TCS

Technical Round

Intro about the company

- Tata Consultancy Services Limited (TCS) is an Indian multinational information technology (IT) service, consulting and business solutions company Headquartered in Mumbai, Maharashtra.
- It is a subsidiary of the Tata Group and operates in 46 countries.
- TCS is now placed among the ‘Big 4’ most valuable IT services brands worldwide.
- TCS is one of the largest private sector employers in India, and the fourth-largest employer among listed Indian companies (after Indian Railways, Indian Army and India Post).

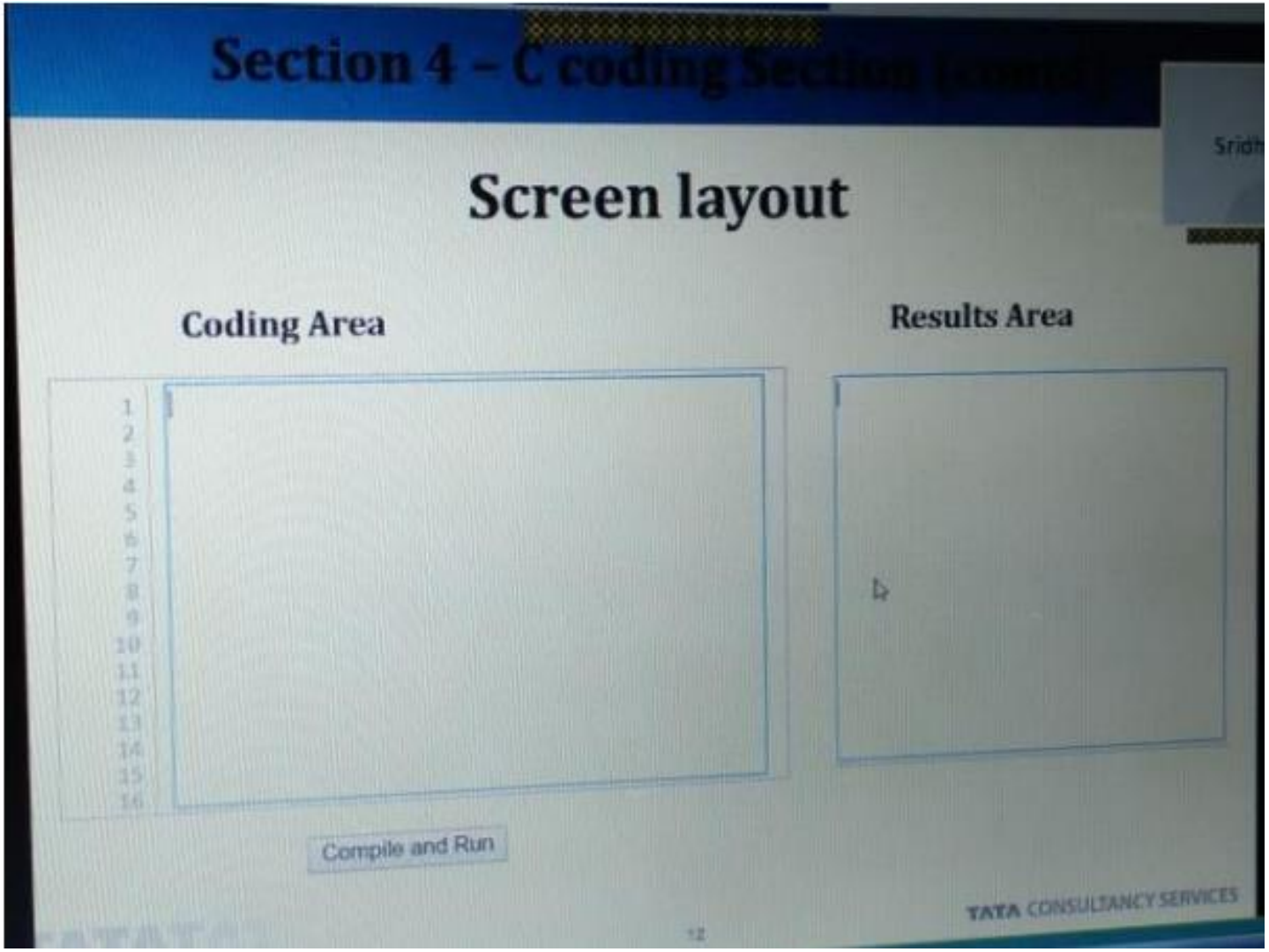
TCS Technical round

- There are two components of technical questions. MCQ's and Programming.

- MCQ (or) Fill in the blanks
  - 20 Minutes
  - 10 Questions

- Coding round
  - 20 Minutes
  - One C program

Coding platform



Coding platform - Instructions

- There is **only one question** for **20 minutes.**
- It has **10 attempts(We can compile only 10 times)**.
- We must **start our code from the scratch**.
- The coding platform is divided into two, one for writing the code and other for output. We **should write the whole program**.
- We **can't use any input functions like scanf(), getch(), getchar()**.
- The input to be provided should be read as **command line arguments.**

Coding platform - Instructions

- We must **only print exact output**.
- Output must not be re-framed by extra words.
- If there is any error, the error will be shown in the output dialog box.
- The errors are clearly mentioned.
- If there are no errors, a message like "compiled successfully" will be printed.
- Along with that they will mention **four test cases** are '**passed**' or '**failed**'. They are indicated like private and public test cases. They have not mentioned what is the test case, which is difficult to understand.
- There is **no time limit**. But, when all the 10 attempts are over, a message like "attempts exhausted" will be shown.
- To compile and run there is a button provided. To run the code, just click on that.

We **can't use any input functions like scanf(), getch(), getchar()**.

The input to be provided should be read as **command line arguments**.

int main(**int argc, char *argv[]**)

COMMAND LINE ARGUMENTS IN C:

main() function of a C program accepts arguments from command line or from other shell scripts by following commands.

They are argc and argv[]

where,

argc – Number of arguments in the command line including program name (integer).

argv[] – This is carrying all the arguments including program name (pointer array).

command line arguments

```
#include <stdio.h>
int main(int argc, char *argv[])   //command line arguments
{
   printf("\n Program name  : %s \n", argv[0]);
   printf("1st arg  : %s \n", argv[1]);
   printf("2nd arg  : %s \n", argv[2]);
return 0;
}
```

PROGRAMS

Program 1:

Factorial program in c using command line arguments.

Factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n.

For example, The value of 5! is 5*4*3*2*1 = 120

```
#include<stdio.h>
int main(int a, char *b[]) //command line arguments
{
        Int x,y,f=1;
        x=atoi(b[1]);
        for(i=1;i<=x;i++)
        {
            f=f*i;
         }
        printf("%d",f);
        return 0;

}
```

//**atoi** function is to convert a character to integer

Program 2:

Write a c program, to find the area of a circle when the diameter is given, using command line arguments.

The input diameter is an integer and the output area should be a floating point variable with 2 point precision

```
#include<stdio.h>
#define PI 3.14
int main(int a, char *b[]) //command line arguments
{
        int d;
        float area =0;
        d= atoi(argv[1]);
        area =(float) PI*(d/2)*(d/2);
        printf("%0.2f", area);
        return 0;
}
```

%0.2f is to print the answer with 2 values after decimal point.

Program 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.

```
#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;
}
```

Program 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.

```
#include<stdio.h>
int main(int x, char *y[])
{
inta,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;
}
```

Try more basic level programs using command line arguments

1) How many times the below loop will be executed?

```
#include<stdio.h>
int main() {
   int x, y;
   for(x=5;x>=1;x--)
   {
      for(y=1;y<=x;y++)
       printf("%d\n",y);
   }
}
```

Options

A. 15 

B. 11

C. 10

D. 13

2) Where the local variables are stored?

A. Disk

B. Stack 

C. Heap

D. 13

3) Select the missing statement?

```
#include<stdio.h>
long int fact(int n);
int main()
{
    \\missing statement
}
long int fact(int n)
{
   if(n>=1)
      return n*fact(n-1);
   else
      return 1;
}
```

Options

A. printf(“%ll\n",fact(5));

B. printf("%u\n",fact(5));

C. printf("%d\n",fact(5));

D. printf("%ld\n",fact(5)); ✓

4) Which of the following indicate the end of the file?

A. Feof()

B. EOF

C. Both feof() and EOF ✓

D. None of the mentioned

5) If a function's return type is not explicitly defined then it's default to _____ (In C).

A. int ✓

B. float

C. void

D. Error

6) For passing command line argument the main function should be like _____

A. int main(char *argv[], int argc)

B. int main(int argc)

C. int main(char *argv[])

D. int main(int argc, char *argv[]) ✓

7) How many times the below loop will be executed?

```
#include<stdio.h>
int main()
{
   int i;
   for(i=0;i<5;i++)
   {
      printf("Hello\n");
   }
}
```

Options

A. 5

B. 1

C. 0

D. 3

8) Which of the following is a User-defined data type?

A. long int

B. double

C. unsigned long int

D. enum ✓

9) Which has the highest precision?

A. float

B. double 

C. unsigned long int

D. Long int

Floating point types

The following table provide the details of standard floating-point types with storage sizes and value ranges and their precision −

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

10) What will be the output/error?(for input: 6, 9

```
#include<stdio.h>
int fg(int,int);
int main()
{
   int n1,n2,g;
   scanf("%d%d",
     &n1,&n2);
   g=fg(n1,n2);
   printf("%d",g);
}
int fg(int x,int y)
{
   while(x!=y)
   {
      if(x>y)
         return fg(x-y,y);
      else
         return fg(x,y-x);
   }
   return x;
}
```

Options

A. 3 

B. 6

C. 9

D. Error

Tips to Crack Test

- Never guess an answer
- Attend only the questions you are reasonably certain of answering correctly
- If you don’t get the answer in the first try then skip the question
- Read the question fully once and find out what they are asking
- Try not to panic at the sight of those long programming questions.
- Most programming questions are very simple which require you to do some **logical thinking** with a **clear mind**.

All The Best