Experiment No. 3

Familiarization with Decision Control Structures & Loop Control Structures.

1. **Objective:** This experiment is designed to demonstrate the use of control statements and loop structures and its effectiveness in C programming

2. **Software requirements:**
    - Any IDE that supports a C compiler (preferably Code::Blocks; any other IDE or text editor with a C compiler installed will also be accepted.)

3. **Theoretical Background and Example Problems**

    Decision Control Structure:
    While writing programs, we often want a set of instructions to be executed in one situation, and an entirely different set of instructions to be executed in another situation.
    This kind of situation is dealt with in C programs using a decision control instruction. A decision control instruction can be implemented in C using:
    (a) The if statement
    (b) The if-else statement
    (c) The conditional operators

    Example 1: Write a program to calculate the salary as per the following table:

Gender	Years of Service	Qualification	Salary
Male	>= 10	Post - Graduate	15000
	>= 10	Graduate	10000
	<10	Post - Graduate	10000
	<10	Graduate	7000
Female	>= 10	Post - Graduate	12000
	>= 10	Graduate	9000
	<10	Post - Graduate	10000
	<10	Graduate	6000

    Ideation: First of all, we have 2 different salary structures based on the gender. So an ***if else if*** block will be used to program salary structures for male and female employees. Then we have to insert one ***if-else*** block in each of the blocks for male and female employees which will be used for employees

with different ranges of 'years of service'. Finally another ***if-else*** block will be inserted in each of the previous if and else blocks to deal with different level of qualifications. Let us take a look at the code.

Code:

```
#include<stdio.h>
#include<stdlib.h>


int main()

{  char gen;
   int yos, sal, qual;
   printf("Enter the gender of the employee. M for male and F for
female:
   ");

   scanf(" %c",&gen);
   printf("\nEnter the years of service of the employee: ");
   scanf("%d", &yos);
   printf("\nEnter the qualification of the employee. 0 for graduate
   and 1
   for post-graduate: ");

   scanf("%d", &qual);
   printf("\n\n");


   if (gen == 'M')

   {
      if (yos >= 10)
      {
           if (qual == 1)
           {
                 sal = 15000;
           }
           else if (qual == 0)
           {
                 sal = 10000;
           }
      }

     else
     {
           if (qual == 1)
           {
                sal = 10000;
```

```
            }
            else if (qual == 0)
            {
                 sal = 7000;
            }
      }
   }


   else if (gen == 'F')
   {
      if (yos >= 10)
      {
            if (qual == 1)
            {
                 sal = 12000;
            }
            else if (qual == 0)
            {
                 sal = 9000;
            }
      }
      else
      {
            if (qual == 1)
            {
                sal = 10000;
            }
            else if (qual == 0)
            {
               sal = 6000;
            }
      }
   }
   printf("Salary is %d\n\n", sal);
   return 0;


}
```

Example 2: If the three sides of a triangle are entered through the keyboard, write a program to check (a) Whether the triangle is valid or not. (The triangle is valid if the sum of two sides is greater than the largest of the three sides.)
(b) Check whether the triangle is isosceles, equilateral, scalene or right angled triangle.

Ideation: We have to scan the length of 3 sides from the user and at first check for the validity of the triangle. An ***if-else*** block will be used for this purpose. If the lengths given form a valid triangle, we can check for whether at least two of the sides are equal using another ***if-else*** block. If at least two sides are equal we can check for isosceles or equilateral using another ***if-else*** block. Otherwise it is scalene triangle. For checking whether the triangle is right-angled, we have to check whether the sides agree with Pythagorean Theorem in any of the three possible combinations using an ***if*** block.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
       int a, b, c; //Three sides of the triangle
       printf("Enter the length of the 1st side: ");
       scanf(" %d", &a);
       printf("\nEnter the length of the 2nd side: ");
       scanf(" %d", &b);
       printf("\nEnter the length of the 3rd side: ");
       scanf(" %d", &c);


       printf("\n\n");


       if ((a+b > c) && (b+c > a) && (c+a > b))
       {
           if ((a == b) || (b == c) || (c == a))
           {
                if ((a == b) && (b == c))
                {
                     printf("Equilateral Triangle\n");
                }
               else
                {
                     printf("Isosceles Triangle\n");
                }
           }
          else
           {
                printf("Scalene Triangle\n");
           }
           int a_sq = pow(a, 2);
           int b_sq = pow(b, 2);
           int c_sq = pow(c, 2);
```

```
        if ((a_sq == b_sq + c_sq) || (b_sq == c_sq + a_sq) ||
(c_sq == a_sq +b_sq))
        {
             printf("Right-angled Triangle\n");
        }
     }
     else
    {
           printf("Not Valid\n");
    }


}
```

Loop Control Structure:

The versatility of the computer lies in its ability to perform a set of instructions repeatedly. This involves repeating some portion of the program either a specified number of times or until a particular condition is being satisfied. This repetitive operation is done through a loop control instruction. There are three methods by way of which we can repeat a part of a program. They are:

(a) Using a for statement

(b) Using a while statement

(c) Using a do-while statement

Example 3: Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55… Starting from the third term, any term in Fibonacci series is the summation of previous two terms. Write a program using while loop to display first n terms of Fibonacci series and a sum of these n terms. n is a number anywhere between 3 to 45 to be given as input by the user. You have to use ***While*** loop to solve the problem. **Ideation:** We have to set the values of two variables indicating two previous terms to 0 & 1. We have to constantly update these variables as we move forward to higher terms.

Each term will be the sum of previous two terms (i.e. the values stored in the above mentioned variables). Another variable for sum of the terms will have to be updated after each term by adding the new term to the previously calculated sum.

Code:

```
#include <stdio.h>
int main()
{
    int t1 = 0, t2 = 1;
    int sum = 1;
    int n;
    /*Ask the user to enter any number between 3 and 45
    It the number is out of this range, display an error message */
```

```
    printf("Enter the number of terms to be displayed: ");
    scanf("%d", &n);


    if (n<3 || n>45)
    {
         printf("\n\nInvalid!!\n\n");
    }
    else
    {
         printf("\nThe Fibonacci series upto %dth term:\n", n);
         printf("%d, %d", 0, 1);
         int i = 3;
         int t;
         while (i <= n)
         {
                 t = t1+t2;
                 t1 = t2;
                 t2 = t;
                 sum = sum + t;
                 printf(", %d", t);
                 i++;
         }
         printf("\n\nThe sum of the first %d terms of Fibonacci series
=
%d\n\n",n, sum);
    }
}
```

Example 4: Write a program to produce the following output:

```
A   B   C   D   E   F   G   F   E   D   C   B   A
A   B   C   D   E   F       F   E   D   C   B   A
A   B   C   D   E               E   D   C   B   A
A   B   C   D                       D   C   B   A
A   B   C                           C   B   A
A   B                               B   A
A                                       A
```

Ideation: There are many ways to generate this output. Students are encouraged to explore different ways of generating patterns. Try making a diamond using * symbols for example.

Code:

```
#include<stdio.h>
int main()
{
     for (int i = 1; i<=7; i++)
     {
        if (i == 1)
        {
               char ch = 'A';
               int k;
               for (int j = -6; j<=6; j++)
               {
                  if (j<0)
                  {
                      k = -j;
                  }
                  else
                  {
                      k = j;
                  }
                  printf("%c ", ch + (6-k));
               }
         }
       else
       {
                int num_space = 2*i - 3;
                int x = num_space/2;
                int k;
                char ch = 'A';
                for (int j = -6; j<=6; j++)
                {
                   if (j<-x || j>x)
                   {
                      if (j<0)
                      {
                         k = -j;
                      }
                      else
                      {
                          k = j;
                      }
```

```
                    printf("%c ", ch + (6-k));
                }
                else
                {
                    printf(" ");
                }
            }
        }
        printf("\n\n");
    }
}
```

4. **Results after running the example programs:**
   Screenshots of the results:

5. **Lab report assignments (submit source code and screenshots of results):**

   a. Any year is entered through the keyboard, write a program to determine whether the year is leap or not.
   b. A certain grade of steel is graded according to the following. conditions:
      (i) Hardness must be greater than 50
      (ii) Carbon content must be less than 0.7
      (iii) Tensile strength must be greater than 5600
      The grades are as follows:
      Grade is 10 if all three conditions are met
      Grade is 9 if conditions (i) and (ii) are met
      Grade is 8 if conditions (ii) and (iii) are met
      Grade is 7 if conditions (i) and (iii) are met
      Grade is 6 if only one condition is met
      Grade is 5 if none of the conditions are met
      Write a program, which will require the user to give values of hardness, carbon content and tensile strength of the steel under consideration and output the grade of the steel.
   c. Write a program for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:

      There are 21 matchsticks.

      The computer asks the player to pick 1, 2, 3, or 4 matchsticks.

      After the person picks, the computer does its picking.

      Whoever is forced to pick up the last matchstick loses the game.

d. Write a program to produce the following output:

```
1
2 3
4 5 6
7 8 9 10
```

e. Population of a town today is 100000. The population has increased steadily at the rate of 10 % per year for last 10 years. Write a program to determine the population at the end of each year in the last decade.

6. Comment/Discussion on the obtained results and discrepancies (if any).