Assignment – 2
Introduction of programing based on C

1. **Research and provide three real-world applications where C programming is extensively used, such as in embedded systems, operating systems, or game development.**
   - **Embedded system: automotive software**
   - **Operating system: windows, linux**
   - **Game development: doom**

2. **Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.**
   - ```
     #include<stdio.h>
     Int main()
     {
     Printf("\n Hello World!");
     Return 0;
     }
     ```

3. **Write a C program that includes variables, constants, and comments. Declare and use different data types (int, char, float) and display their values.**
   - ```
     #include<stdio.h>
     #define pie w3.14
     Int main()
     {
     Int n1=5;
     Char a='A';
     Float n2=10.5;
     /* here take n1 for the any numerical value.
             Take n2 for the float value.
             Take a for any alphabetic value.
             Take pie is for constant.*/
     Printf("\n the value of n1=%d",n1);
     Printf("\n the value of n2=%f",n2);
     Printf("\n the value of a= %c",a);
     Printf("\n the value of pie=%.2f",pie);
     Return 0;
     }
     ```

4. Write a C program that accepts two integers from the user and performs arithmetic, relational, and logical operations on them. Display the results.

```
#include <stdio.h>
int main()
{
        int n1,n2;
        printf ("\n enter the value of n1= ");
        scanf ("%d",&n1);
        printf ("\n enter the value of n2= ");
        scanf ("%d",&n2);
        // Arithmatic opreation //

        printf("\n the addition of %d and %d is %d",n1,n2,n1+n2);
        printf("\n the multiplication of %d and %d is
%d",n1,n2,n1*n2);
        printf("\n the sutraction of %d and %d is %d",n1,n2,n1-n2);
        printf("\n the division of %d and %d is
%f",n1,n2,(float)n1/(float)n2);

        // Relational opreation //

        printf ("\n\n n1>n2 : %d",n1>n2);
        printf ("\n n1<n2 : %d",n1<n2);
        printf ("\n n1<=n2 : %d",n1<=n2);
        printf ("\n n1>=n2 : %d",n1>=n2);
        printf ("\n n1==n2 : %d",n1==n2);
        printf ("\n n1!=n2 : %d",n1!=n2);

        // Logical opreation //

        printf ("\n\n n1>0 && n2>0 : %d",n1>0 && n2>0);
        printf ("\n n1>0 | |  n2>0 : %d",n1>0 | |  n2>0);
        printf ("\n !(n1>0) : %d",!(n1>0));

        return 0;
}
```

5. **Write a C program to check if a number is even or odd using an if-else statement. Extend the program using a switch statement to display the** month name based on the user's input (1 for January, 2 for February, **etc.).**

```
#include <stdio.h>
int main()
{
        int n;
        up:
        printf ("\n enter the value of n = ");
        scanf ("%d",&n);

        if (n %2 == 0)
        {
                printf ("\n %d is even number",n);
        }
        else
        {
                printf ("\n %d is odd number",n);
        }
        switch (n)
        {
                case 1 :
                        printf ("\n %d for january",n);
                break;
                case 2 :
                        printf ("\n %d for febuary",n);
                break;
                case 3 :
                        printf ("\n %d for march",n);
                break;
                case 4 :
                        printf ("\n %d for april",n);
                break;
                case 5 :
                        printf ("\n %d for may",n);
                break;
                case 6 :
                        printf ("\n %d for june",n);
                break;
                case 7 :
```

```
                printf ("\n %d for july",n);
            break;
            case 8 :
                printf ("\n %d for august",n);
            break;
            case 9 :
                printf ("\n %d for september",n);
            break;
            case 10 :
                printf ("\n %d for october",n);
            break;
            case 11 :
                printf ("\n %d for november",n);
            break;
            case 12 :
                printf ("\n %d for december",n);
            break;
        }
        goto up;
        return 0;
}
```

6. **Write a C program to print numbers from 1 to 10 using all three types of loops (while, for, do-while).**

```
#include <stdio.h>
int main()
{
        int i;
        printf ("\n\n For loop :=");
        for (i=1;i<=10;i++)
        {
                printf ("\n %d",i);
        }
        printf ("\n\n While loop :=");
        i=1;
        while (i<=10)
        {
                printf ("\n %d",i);
                i++;
        }
        printf ("\n\n Do while loop :=");
```

```
        i=1;
        do
                {
                        printf ("\n %d",i);
                        i++;
                }
        while (i<=10);
        return 0;
}
```

7. **Write a C program that uses the break statement to stop printing numbers when it reaches 5. Modify the program to skip printing the number 3 using the continue statement.**

```
#include <stdio.h>
 int main()
{
   int num,i;
   printf("\n enter the number = ");
   scanf("%d",&num);
   for (i=1;i<=num;i++)
   {
      if (i%3==0)
      {
         continue;
      }
      if (i%5==0)
      {
         break;
      }
      printf("\n %d",i);
   }

   return 0 ;
}
```

8. Write a C program that calculates the factorial of a number using a function. Include function declaration, definition, and call.

```
#include <stdio.h>
//without return type with argument
void fact(int n1);//declaration
void fact(int n1)//defition
{
   int ans = 1,i;
   for (i=1;i<=n1;i++)
   {
      ans = ans*i;
   }
   printf ("\n the factorial of n1 = %d",ans);
}
int main()
{
   int num1;
   printf ("\n enter the number = ");
   scanf ("%d",&num1);
   fact(num1);//calling
   return 0;
}
```

9. Write a C program that stores 5 integers in a one-dimensional array and prints them. Extend this to handle a two-dimensional array (3x3 matrix) and calculate the sum of all elements.

```
#include <stdio.h>
int main()
{
   int c[5],i;
   for (i=0;i<5;i++)
   {
      printf ("\n enter the elementc[%d] = ",i+1);
      scanf ("%d",&c[i]);
   }
   printf ("\n array = ");
   for (i=0;i<5;i++)
   {
      printf ("%d ",c[i]);
   }
   int a[3][3],b[3][3],k,j,sum=0;
```

```
    for (k=0;k<3;k++)
    {
       for (j=0;j<3;j++)
       {
          printf ("\n enter the element a[%d][%d]= ",k,j);
          scanf ("%d",&a[k][j]);
       }
    }
    for (k=0;k<3;k++)
    {
       for (j=0;j<3;j++)
       {
          printf ("\n enter the element b[i][j]= ",k,j);
          scanf ("%d",&b[k][j]);
       }
    }
    printf ("\n 1st \t 2nd =\t sum \n");
    for (k=0;k<3;k++)
    {
       for (j=0;j<3;j++)
       {
          printf ("%d ",a[k][j]);
       }
            printf ("\t");
          for (j=0;j<3;j++)
          {
             printf ("%d ",b[k][j]);
          }
          printf ("\t");
          for (j=0;j<3;j++)
          {
             sum = a[k][j]+b[k][j];
             printf ("%2d ",sum);
          }
          printf ("\n");
    }

    return 0;
}
```

10. Write a C program to demonstrate pointer usage. Use a pointer to modify the value of a variable and print the result.

```
#include <stdio.h>
int main()
{
        int a[100];
        int *ptr = &a;
        printf ("\n enter the variable = ");
        scanf ("%d",&a);
        printf ("\n address of the value = %p",ptr);
        printf ("\n value of the variable = %d",*ptr);
        return 0;
}
```

11. Write a C program that takes two strings from the user and concatenates them using strcat(). Display the concatenated string and its length using strlen().

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str1[100],str2[100];

   printf ("\n enter the value in str1 = ");
   gets (str1);
   printf ("\n enter the value in str2 = ");
   gets (str2);

   printf ("\n original value of str1 = %s",str1);
   printf ("\n original value of str2 = %s",str2);

   strcat(str1,str2);
   printf ("\n the value of string after using concate function =
%s",str1);
   printf ("\n the value of string after using concate function =
%s",str2);

   int length = strlen(str1);
   printf ("\n the length of str1 = %d",length);
   printf ("\n the length of str2 = %d",length=strlen(str2));
   return 0;}
```

12. Write a C program that defines a structure to store a student's details (name, roll number, and marks). Use an array of structures to store details of 3 students and print them.

```
#include <stdio.h>
struct student
{
   char name[100];
   int mark;
   int roll;
};
int main()
{
   struct student s[10];
   int size,i;
   printf ("\n enter the size = ");
   scanf ("%d",&size);
   for (i=0;i<size;i++)
   {
      printf ("\n enter the details of students %d",i+1);
      printf ("\n name = ");
      scanf ("%s",&s[i].name);
      printf ("\n roll no. = ");
      scanf ("%d",&s[i].roll);
      printf ("\n marks = ");
      scanf ("%d",&s[i].mark);
   }
   printf ("\n name \t roll no. \t marks");
   for (i=0;i<size;i++)
   {
      printf ("\n %s \t %d \t\t %d",s[i].name,s[i].roll,s[i].mark);
   }
   return 0;
}
```

13. Write a C program to create a file, write a string into it, close the file, then open the file again to read and display its contents.

```
#include <stdio.h>

int main()
{
   FILE *fp1;
   char text[100];
   fp1 = fopen("first.txt","w");
   fprintf (fp1,"\nhello world!");
   fprintf (fp1,"\nhow are you?");
   fclose(fp1);
   fp1 = fopen("first.txt","r");
   while(fgets(text,sizeof(text),fp1))
   {
      printf ("%s",text);
   }
   fclose(fp1);
   printf ("\n operation sucessfull");

   return 0;
}
```