

TCS NQT Coding Questions

Numbers series Coding questions are one of the types of question that were asked in the recent TCS 2018 drive. Those coding questions will be of a mixture of two series where odd terms of one kind and the even terms of the other kind. For Example, **Consider the given series: 1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, ...**

This series is a mixture of 2 series – all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order. Now write a program to find the Nth term in this series.

The above coding question was asked in the recent TCS drive 2018 which happened in the month of September 2018.

TCS Coding questions – Important instructions

Instruction for 2020

- 1) One question, 15 minutes another Question 30 minutes
- 2) Choice of C / C++ / Java / Perl / Python 2.7.
- 3) Provided an IDE to debug.
- 4) For Java, the class name should be named Maze.
- 5) The input to the program either through STDIN / Command line arguments, as per the instructions.
- 6) The program should write the output to STDOUT.
- 7) Public and private test cases based evaluation.

Points to note

- 1) While printing the output no leading or trailing spaces should be printed.
- 2) Other than the required output, no other text should be printed.
- 3) If the output is a number, no leading sign must be printed unless it is a negative number.

4) No scientific notation (3.9265E + 2).

5) All floating point numbers must contain that many decimal places as mentioned in the question.

TCS Coding question – 1

Consider the following

series: 1,1,2,3,4,9,8,27,16,81,32,243,64,729,128,2187...

This series is a mixture of 2 series – all the odd terms in this series form a geometric series and all the even terms form yet another geometric series.

Write a program to **find the Nth term** in the series.

The value N is a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term, no other character/string or message should be written to STDOUT. For example, if N=16, the 16th term in the series is 2187, so only value 2187 should be printed to STDOUT.

You can assume that N will not exceed 30.

```
C
#include"stdio.h"
#include"math.h"
int main()
{
//code
int n;
scanf("%d", &n);
if(n % 2 == 1)
{
int a = 1;
int r = 2;
int term_in_series = (n+1)/2;
int res = pow(2, term_in_series - 1);
printf("%d ", res);
}
else
{
int a = 1;
int r = 3;
int term_in_series = n/2;
```

```
int res = pow(3, term_in_series - 1);
printf("%d ", res);
}
return 0;
}
```

Input: 16

Output: 2187

TCS Coding question – 2

Consider the following

series: 0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8

This series is a mixture of 2 series all the odd terms in this series form even numbers

in ascending order and every even term is derived from the previous term using the formula $(x/2)$.

Write a program to find the nth term in this series.

The value n is a positive integer that should be read from STDIN the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.

For example, if $n=10$, the 10 th term in the series is to be derived from the 9th term in the series. The 9th term is 8 so the 10th term is $(8/2)=4$. Only the value 4 should be printed to STDOUT.

You can assume that the 'n' will not exceed 20,000.

```
C
#include "stdio.h"
#include "math.h"
int main()
{
//code
int n;
scanf("%d", &n);
if(n % 2 == 1)
{
int a = 1;
```

```

int r = 2;
int term_in_series = (n+1)/2;
int res = 2 * (term_in_series - 1);
printf("%d ", res);
}
else
{
int a = 1;
int r = 3;
int term_in_series = n/2;
int res = term_in_series - 1;
printf("%d ", res);
}
return 0;
}

```

Input: 10

Output: 4

TCS Coding question – 3

Write a c program, to check whether the given year is a leap year or not using command line arguments. A leap year is a calendar year containing one additional day (Feb 29th) added to keep the calendar year synchronized with the astronomical year.

```

c
#include<stdio.h>
int main(int a, char*b[])
{
int year; year=atoi(b[1]);
if(year%100==0)
{
if(year%400==0)
{
printf("LEAP YEAR");
}
else{
printf("NOT LEAP YEAR"); } }
else if(year%4==0)
{
printf("LEAP YEAR");
}
else{

```

```
printf("NOT LEAP YEAR");  
}  
return 0; }
```

TCS Coding question – 4

Write a c program, to find the GCD of the given 2 numbers, using command line arguments. The input is 2 integer and the output GCD also should be an integer value.

```
C  
#include<stdio.h>  
int main(int x, char *y[])  
{  
    int a,b,small,i;  
    a=atoi(y[1]);  
    b=atoi(y[2]);  
    small=a>b?b:a;  
    for(i=small;i>=1;i--)  
    {  
        if((a%i==0)&&(b%i==0))  
        {  
            printf("%d",i);  
            break;  
        }  
    }  
    return 0;  
}
```

TCS Coding question – 5

C Program to check whether a given number is a prime number or not. The given number N, a positive integer, will be passed to the program using the first command line parameter. If it is a prime number the output should be the square root of the number up to 2 decimal point precision, If it is not a prime number then print 0.00 to stdout

```
C  
#include<stdio.h>  
#include<math.h>  
int main(int a, char *b[])  
{  
    int number,i,flag = 1;  
    number = atoi(b[1]);  
    for(i=2; i<number; i++)  
    {
```

```

if(number%i == 0)
{
flag = 0;
break;
}
}
if(flag == 1)
printf("%.2f",sqrt(number));
else
printf("0.00");
return 0;
}

```

TCS Coding question – 6

C Program to check whether a given number is a strong number or not. The given number N, a positive integer, will be passed to the program using the first command line parameter. If it is a strong number, the output should be "YES", If it is not a prime number then output should be "NO" to stdout. Other than YES or NO, no other extra information should be printed to stdout.

```

C
#include<stdio.h>
#include<math.h>
int main(int a, char *b[])
{
int number, i, temp, sum = 0, factorial = 1;
number = atoi(b[1]);
temp = number;
while(number != 0)
{
int rem = number%10;
for(i=2; i<=rem; i++)
{
factorial = factorial * i;
}
sum = sum + factorial;
number = number/10;
factorial = 1;
}
if(temp == sum)
printf("YES");
else

```

```
printf("NO");  
return 0;  
}
```

TCS Coding question – 7

Write a C program which will convert a given decimal integer number N to its binary equivalent. The given number N, a positive integer, will be passed to the program using the first command line parameter. Print the equivalent binary number to stdout. Other than the binary number, no other extra information should be printed to stdout Example: Given input "19", here N=19, expected output 10011

```
C  
#include<stdio.h>  
#include<math.h>  
int main(int a, char *argv[])  
{  
    int number, count, i;  
    int b[32];  
    number = atoi(argv[1]);  
    count = 0;  
    while(number != 0)  
    {  
        b[count]=number%2;  
        number = number/2;  
        count++;  
    }  
    for(i=(count-1); i>=0; i--)  
        printf("%d", b[i]);  
    return 0;  
}
```

TCS Coding question – 8

Write a c program that will find the sum of all prime numbers in a given range. The range will be specified as command line parameters. The first command line parameter, N1 which is a positive integer, will contain the lower bound of the range. The second command line parameter N2, which is also a positive integer will contain the upper bound of the range. The program should consider all the prime numbers within the range,

excluding the upper bound and lower bound. Print the output in integer format to stdout. Other than the integer number, no other extra information should be printed to stdout. Example Given inputs "7" and "24" here N1= 7 and N2=24, expected output as 83.

```
C
#include<stdio.h>
int main(int argc, char *argv[])
{
    int N1, N2, j, i, count, sum = 0;
    N1 =atoi(argv[1]);
    N2 =atoi(argv[2]);
    for(i=N1+1; i<N2; ++i)
    {
        count = 0;
        for(j=2; j<=(i/2); j++)
        {
            if(i%j==0)
            {
                count++;
                break;
            }
        }
        if(count==0)
            sum = sum + i;
    }
    printf("%d",sum);
    return 0;
}
```

TCS Coding question – 9

Write a C program to check whether the given number is a perfect square or not using command line arguments.

```
C
#include<stdio.h>
#include<math.h>
int main(int a, char *b[])
{
    int n, i;
    n= atoi(b[1]);
    for(i = 0; i <= n; i++)
    {
        if(n == i * i)
    }
```



```

{
printf("YES");
return 0;
}
}
printf("NO");
return 0;
}

```

TCS Coding question – 10

Write a C program to check whether the given number is Palindrome or not using command line arguments.

```

C
#include<stdio.h>
#include<math.h>
int main(int a,int *b[])
{
int number, rem, sum = 0;
number = atoi(b[1]);
int copy = number;
while(number != 0)
{
rem =number%10;
sum = sum * 10 + rem;
number = number/10;
}
if(copy == sum)
printf("Palindrome");
else
printf("Not Palindrome");
return 0;
}

```

TCS Coding question – 11

Write a C program to convert the vowels to an uppercase in a given string using command line arguments.

Example: if the input is tata, then the expected output is tAtA.

```

C
#include<stdio.h>
int main(int argc, char *argv[])
{
char *str = argv[1];
int i;

```

```

for(i =0; str[i] !='\0'; i++)
{
if(str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u')
{
str[i] = str[i] - 32;
}
}
printf("%s", str);
return 0;
}

```

TCS Coding question – 12

Write a C program to find the hypotenuse of a triangle using command line arguments

```

C
#include<stdio.h>
int main(int a, char*b[])
{
float hyp;
int opp=atoi(b[1]);
int adj=atoi(b[2]);
hyp=sqrt((opp*opp)+(adj*adj));
printf("%0.2f",hyp);
return 0;
}

```

TCS Coding question – 13

Write a C program to find whether the given number is an Armstrong number or not using command line arguments.
An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3 = 371$.

```

C
#include<stdio.h>
#include<math.h>
int main(int a, char*b[])
{
int n;
n= atoi(b[1]);
int sum=0;
int temp=n;

```

```

int cnt=0;
while(n!=0)
{
n=n/10;
cnt++;
}
n=temp;
while(n!=0)
{
int rem=n%10;
sum=sum+pow(rem,cnt);
n=n/10;
}
if(temp==sum)
{
printf("yes");
}
else
{
printf("no");
}
return 0;
}

```

TCS Coding question – 14

Write a program to generate Fibonacci Series.

```

C
#include<stdio.h>
#include<math.h>
int main(int a, char *b[])
{
int i, n, t1 = 0, t2 = 1, nextTerm;
n=atoi(b[1]);
for (i = 1; i <= n; ++i)
{
printf("%d ", t1);
nextTerm = t1 + t2;
t1 = t2;
t2 = nextTerm;
}
return 0;
}

```

TCS Coding question – 15

Calculate length of hypotenuse of right-angled triangle.

```
C
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
if(argc<2)
{
printf("please use \"prg_name value1 value2 ... \"\n");
return -1;
}
int a,b,side1,side2,side3;
a=atoi(argv[1]);
b=atoi(argv[2]);
side1=pow(a,2);
side2=pow(b,2);
side3=sqrt((side1+side2));
printf("the hypotenuse is %d",side3);
return 0;
}
```