lucky Numbers V1

Category: Loop

Problem Analysis:

- **Observation:** Currently he is in N'th stairs. His destination is M'th stairs. If he is in i'th stair he can move to (i+1)'th stair. So he will visit the stairs n,n+1,n+2.....m-2,m-1,m. You just need to check if his stair number is divisible by 3 or 5, if it is divisible by 3 or 5 then the stair number is a lucky number and you will count that to the answer.
- **Solution Idea:** You need a counter which will be initialized as 0. You need to simply loop through N to M. If the current stair number is divisible by 3 or 5 then you will increment the counter by 1 (counter = counter +1). After the loop ends, you will print the counter which will be the number of lucky numbers he had encountered.

Special Thanks: Nesar Ahmed

Code:

```
#include<stdio.h>

int main()
{
    int n,m;
    scanf("%d %d",&n,&m);

    int i,counter=0;
    for(i=n;i<=m;i++){

        if(i%3==0 || i%5==0){
            counter++;
        }
    }

    printf("%d\n",counter);

    return 0;
}
```

F. Palindrome Revelation

Category: String

Problem Analysis:

- First, you need to understand the concept of palindromes, which is: if you reverse a string, the reversed string is exactly the same as the previous string.
- Second, there is a condition that the concept of palindrome does not have uppercase and lowercase effects.
- Solution Idea: You just have to convert the full string in lowercase or uppercase then check if it's the same or not after reversing the full string.

Special Thanks: Md. Minhajul Islam

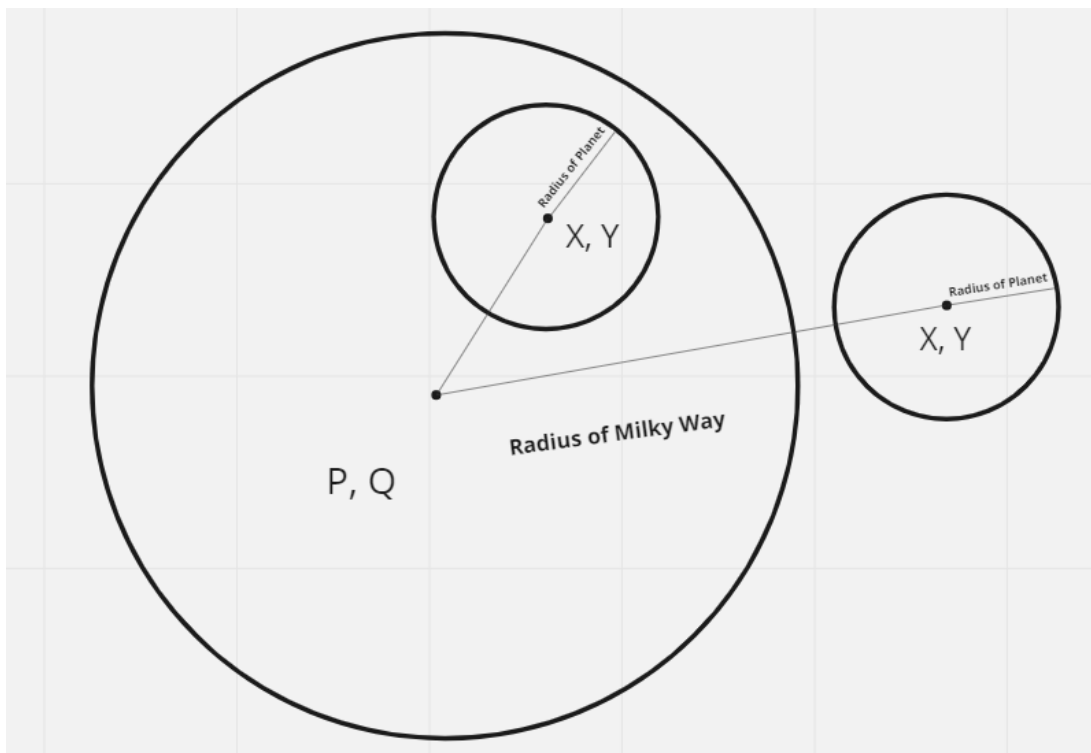
Code:

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    char str[1001]; //max limit is 1000
    scanf("%s", str);
    for(int i = 0; i < n; i++){
        if(str[i] >= 'A' && str[i] <= 'Z'){
            str[i] = str[i]+32; //converting to lowercase
        }
    }
    char reversed[1001];
    for(int j = 0; j < n; j++){
        reversed[j] = str[(n-1)-j]; //reversing the string
    }
    int palindrome = 1;
    for(int k = 0; k < n; k++){
        if(str[k] != reversed[k]){
            palindrome = 0; // comparing reversed with str
            break;
        }
    }
    if(palindrome) printf("Yes");
    else printf("No");
    return 0;
}
```

G. Milky Way Galaxy

Category: Geometry

Problem Analysis:



- From the above picture we can see all values are given. We just have to calculate if a planet is inside of the Milky Way Galaxy or outside.
- Solution Idea: If the distance between the milky way galaxy central coordinate and planets central coordinate + radius of planet is less than or equal to milky way galaxy's radius then its inside otherwise outside.

Special Thanks: Anisur Rahman Sunny

Code:

```
#include<stdio.h>
#include<math.h>
int main() {
    int rm, p, q, n, inside = 0;
    scanf("%d %d %d", &rm, &p, &q);
    scanf("%d", &n);
```

```

for(int i = 0; i < n; i++){
    int rp, x, y;
    scanf("%d %d %d", &rp, &x, &y);
    double d = sqrt(pow(p-x, 2) + pow(q-y, 2));
    d += rp;
    if(d <= rm) inside++;
}
printf("%d", inside);
return 0;
}

```

H. Lucky Number V2

Category: Number theory

Problem Analysis:

Analysis Condition:

- 1st Condition GCD will be more than 1. so we can find that there are lots of pairs ($1 \leq a, b \leq N$) in which pairs(a,b) have at least one common prime.
- 2nd condition b must be divisors by a. According to this condition, we can say that ($1 < b \leq B \leq N$) for every b, a will be divisors of b.

Assumption:

we need to find the number of divisors, for every particular ($1 \leq b \leq N$). we know finding all divisor complexity $O(\sqrt{n})$. BT here has test case 10^6 .

Total Complexity=($10^6 * O(\sqrt{n})$) it's huge need for optimization.

We can pre calculate all numbers divisors, and store them as a cumulative sum.

Such as,

DC= Divisor Count

index 1=DC(1)

index 2=index [1]+DC(2)

index 3=index 2+DC(2)

index 4=index 3+DC(3)

.

.

index n = index (n-1)+DC(n)

In Every test case, take input as N and print **Index N.

Note: **Take care of 1. Every number has a divisor 1 but gcd(1, itself) does not satisfy **1st** condition.

Solution:

Code:

```
#include <stdio.h>
#include<stdlib.h>
const int maxn =1e7+10;
int div_cnt[maxn],ans[maxn];
int gcd(int a, int b) {if (b == 0)return a; return gcd(b, a % b);}
int lcm(int a, int b) {a = abs(a); b = abs(b); return (a / gcd(a, b)) * b;}
void pre_cal(){
    for (int i=2;i<maxn;i++)
    {
        for (int j=i;j<maxn; j+=i)div_cnt[j]++;
    }
    for (int i=1;i<maxn;i++)ans[i]+=(ans[i-1]+div_cnt[i]);
}
int main()
{
    #ifdef ONLINEJUDGE
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif
    pre_cal();
    int t; scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        printf("%d\n",ans[n] );
    }
}
```



```
}  
return 0;}
```