INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Programming and Data Structures (CS10003)   **Spring Semester 2021-22**
Short Test-1   **Date: 5-May-2022**   **Marks=30**

___

Instructions:

1. *Write your answers on paper. Answers must be handwritten. Typed or written answers using an electronic device are not allowed.*
2. *Write your name and roll number on each page. Write page numbers for each page.*
3. *Answer all six questions. All parts of a question must be answered in the same place.*
4. *Scan all pages and collate. Create a single PDF file (of size <= 10 MB). You could also take pictures of different pages, combine them to make a single pdf file.*
5. *The name of the file should be <RollNumber>_Short_1. Ensure your roll number in the filename has only digits and uppercase characters.*
6. *Upload your file by clicking on 'Add Work' button. Make sure you click on the 'Turn in' button to submit your file.*
7. *7:10 PM is a strict deadline after which no submissions will be allowed.*
8. *We will not accept submission by any other means.*

___

1. What is the output of the following C program when the input N is the number formed by the last two digits of <u>your</u> roll number? Explain the execution.

```c
#include <stdio.h>
int main()
{
    int x, N;
    scanf("%d", &N);
    printf("%d\n", N);

    N /= 2;
    x = (N>15)?(N=4):(N=12);
    if (x < 10)
        if (x < 15) printf("A");
    else printf("B");
    printf("C");
}
```

[4 marks]

SOLUTION: For all roll numbers less than 32, the predicate N > 15 will be false, and therefore x will be assigned 12. Therefore, the program will print the value of N and then the letter C. For roll numbers greater than or equal to 32, x will be assigned 4. The program will print the value of N and then AC.

2. What is printed by the following program when the input N is the number formed by the last two
    digits of your roll number? Explain the execution.

    ```c
    #include <stdio.h>
    int main()
    {
        int i, j, N, count=0;
        scanf("%d", &N);
        printf("%d\n", N);

        N += 10;
        for (i=0; i< N; i++)
            for (j=i; j>=0; j--)
                count++;
        printf("%d\n", count);
    }
    ```

    [4 marks]

    SOLUTION: If N is the number formed by the last two digits of the roll number, then it prints N
    and then prints the value of (N +10)(N + 11)/2, that is, the sum of natural numbers up to N+10

3. Write a program that reads the volume, V, as a floating-point number. It then computes and
    prints the radius of a sphere having volume V, and then computes and prints the height of a
    cylinder having the same radius and the same volume V. You may use math library functions if
    necessary.

    [4 marks]

    SOLUTION:

    ```c
    #include <stdio.h>
    #include <math.h>

    #define PI 3.14159265359

    int main(){

      double V;
      double r_sphere, h_cylinder;

      printf("Volume = ");
      scanf("%lf", &V);

      r_sphere = cbrt(3*V/(4*PI));
      printf("Radius of a sphere with volume %g = %g\n", V, r_sphere);

      h_cylinder = V/(PI*r_sphere*r_sphere);
      printf("Height of a cylinder with radius %g and volume %g = %g\n",
                                          r_sphere, V, h_cylinder);

      return 0;
    }
    ```

4. What is printed by the following program when the input is the first two letters of your first name (given name) in capital letters? For example, if your first name is Jayant, then the input is JA. Explain the execution.

```
#include<stdio.h>
int main()
{
     char a, b;
     a = getchar(); b = getchar();
     printf("a = %c, b = %c\n", a, b);

     printf("%c\n", 'a'+ a - 'A');
     printf("%c\n", 'b' + b - 'B');
     printf("%d\n", a - 'A');
}
```

[4 marks]

SOLUTION: The first printf will print the lowercase of the first letter, and the second printf will print the lowercase of the second letter. This is because 'a' – 'A' and 'b' – 'B' are the same. The third printf will print the letter number (starting with zero) for the first input. That is, if the first input is A, then it will print 0, for B it will print 1, and for J it will print 9.

5. The following C program aims to print the number of days from the date 1-1-*y* to the date 1-*m*-*y*, where *y* is a positive integer representing a year and *m* is an integer in the range 1 to 12 representing month. The count should include 1-1-y but exclude 1-m-y. Unfortunately, the student who wrote this program made some mistakes. Rearrange the lines of code so that it fulfils its objective correctly. Do not add any new statement. Do not modify any arithmetic. Explain why the rearranged code is correct.

```
#include<stdio.h>
int main()
{
     int days=0, is_leap;
     int m, y;
     printf("Enter a month and year: ");
     scanf("%d %d", &m, &y);

     if ((m < 1) || (m >12) || (y < 1)){
          printf("Invalid input! \n");
          return 1;
     }

     // adding an extra day if the year is leap
     is_leap = (m>=3 && ((y%4 == 0 && y%100 != 0) || y%400 == 0));
     days= (is_leap)? (days+1):days;

     switch(m){
          case 2: days+=31;
          case 3: days += 28;
          case 4: days += 31;
          case 5: days += 30;
```

```
                case 6: days += 31;
                case 7: days += 30;
                case 8: days += 31;
                case 9: days += 31;
                case 10: days += 30;
                case 11: days += 31;
                case 12: days += 30;
        }

        printf("Number of days between 1-1-%d and 1-%d-%d = %d\n",
                                                y,m,y,days);

        return 0;
    }
```

[4 marks]

SOLUTION: Rewrite the switch statement as follows. Everything else remains the same.

```
            switch(m){
                    case 12: days += 30;
                    case 11: days += 31;
                    case 10: days += 30;
                    case 9: days += 31;
                    case 8: days += 31;
                    case 7: days += 30;
                    case 6: days += 31;
                    case 5: days += 30;
                    case 4: days += 31;
                    case 3: days += 28;
                    case 2: days+=31;
            }
```

Sample test cases

(i)   m=1, y=2022, days=0   (ii)   m=2, y=2022, days=31
(iii)   m=3, y=2022, days=59   (iv)   m=3, y=2020, days=60
(v)   m=3, y=2000, days=60   (vi)   m=3, y=2100, days=59

6. Write a C program which reads a sequence of positive integers till the user types -1. It counts the lengths of the non-decreasing sub-sequences, and prints the maximum among them.
   For example, for a given input {6, 7, 2, 29, 17, 5, 5, 11, 6, 7, 8, -1}, the non-decreasing sub-sequences are: {6, 7}, {2, 29}, {17}, {5, 5, 11} and {6, 7, 8}. Thus, the answer should be 3.
   Do not use an array. Assume that the first integer read is not -1 and a single integer is a sequence of length 1

[10 marks]

SAMPLE ANSWER

```c
int main ( ){
        int prevno, curno, curlength, maxlength ;
        curlength = 1; maxlength = 0;
        scanf ("%d", &prevno) ;
        scanf ("%d", &curno) ;
        while (curno != -1) {
                if (curno >= prevno)
                        curlength++;
                else {
                        if (curlength > maxlength)  {
                                maxlength = curlength;
                        }
                        curlength = 1;
                 }
                 prevno = curno ;
                 scanf ("%d", &curno) ;
        }
        if (curlength > maxlength)
                maxlength = curlength;
        printf ("Maximum length is %d\n", maxlength) ;
        return 0;
}
```