Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure Lab

Lab-1

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Q1. Ramesh's basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary.

```c
#include <stdio.h>
int main()
{
    // Variable declarations
    int basic_salary;   // Stores the basic salary as an integer
    float gross_salary, DA, HRA;   // Stores gross salary, Dearness Allowance (DA), and House Rent Allowance (HRA) as floating-point numbers

    // Prompt the user to enter the basic salary
    printf("Enter the basic salary: ");
    scanf("%d", &basic_salary);   // Read the input from the user and store it in 'basic_salary' using the address-of (&) operator

    DA = basic_salary * 40 / 100;   // DA is 40% of the basic salary
    HRA = basic_salary * 20 / 100;  // HRA is 20% of the basic salary

    // Calculate the gross salary by adding the basic salary, DA, and HRA
    gross_salary = basic_salary + DA + HRA;

    // Print the calculated values to the console
    printf("\n Basic Salary: %.2f", basic_salary);    // %.2f formats the float with two decimal places
    printf("\n Dearness Allowance: %.2f", DA);
    printf("\n House Rent Allowance: %.2f", HRA);
    printf("\n Gross Salary: %.2f", gross_salary);

    return 0;   // Indicate successful program execution by returning 0
}
```

```
PS D:\MCA\MCA-DSA> cd .\LAB-1\
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Enter the basic salary :100000

 Basic Salary: -1.#R
 Dearness Allowance: 40000.00
 House Rent Allowence: 20000.00
 Gross Salary: 160000.00
PS D:\MCA\MCA-DSA\LAB-1>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q2. The distance between two cities (in km.) is input through the keyboard. Write a program to convert and print this distance in meters, feet, inches and centimeters.

```c
#include <stdio.h>
int main()
{
    // Variable declarations to store the distances in various units
    float distance, miles, inches, centimeter, meter;

    // Prompt the user to enter the distance in kilometers
    printf("Enter the distance in km: ");
    scanf("%f", &distance); // Read the input from the user and store it
in 'distance'

    // Convert the distance to miles, inches, centimeters, and meters
    miles = distance * 0.62137119; //Conversion factor for km to miles
    inches = distance * 39370.1;  //Conversion factor for km to inches
    centimeter = distance * 100000; // Conversion factor for km to cm
    meter = distance * 1000;        // Conversion factor for km to m

    // Print the converted distances to the console
    printf("Distance in centimeters: %f\n", centimeter);
    printf("Distance in inches: %f\n", inches);
    printf("Distance in miles: %f\n", miles);
    printf("Distance in meters: %f", meter);

    return 0; // Indicate successful program execution by returning 0
}
```

```
PS D:\MCA\MCA-DSA> cd .\LAB-1\
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Enter the distance in km :67
Distance in centimeters 6700000.000000
Distance in inches 2637796.750000
Distance in miles 41.631870
Distance in meter 67000.000000
PS D:\MCA\MCA-DSA\LAB-1>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q3. If the marks obtained by a student in five different subjects are input through the keyboard, find out the aggregate marks and percentage marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.

```c
#include <stdio.h>
void main()
{
    // Variable declarations to store marks and calculations
    int subject1, subject2, subject3, subject4, subject5, total;
    float percentage;
    // Prompt the user to enter the marks for each subject
    printf("Enter the marks of Subject1: ");
    scanf("%d", &subject1);
    printf("Enter the marks of Subject2: ");
    scanf("%d", &subject2);
    printf("Enter the marks of Subject3: ");
    scanf("%d", &subject3);
    printf("Enter the marks of Subject4: ");
    scanf("%d", &subject4);
    printf("Enter the marks of Subject5: ");
    scanf("%d", &subject5);
    // Calculate the total marks
    total = subject1 + subject2 + subject3 + subject4 + subject5;
    // Calculate the percentage by dividing the total marks by the number
of subjects (5 in this case)
    percentage = total / 5;

    // Print the aggregate marks and percentage marks to the console
    printf("\nAggregate marks: %d", total);
    printf("\nPercentage marks: %0.2f %%", percentage);
}
```

```
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Enter the marks of Subject1: 98
Enter the marks of Subject2: 87
Enter the marks of Subject3: 68
Enter the marks of Subject4: 78
Enter the marks of Subject5: 91

Aggregate marks: 422
Percentage marks: 84.00 %
PS D:\MCA\MCA-DSA\LAB-1>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q4. Temperature of a city in Fahrenheit degrees is input through the keyboard. Write a program to convert this temperature into Centigrade degrees.

```c
#include <stdio.h>

int main()
{
    // Variable declarations to store temperature values
    float fahrenheit, centigrade;

    // Prompt the user to enter the temperature in Fahrenheit
    printf("Enter Temperature in Fahrenheit: ");
    scanf("%f", &fahrenheit); // Read the input from the user and store
it in 'fahrenheit' using the address-of (&) operator

    // Convert Fahrenheit to Centigrade (Celsius)
    centigrade = (fahrenheit - 32) * 5 / 9;

    // Print the temperature in Centigrade (Celsius) to the console
    printf("The Temperature in Centigrade Degree: %.2f", centigrade);

    return 0; // Indicate successful program execution by returning 0
}
```

```
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question4.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Enter Temperature in Fahrenheit: 92
The Temperature in Centigrade Degree: 33.33
PS D:\MCA\MCA-DSA\LAB-1>
```

Q5. The length & breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area & perimeter of the rectangle, and the area & circumference of the circle.

```c
#include <stdio.h>
int main()
{
    // Variable declarations to store the measurements and calculations
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    float length, bredth, radius, areaOfCircle, circumference_Circle,
areaOfRect, perimeterOfRectangle;

    // For rectangle
    printf("Enter the length of rectangle: ");
    scanf("%f", &length);//Read the length of the rectangle from the user

    printf("Enter the breadth of rectangle: ");
    scanf("%f", &bredth);//Read the breadth of the rect from the user
    // For circle
    printf("Enter the radius of circle: ");
    scanf("%f", &radius); // Read the radius of the circle from the user

    // Calculate area & perimeter of the rectangle...
    areaOfRect = length * bredth;// Area of Rectangle = Length x Breadth
    perimeterOfRectangle = 2 * (length + bredth); // Perimeter of
Rectangle = 2 * (Length + Breadth)

    // Calculate area & circumference of the circle...
    areaOfCircle = 3.14 * radius * radius;  // Area of Circle = Pi * r^2
where Pi = 3.14
    circumference_Circle = 2 * 3.14 * radius; // Circumference of Circle
= 2 * Pi * r

    // Print the calculated values to the console
    printf("\nThe area of the rectangle: %0.2f", areaOfRect);
    printf("\nThe perimeter of the rectangle: %.2f",
perimeterOfRectangle);
    printf("\n\nThe area of the circle: %.2f", areaOfCircle);
    printf("\nThe circumference of the circle: %.2f",
circumference_Circle);

    return 0; // Indicate successful program execution by returning 0
}
```

```
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question5.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Enter the length of rectangle: 10
Enter the breadth of rectangle: 20
Enter the radius of circle: 5

The area of the rectangle: 200.00
The perimeter of the rectangle: 60.00

The area of the circle: 78.50
The circumference of the circle: 31.40
PS D:\MCA\MCA-DSA\LAB-1>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q6. The Paper of size A0 has dimensions 1189 mm * 841mm.Each subsequent size A(n) is defined as A(n-1) cut in half parallel to its shorter sides. Thus paper of size A1would have dimensions 841mm * 594mm. Write a program to calculate and print paper sizes A0,A1,A2….A8.

```c
#include <stdio.h>

int main()
{
    // Variable declarations to store the length and breadth
    int length, bredth;
    // Variables to store intermediate calculations for length and
breadth
    float l1, l2, l3, l4, l5, l6, l7, l8;
    float b1, b2, b3, b4, b5, b6, b7, b8;

    // Prompt the user to enter the length and breadth
    printf("Please enter the length and bredth: ");
    scanf("%d %d", &length, &bredth); // Read the input from the user and
store it in 'length' and 'bredth'

    // Calculate the intermediate lengths (l1 to l8) and breadths (b1 to
b8) using halving method
    l1 = length / 2;
    l2 = l1 / 2;
    l3 = l2 / 2;
    l4 = l3 / 2;
    l5 = l4 / 2;
    l6 = l5 / 2;
    l7 = l6 / 2;
    l8 = l7 / 2;
    b1 = bredth / 2;
    b2 = b1 / 2;
    b3 = b2 / 2;
    b4 = b3 / 2;
    b5 = b4 / 2;
    b6 = b5 / 2;
    b7 = b6 / 2;
    b8 = b7 / 2;

    // Print the calculated values (A0 to A8) to the console
    printf("A0 - %f %f\n", l1, b1);
    printf("A1 - %f %f\n", l1, b1);
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    printf("A2 - %f %f\n", l2, b2);
    printf("A3 - %f %f\n", l3, b3);
    printf("A4 - %f %f\n", l4, b4);
    printf("A5 - %f %f\n", l5, b5);
    printf("A6 - %f %f\n", l6, b6);
    printf("A7 - %f %f\n", l7, b7);
    printf("A8 - %f %f\n", l8, b8);

    return 0; // Indicate successful program execution by returning 0
}
```

```
PS D:\MCA\MCA-DSA\LAB-1> gcc .\Question6.c
PS D:\MCA\MCA-DSA\LAB-1> .\a.exe
Please enter the length and bredth :1189 841
A0 -594.000000 420.000000
A1 -594.000000 420.000000
A2 -297.000000 210.000000
A3 -148.500000 105.000000
A4 -74.250000 52.500000
A5 -37.125000 26.250000
A6 -18.562500 13.125000
A7 -9.281250 6.562500
A8-4.640625 3.281250
PS D:\MCA\MCA-DSA\LAB-1>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure Lab

Lab-2

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Q1. WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or scalene. Take sides of the triangle as input from a user.

```c
#include <stdio.h>

void main()
{
    int side1, side2, side3;

    // User input for the sides of the triangle
    printf("Enter the side1 of the triangle: ");
    scanf("%d", &side1);
    printf("Enter the side2 of the triangle: ");
    scanf("%d", &side2);
    printf("Enter the side3 of the triangle: ");
    scanf("%d", &side3);

    // Checking for different types of triangles
    if (side1 == side2 && side2 == side3)
    {
        printf("Equilateral Triangle"); // All sides are equal
    }
    else if (side1 == side2 || side2 == side3 || side3 == side1)
    {
        printf("Isosceles Triangle"); // Two sides are equal
    }
    else if (side1 * side1 + side2 * side2 == side3 * side3 || side2 *
side2 + side3 * side3 == side1 * side1 || side3 * side3 + side1 * side1
== side2 * side2)
    {
        printf("Right Angle Triangle"); // Satisfies Pythagorean theorem
    }
    else if (side1 != side2 && side2 != side3 && side3 != side1)
    {
        printf("Scalene Triangle"); // All sides are different
    }
    else
    {
        printf("Invalid Triangle"); // Invalid input or sides cannot form
a triangle
    }
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-2> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the side1 of the triangle: 10
Enter the side2 of the triangle: 10
Enter the side3 of the triangle: 20
Isosceles Triangle
PS D:\MCA\MCA-DSA\LAB-2> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the side1 of the triangle: 30
Enter the side2 of the triangle: 30
Enter the side3 of the triangle: 30
Equilateral Triangle
PS D:\MCA\MCA-DSA\LAB-2>
```

Q2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI= weight(kgs)/Height(Mts)*Height(Mts).

	BMI
Starvation	<15
Anorexic	15.1 to 17.5
Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 25.9
Obese	30 to 39.9
Morbidity Obese	40.0 above

```c
#include <stdio.h>

void main()
{
    float height, weight, bmi;

    // User input for weight and height
    printf("Enter your weight in (kilograms): ");
    scanf("%f", &weight);
    printf("Enter your height in (meters): ");
    scanf("%f", &height);

    // Calculate BMI
    bmi = weight / (height * height);
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    // Display BMI value
    printf("Your BMI is: %.2f\n", bmi);

    // Categorize BMI into different groups
    if (bmi < 15)
        printf("Starvation"); // BMI less than 15
    else if (bmi >= 15.1 && bmi <= 17.5)
        printf("Anorexic"); // BMI between 15.1 and 17.5
    else if (bmi >= 17.6 && bmi <= 18.5)
        printf("Underweight"); // BMI between 17.6 and 18.5
    else if (bmi >= 18.6 && bmi <= 24.9)
        printf("Ideal"); // BMI between 18.6 and 24.9
    else if (bmi >= 25 && bmi <= 25.9)
        printf("Overweight"); // BMI between 25 and 25.9
    else if (bmi >= 30 && bmi <= 39.9)
        printf("Obese"); // BMI between 30 and 39.9
    else
        printf("Morbidity Obese"); // BMI greater than or equal to 40
}
```

```
PS D:\MCA\MCA-DSA\LAB-2> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter your weight in kilograms: 81
Enter your height in meters: 1.76
Your BMI is: 26.15
Morbidity Obese
```

Q3. WAP to check if three points (x1,y1), (x2,y2) and (x3,y3) are collinear or not.

```
#include <stdio.h>

int main()
{
    int x1, x2, x3, y1, y2, y3;

    // User input for the coordinates of three points
    printf("Enter the value of x1 and y1: ");
    scanf("%d %d", &x1, &y1);
    printf("Enter the value of x2 and y2: ");
    scanf("%d %d", &x2, &y2);
    printf("Enter the value of x3 and y3: ");
    scanf("%d %d", &x3, &y3);
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    // Calculate the value to check if the points are collinear
    int value = x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2);

    // Check if the value is zero to determine collinearity
    if (value == 0)
    {
        printf("Entered points are collinear"); // Points are collinear
if the value is zero
    }
    else
    {
        printf("Entered points are not collinear"); // Points are not
collinear if the value is not zero
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-2> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the value of x1 and y1: 1 2
Enter the value of x2 and y2: 1 4
Enter the value of x3 and y3: 1 6
Entered points are collinear
PS D:\MCA\MCA-DSA\LAB-2>
```

Q4. According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is input through the keyboard write a program to find out what is the day on 1st January of this year.

```
#include <stdio.h>
void main()
{
    int year, daysInYear, daysOfName;
    printf("Enter the year :");
    scanf("%d", &year);
    daysInYear = year * 365;
    daysOfName = daysInYear % 7;
    if (daysOfName == 1)

        printf("Monday");

    else if (daysOfName == 2)
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        printf("Tuesday");
    else if (daysOfName == 3)
        printf("Wednesday");
    else if (daysOfName == 4)
        printf("Thrusday");
    else if (daysOfName == 5)
        printf("Friday");
    else if (daysOfName == 6)
        printf("Saturday");
    else if (daysOfName == 0)
        printf("Sunday");
    else
        printf("Invalid day");
}
```

```
PS D:\MCA\MCA-DSA\LAB-2> gcc .\Question4.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the year :2022
Saturday
PS D:\MCA\MCA-DSA\LAB-2> GCC .\Question4.c
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the year :2023
Sunday
PS D:\MCA\MCA-DSA\LAB-2>
```

Q5. WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three.

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
#include <stdio.h>

int main()
{
    // Declare variables to store dimensions and perimeters
    int len1, len2, len3, wid1, wid2, wid3;
    int per1, per2, per3;

    // Input dimensions of Rectangle 1
    printf("Enter the length and breadth of Rectangle 1: ");
    scanf("%d %d", &len1, &wid1);

    // Input dimensions of Rectangle 2
    printf("Enter the length and breadth of Rectangle 2: ");
    scanf("%d %d", &len2, &wid2);

    // Input dimensions of Rectangle 3
    printf("Enter the length and breadth of Rectangle 3: ");
    scanf("%d %d", &len3, &wid3);

    // Calculate perimeters for each rectangle
    per1 = 2 * (len1 + wid1);
    per2 = 2 * (len2 + wid2);
    per3 = 2 * (len3 + wid3);

    // Compare perimeters using ternary operators and print results
    (per1 > per2 && per1 > per3) ? printf("Rectangle 1 has the greater
perimeter\n") : (per2 > per3) ? printf("Rectangle 2 has the greater
perimeter\n")

                            : printf("Rectangle 3 has the greater
perimeter\n");

    return 0; // Return 0 to indicate successful program execution
}
```

```
PS D:\MCA\MCA-DSA\LAB-2> .\a.exe
Enter the length and breadth of Rectangle 1: 10 20
Enter the length and breadth of Rectangle 2: 30 40
Enter the length and breadth of Rectangle 3: 10 30
Rectangle 2 has the greater perimeter
PS D:\MCA\MCA-DSA\LAB-2>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure Lab

Lab-3

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Q1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative, and Zeroes entered.

```c
#include <stdio.h>

int main()
{
    int positive_count = 0;
    int negative_count = 0;
    int zero_count = 0;
    float num;
    char user_choice = 'y';

    while (user_choice == 'y' || user_choice == 'Y')
    {
        printf("Enter a number: ");
        scanf("%f", &num);

        if (num > 0)
        {
            positive_count++;
        }
        else if (num < 0)
        {
            negative_count++;
        }
        else
        {
            zero_count++;
        }

        printf("Do you want to enter another number? (y/n): ");
        scanf(" %c", &user_choice);
    }

    printf("Count of positive numbers: %d\n", positive_count);
    printf("Count of negative numbers: %d\n", negative_count);
    printf("Count of zeros: %d\n", zero_count);

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA> cd .\LAB-3\
PS D:\MCA\MCA-DSA\LAB-3> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe
Enter a number: 1
Do you want to enter another number? (y/n): y
Enter a number: -2
Do you want to enter another number? (y/n): y
Enter a number: 0
Do you want to enter another number? (y/n): n
Count of positive numbers: 1
Count of negative numbers: 1
Count of zeros: 1
PS D:\MCA\MCA-DSA\LAB-3>
```

Q2. WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting.

```c
#include <stdio.h>
void main()
{
    //* By using For loop

    // int number;
    // printf("Enter the number :");
    // scanf("%d", &number);
    // for (int i = 1; i <= 10; i++)
    // {
    //     printf("%d x %d = %d\n", number, i, number * i);
    // }

    //*By using while loop

    // int number;
    // printf("Enter the number :");
    // scanf("%d", &number);
    // int i = 1;
    // while (i <= 10)
    // {
    //     printf("%d x %d = %d\n", number, i, number * i);
    //     i++;
    // }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    //*By using do while loop

    int number;
    printf("Enter the number :");
    scanf("%d", &number);
    int i = 1;
    do
    {
        printf("%d x %d = %d\n", number, i, number * i);
        i++;
    } while (i <= 10);
}
```

```
PS D:\MCA\MCA-DSA\LAB-3> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe
Enter the number :5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
PS D:\MCA\MCA-DSA\LAB-3>
```

Q3. WAP to generate the following set of output.

```c
      #include <stdio.h>
void main()
{
    // Pattern 1
    int count = 1;
    for (int i = 0; i <= 3; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            printf("%d", count);
            count++;
        }

        printf("\n");
    }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    // Pattern 2
    int num;
    printf("Enter the number of rows :");
    scanf("%d", &num);
    for (int i = 0; i < num; i++)
    {
        int value = 1;
        for (int j = 0; j <= i; j++)
        {
            printf("%d", value);
            value = value * (i - j) / (j + 1);
        }
        printf("\n");
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-3> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe

1
23
456
Enter the number of rows :5
1
11
121
1331
14641
PS D:\MCA\MCA-DSA\LAB-3>
```

Q4. The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.

```c
#include <stdio.h>
void main()
{
    int initial_population = 100000;
    double growth_rate = 0.10;
    int years = 10;
    int population;
    for (int i = 0; i < years; i++)
    {
        population = 100000 + (int)(initial_population * growth_rate);
        printf("%d \t %d\n", i, population);
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        initial_population = population;
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-3> gcc .\Question4.c
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe
0       110000
1       111000
2       111100
3       111110
4       111111
5       111111
6       111111
7       111111
8       111111
9       111111
```

Q5. Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to a reasonable limit.

```
#include <stdio.h>

int main()
{
    int i, num, x, y, count;
    printf("Enter the range in which you find the number:");
    scanf("%d", &num);
    for (i = 1; i <= num; i++)
    {
        count = 0;
        for (x = 1; x * x * x < i; x++)
        {
            for (y = x; x * x * x + y * y * y <= i; y++)
            {
                if (x * x * x + y * y * y == i)
                {
                    count++;
                }
            }
        }
        if (count == 2)
        {
            printf("%d\n", i);
        }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-3> gcc .\Question5.c
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe
Enter the range in which you find the number:2000
1729
PS D:\MCA\MCA-DSA\LAB-3> .\a.exe
Enter the range in which you find the number:5000
1729
4104
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure Lab

Lab-4

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Q1. WAP for printing Fibonacci sequence. Take input from the user to print up to a certain limit.

```c
#include <stdio.h>

// Function to calculate the Fibonacci sequence
int fibonacci(int num)
{
    if (num == 0 || num == 1) // Base cases: Fibonacci of 0 and 1 is 1
    {
        return 1;
    }
    else
    {
        // Recursive step: Fibonacci of num is the sum of the previous
two Fibonacci numbers
        return fibonacci(num - 1) + fibonacci(num - 2);
    }
}
// Main function
void main()
{
    int num;
    printf("Enter the number: "); // Prompt user to input a number
    scanf("%d", &num);            // Read the input number

    printf("Fibonacci sequence up to %d:\n", num);
    for (int i = 0; i <= num; i++)
    {
        printf("%d, ", fibonacci(i)); // Print Fibonacci sequence up to
the given number
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number :10
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number :5
1, 1, 2, 3, 5, 8,
PS D:\MCA\MCA-DSA\LAB-4>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q2. WAP to swap two variables without using a third variable, depict the same using call by value concept.

```c
#include <stdio.h>

// Function to swap the values of two numbers using pointers
void swapTwoNumbers(int *num1, int *num2)
{
    // Swapping using arithmetic operations
    *num1 = *num1 + *num2; // Add num1 and num2 and store in num1
    *num2 = *num1 - *num2; // Subtract num2 from the new num1 and store
in num2
    *num1 = *num1 - *num2; // Subtract the new num2 from the new num1 and
store in num1
}

int main()
{
    int num1, num2;
    printf("Enter the value of num1: ");
    scanf("%d", &num1); // Read the value of num1 from the user
    printf("Enter the value of num2: ");
    scanf("%d", &num2); // Read the value of num2 from the user

    // Without using functions
    // printf("The value before swapping is %d, %d\n", num1, num2);
    // num1 = num1 + num2;
    // num2 = num1 - num2;
    // num1 = num1 - num2;
    // printf("The value after swapping is %d, %d", num1, num2);

    // Calling function to swap two numbers using pointers
    printf("The value before swapping is %d, %d\n", num1, num2);
    swapTwoNumbers(&num1, &num2); // Call the swap function
    printf("The value after swapping  is %d, %d", num1, num2);

    return 0; // Indicate successful execution of the program
}
```

```
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the value of num1 :10
Enter the value of num2 :30
The value before swaping is 10 , 30
The value after swaping  is 30 , 10
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q3. A positive integer is entered through the keyboard. Write a Function to print the prime factors of this number.

For example, 24 have prime factors: 2,2,2, and 3, whereas 35 have prime factors 5 and 7.

```c
#include <stdio.h>

// Function to find and print prime factors of a number
void primeFactors(int num)
{
    for (int count = 2; num > 1; count++)
    {
        while (num % count == 0)
        {
            printf("%2d ,", count); // Print the prime factor
            num = num / count;      // Reduce the number by dividing it
by the prime factor
        }
    }
}

int main()
{
    int num;
    printf("Enter the number: ");
    scanf("%d", &num); // Read the input number from the user

    printf("Prime factors of %d are: ", num);
    primeFactors(num); // Call the function to find and print prime
factors

    return 0; // Indicate successful execution of the program
}
```

```
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number: 24
Prime factors of 24 are:  2 * 2 * 2 * 3 *
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number: 24
Prime factors of 24 are:  2 , 2 , 2 , 3 ,
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number: 35
Prime factors of 35 are:  5 , 7 ,
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Q4. WAP which makes use of Switch and functions to implement the conversion of a given number to a given format.

For eg. If a decimal number is made input, it should ask for which number system you want to convert, and the conversion process should do that. Like Decimal to Binary, the program should take 65 as input and **1000001.**

```c
#include <stdio.h>

// Function to convert a decimal number to binary representation
void numberToBinary(int num)
{
    int a[10], i, j;

    // Convert decimal number to binary by repeatedly dividing by 2
    for (i = 0; num > 0; i++)
    {
        a[i] = num % 2; // Store the remainder (binary digit)
        num = num / 2;  // Update num by dividing it by 2
    }

    printf("\nBinary representation of the given number is: ");
    for (j = i - 1; j >= 0; j--)
    {
        printf("%d", a[j]); // Print the binary digits in reverse order
    }
}

// Function to choose the desired conversion format
void convert(int number, int format)
{
    switch (format)
    {
    case 1:
        printf("Binary representation of %d is: ", number);
        numberToBinary(number); // Call function to convert to binary
        break;
    default:
        printf("Invalid format");
    }
}

int main()
{
    int num, format;

    printf("Enter the number: ");
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    scanf("%d", &num); // Read the input number from the user

    printf("Enter the format:\nPress 1 for binary: ");
    scanf("%d", &format); // Read the desired format from the user

    convert(num, format); // Call the function to convert and print the
result

    return 0; // Indicate successful execution of the program
}
```

```
PS D:\MCA\MCA-DSA\LAB-4> gcc .\Question4.c
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number: 65
Enter the format:
Press 1 for binary: 1
Binary representation of 65 is:
Binary representation of the given number is: 1000001
PS D:\MCA\MCA-DSA\LAB-4> .\a.exe
Enter the number: 34
Enter the format:
Press 1 for binary: 2
Invalid format
PS D:\MCA\MCA-DSA\LAB-4>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-5

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

A. WAP to implement the following scenarios. Take all the input from user, nothing should be imagined or hard coded.

1. Transpose of a matrix

```c
#include <stdio.h>

int main()
{
    int arr[2][3], i, j;

    // Input values for the matrix
    printf("Enter values for a 2x3 matrix:\n");
    for (i = 0; i < 2; i++)
   {
       for (j = 0; j < 3; j++)
        {
         scanf("%d", &arr[i][j]);
        }
    }

   // Display the original matrix
    printf("The matrix is:\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", arr[i][j]);
        }
        printf("\n");
    }
    // Display the transpose of the matrix
    printf("The transpose of the matrix is:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 2; j++)
        {
            printf("%d\t", arr[j][i]);
        }
        printf("\n");
    }

return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionA1.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Enter values for a 2x3 matrix:
1 2 3
3 4 6
The matrix is:
1       2       3
3       4       6
The transpose of the matrix is:
1       3
2       4
3       6
PS D:\MCA\MCA-DSA\LAB-5>
```

2.Check if a matrix is Syymetrical or not

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

void main()
{
    int a[3][3], i, j;
    bool isSymmetric = true;
    // Input values for the matrix
    printf("Enter the elements of the matrix:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    // Display the elements of the matrix
    printf("The elements of the matrix are:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }

    // Display the transpose of the matrix
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    printf("Transpose of the matrix is:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", a[j][i]);
        }
        printf("\n");
    }
    // Check if the matrix is symmetric
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            if (a[i][j] != a[j][i])
            {
                isSymmetric = false;
                break; // Exit the loop as soon as a non-symmetric
element is found
            }
        }
    }
    // Check and display if the matrix is symmetric or not
    if (isSymmetric)
    {
        printf("Matrix is symmetric.\n");
    }
    else
    {
        printf("Matrix isn't symmetric.\n");
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionA2.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Enter the elements of the matrix:
1 2 3
2 4 5
3 5 8
The elements of the matrix are:
1       2       3
2       4       5
3       5       8
Transpose of the matrix is:
1       2       3
2       4       5
3       5       8
Matrix is symmetric.
PS D:\MCA\MCA-DSA\LAB-5>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

3.Check the inverse of a matrix

```c
#include <stdio.h>

// Function to print a 3x3 matrix
void printMatrix(double A[3][3])
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%.2f\t", A[i][j]);
        }
        printf("\n");
    }
}

// Global variable to store the determinant
double det = 0;

// Function to calculate the inverse of a 3x3 matrix
void inverseMatrix(double A[3][3], double A_inv[3][3])
{
    // Calculate the determinant of A using the formula for a 3x3 matrix
    for (int i = 0; i < 3; i++)
    {
        det += (A[0][i] * (A[1][(i + 1) % 3] * A[2][(i + 2) % 3] -
A[1][(i + 2) % 3] * A[2][(i + 1) % 3]));
    }

    // Check if the matrix is singular (determinant is zero)
    if (det == 0)
    {
        printf("Matrix is singular. Inverse does not exist.\n");
        return;
    }

    // Calculate the inverse of A using the adjugate and determinant
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            A_inv[i][j] = ((A[(j + 1) % 3][(i + 1) % 3] * A[(j + 2) %
3][(i + 2) % 3]) -
                           (A[(j + 1) % 3][(i + 2) % 3] * A[(j + 2) %
3][(i + 1) % 3])) /
                          det;
        }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    }
}

int main()
{
    double A[3][3] = {{2.0, 3.0, 4.0},
                      {1.0, 5.0, 6.0},
                      {7.0, 8.0, 9.0}};

    double A_inv[3][3];

    printf("Original Matrix A:\n");
    printMatrix(A);

    inverseMatrix(A, A_inv);

    // Check if the determinant is nonzero (inverse exists)
    if (det != 0)
    {
        printf("\nInverse Matrix A_inv:\n");
        printMatrix(A_inv);
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionA3.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Original Matrix A:
2.00    3.00    4.00
1.00    5.00    6.00
7.00    8.00    9.00

Inverse Matrix A_inv:
0.20    -0.33   0.13
-2.20   0.67    0.53
1.80    -0.33   -0.47
PS D:\MCA\MCA-DSA\LAB-5>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

B. WAP to merge two arrays and append them in the following order.

1. Add the first array to the end of another one
2. Add Second Array to the end of the first one
3. Merge the arrays and sort them.

```c
#include <stdio.h>

int main()
{
    // Define and initialize the first array (arr1)
    int arr1[] = {5, 2, 9};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);

    // Define and initialize the second array (arr2)
    int arr2[] = {7, 1, 3};
    int size2 = sizeof(arr2) / sizeof(arr2[0]);

    // Calculate the size of the merged array
    int mergedSize = size1 + size2;

    // Create an array to store the merged elements
    int mergedArray[mergedSize];

    // Copy elements from arr1 and arr2 to mergedArray
    for (int i = 0; i < size1; i++)
    {
        mergedArray[i] = arr1[i];
    }
    for (int i = 0; i < size2; i++)
    {
        mergedArray[size1 + i] = arr2[i];
    }

    // Sort the merged array using bubble sort
    for (int i = 0; i < mergedSize - 1; i++)
    {
        for (int j = 0; j < mergedSize - i - 1; j++)
        {
            if (mergedArray[j] > mergedArray[j + 1])
            {
                // Swap elements if they are in the wrong order
                int temp = mergedArray[j];
                mergedArray[j] = mergedArray[j + 1];
                mergedArray[j + 1] = temp;
            }
```

```
        }
    }

    // Print the sorted merged array
    printf("Merged and sorted array:\n");
    for (int i = 0; i < mergedSize; i++)
    {
        printf("%d ", mergedArray[i]);
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionB.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Merged and sorted array:
1 2 3 5 7 9
PS D:\MCA\MCA-DSA\LAB-5>
```

C. WAP using pointers to find the smallest number in an array using pointer.

```
#include <stdio.h>

int main()
{
    int num;

    // Prompt the user to enter the size of the array
    printf("Enter the size of array: ");
    scanf("%d", &num);

    // Declare an integer array of the given size and a pointer to an
integer
    int arr[num], *small;

    // Input elements into the array
    for (int i = 0; i < num; i++)
    {
        scanf("%d", &arr[i]);
    }

    // Initialize the 'small' pointer to point to the first element of
the array
    small = &arr[0];
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    // Find the smallest element in the array using pointer arithmetic
    for (int i = 0; i < num; i++)
    {
        if (*(arr + i) < *small)
            *small = *(arr + i);
    }

    // Print the smallest element in the array
    printf("Smallest element in the array is %d", *small);

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionC.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Enter the size of array: 5
30 40 10 60 50
Smallest element in the array is 10
PS D:\MCA\MCA-DSA\LAB-5>
```

D. WAP which performs following task.

1. Initialize an integer array of 10 elements in main()
2. Pass the entire array to a function modify()
3. In modify() multiply(you can use division, addition or subtraction) each element of array by 3
4. Return the control to main() and print the new array elements in main().

```
#include <stdio.h>

// Function to modify the array elements by multiplying them by 3
void modify(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        arr[i] *= 3; // Multiply each element by 3
    }
}

int main()
{
    int arr[10]; // Initialize an integer array of 10 elements

    // Initialize the array elements in main()
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    printf("Enter 10 integers:\n");
    for (int i = 0; i < 10; i++)
    {
        scanf("%d", &arr[i]);
    }

    // Call the modify function to multiply each element by 3
    modify(arr, 10);

    // Print the modified array elements in main()
    printf("Modified array elements:\n");
    for (int i = 0; i < 10; i++)
    {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-5> gcc .\QuestionD.c
PS D:\MCA\MCA-DSA\LAB-5> .\a.exe
Enter 10 integers:
1 2 3 4 5 6 7 8 9 10
Modified array elements:
3 6 9 12 15 18 21 24 27 30
PS D:\MCA\MCA-DSA\LAB-5>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-6

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Q1.A record contains the name of a cricketer, his age, the number of test matches he has played, and the average runs he scored in each test match. Create an array of structures to hold records of 20 such cricketers and then write a program to read these records and arrange them in ascending order by runs, Use the qsort standard library function.

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

// Define a structure to represent cricketer data
struct cricketer
{
    char name[50];
    int crick_age;
    int match;
    float avg_match;
};

// Comparison function for qsort to compare cricketers based on average runs
int compare(const void *a, const void *b)
{
    const struct cricketer *cricketerA = (const struct cricketer *)a;
    const struct cricketer *cricketerB = (const struct cricketer *)b;

    return (cricketerA->avg_match > cricketerB->avg_match) - (cricketerA->avg_match < cricketerB->avg_match);
}

int main()
{
    int i, n;

    // Prompt the user to enter the number of cricketers' data
    printf("Enter the number of cricketers' data you want to insert: ");
    scanf("%d", &n);

    // Declare an array of structures to store cricketer data
    struct cricketer obj1[20];

    // Input cricketer data from the user
    for (i = 0; i < n; i++)
    {
        printf("Enter data of cricketer %d\n", i + 1);
        printf("Name: ");
        scanf("%s", obj1[i].name);
        printf("Age: ");
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        scanf("%d", &obj1[i].crick_age);
        printf("Matches: ");
        scanf("%d", &obj1[i].match);
        printf("Average runs: ");
        scanf("%f", &obj1[i].avg_match);
    }

    // Sort the records using qsort and the compare function
    qsort(obj1, n, sizeof(struct cricketer), compare);

    // Display the sorted records
    printf("Sorted records:\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\t%s\t%d\t%d\t%.2f\n", i + 1, obj1[i].name,
obj1[i].crick_age, obj1[i].match, obj1[i].avg_match);
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter the number of cricketers' data you want to insert: 4
Enter data of cricketer 1
Name: Akash
Age: 21
Matches: 2
Average runs: 56
Enter data of cricketer 2
Name: shivam
Age: 22
Matches: 3
Average runs: 36
Enter data of cricketer 3
Name: naman
Age: 23
Matches: 5
Average runs: 142
Enter data of cricketer 4
Name: sorav
Age: 20
Matches: 5
Average runs: 180
Sorted records:
1       shivam  22      3       36.00
2       Akash   21      2       56.00
3       naman   23      5       142.00
4       sorav   20      5       180.00
```

Name:-Aakash Bhatt
Sap-id:- 500124633

2.Create a structure to specify data of customers in a bank. The data to be stored is Account number, Name, and Balance in the account. Assume a maximum of 200 customers in the bank.

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

// Define the structure for customer data
struct Customer
{
    int account_number;
    char name[50];
    double balance;
};

int main()
{
    // Declare an array of Customer structures to store customer data
    struct Customer customers[200];

    int num_customers;

    // Prompt the user to enter the number of customers (up to 200)
    printf("Enter the number of customers (up to 200): ");
    scanf("%d", &num_customers);

    // Input customer data for each customer
    for (int i = 0; i < num_customers; i++)
    {
        printf("Customer #%d:\n", i + 1);

        // Input account number
        printf("Account Number: ");
        scanf("%d", &customers[i].account_number);

        // Input customer name (assuming single-word names)
        printf("Name: ");
        scanf("%s", customers[i].name);

        // Input customer balance
        printf("Balance: ");
        scanf("%lf", &customers[i].balance);
    }

    // Display customer data
    printf("\nCustomer Data:\n");
    for (int i = 0; i < num_customers; i++)
    {
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        printf("Customer #%d:\n", i + 1);
        printf("Account Number: %d\n", customers[i].account_number);
        printf("Name: %s\n", customers[i].name);
        printf("Balance: %.2lf\n", customers[i].balance);
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-6> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter the number of customers (up to 200): 2
Customer #1:
Account Number: 987423
Name: Aakash
Balance: 3000
Customer #2:
Account Number: 348709
Name: shivam
Balance: 1000

Customer Data:
Customer #1:
Account Number: 987423
Name: Aakash
Balance: 3000.00
Customer #2:
Account Number: 348709
Name: shivam
Balance: 1000.00
PS D:\MCA\MCA-DSA\LAB-6>
```

3.Write a function to print the account number and name of each customer with a balance below Rs 100.

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

// Define the structure for customer data
struct Customer
{
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    int account_number;
    char name[50];
    double balance;
};

// Function to print customers with a balance below Rs 100
void printCustomersBelow100(struct Customer customers[], int
num_customers)
{
    printf("Customers with a balance below Rs 100:\n");
    for (int i = 0; i < num_customers; i++)
    {
        if (customers[i].balance < 100.0)
        {
            printf("Account Number: %d\n", customers[i].account_number);
            printf("Name: %s\n", customers[i].name);
            printf("Balance: %.2lf\n", customers[i].balance);
            printf("\n");
        }
    }
}

int main()
{
    struct Customer customers[200];

    int num_customers;

    // Prompt the user to enter the number of customers
    printf("Enter the number of customers: ");
    scanf("%d", &num_customers);

    // Input customer data for each customer
    for (int i = 0; i < num_customers; i++)
    {
        printf("Customer #%d:\n", i + 1);

        // Input account number
        printf("Account Number: ");
        scanf("%d", &customers[i].account_number);

        // Input customer name (assuming single-word names)
        printf("Name: ");
        scanf("%s", customers[i].name);

        // Input customer balance
        printf("Balance: ");
        scanf("%lf", &customers[i].balance);
```

```
    }

    // Call the function to print customers with a balance below Rs 100
    printCustomersBelow100(customers, num_customers);

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-6> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter the number of customers: 3
Customer #1:
Account Number: 234890
Name: Aakash
Balance: 345
Customer #2:
Account Number: 290783
Name: Shivam
Balance: 70
Customer #3:
Account Number: 478132
Name: Naman
Balance: 500
Customers with a balance below Rs 100:
Account Number: 290783
Name: Shivam
Balance: 70.00

PS D:\MCA\MCA-DSA\LAB-6>
```

4.If a customer requests for withdrawal or deposit, the form contains the fields: Acct no, amount, code(1 for deposit and 0 for withdrawal) WAP to give a message " The balance is insufficient for the specified withdrawal", if on withdrawal the balance falls below Rs 100.

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

// Define the structure for customer data
struct Customer
{
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    int account_number;
    char name[50];
    double balance;
};

// Function to perform a deposit or withdrawal transaction
void performTransaction(struct Customer customers[], int num_customers,
int acct_no, double amount, int code)
{
    for (int i = 0; i < num_customers; i++)
    {
        if (customers[i].account_number == acct_no)
        {
            if (code == 1)
            {
                // Deposit
                customers[i].balance += amount;
                printf("Deposit of Rs %.2lf successful.\n", amount);
            }
            else if (code == 0)
            {
                // Withdrawal
                if (customers[i].balance - amount < 100.0)
                {
                    printf("The balance is insufficient for the specified
withdrawal.\n");
                }
                else
                {
                    customers[i].balance -= amount;
                    printf("Withdrawal of Rs %.2lf successful.\n",
amount);
                }
            }
            else
            {
                printf("Invalid transaction code.\n");
            }
            return;
        }
    }

    printf("Account number %d not found.\n", acct_no);
}

int main()
{
    struct Customer customers[200];
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    int num_customers;

    // Prompt the user to enter the number of customers
    printf("Enter the number of customers: ");
    scanf("%d", &num_customers);

    // Input customer data for each customer
    for (int i = 0; i < num_customers; i++)
    {
        printf("Customer #%d:\n", i + 1);

        // Input account number
        printf("Account Number: ");
        scanf("%d", &customers[i].account_number);

        // Input customer name (assuming single-word names)
        printf("Name: ");
        scanf("%s", customers[i].name);

        // Input customer balance
        printf("Balance: ");
        scanf("%lf", &customers[i].balance);
    }

    int acct_no, code;
    double amount;

    // Prompt the user to enter transaction details
    printf("Enter Account Number: ");
    scanf("%d", &acct_no);
    printf("Enter Transaction Code (1 for Deposit, 0 for Withdrawal): ");
    scanf("%d", &code);
    printf("Enter Amount: ");
    scanf("%.2lf", &amount);

    // Call the function to perform the transaction
    performTransaction(customers, num_customers, acct_no, amount, code);

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-6> gcc .\Question4.c
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter the number of customers: 3
Customer #1:
Account Number: 454645
Name: Aakash
Balance: 324
Customer #2:
Account Number: 23809
Name: Shivam
Balance: 90
Customer #3:
Account Number: Naman
Name: Balance: 180
Enter Account Number: 23809
Enter Transaction Code (1 for Deposit, 0 for Withdrawal): 0
Enter Amount: 50
The balance is insufficient for the specified withdrawal.
PS D:\MCA\MCA-DSA\LAB-6>
```

5.WAP to count the number of occurrences of any two vowels in succession in a line of text. For example in the following sentence "Please read this application and give me gratuity". such occurrences, ea, ea and ui.

```c
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

// Function to check if a character is a vowel
bool isVowel(char c)
{
    switch (c)
    {
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':
        return true;
    default:
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
        return false;
    }
}

int main()
{
    char text[1000];

    // Prompt the user to enter a line of text
    printf("Enter a line of text: ");
    fgets(text, sizeof(text), stdin);

    int count = 0;
    int len = strlen(text);

    // Loop through the characters in the input text
    for (int i = 0; i < len - 1; i++)
    {
        // Check if the current character and the next character are both
vowels
        if (isVowel(text[i]) && isVowel(text[i + 1]))
        {
            // If two vowels are found in succession, print them
            printf("Found two vowels in succession: %c%c\n", text[i],
text[i + 1]);
            count++;
        }
    }

    // Print the total number of occurrences of two vowels in succession
    printf("Total number of occurrences of two vowels in succession:
%d\n", count);

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-6> gcc .\Question5.c
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter a line of text: Please read this expression and give me gratuity.
Found two vowels in succession: ea
Found two vowels in succession: ea
Found two vowels in succession: io
Found two vowels in succession: ui
Total number of occurrences of two vowels in succession: 4
PS D:\MCA\MCA-DSA\LAB-6>
```

Name:-Aakash Bhatt
Sap-id:- 500124633

6.WAP to receive an integer and printout the number in words. For example, if the number is 5678, it should print Five thousand six hundred and seventy eight.

```
#include <stdio.h>

// Function to print the word representation of a number (0-9)
void printDigit(int digit)
{
    const char *words[] = {"Zero", "One", "Two", "Three", "Four", "Five",
"Six", "Seven", "Eight", "Nine"};
    printf("%s ", words[digit]);
}

// Function to convert a two-digit number into words
void convertTwoDigits(int num)
{
    if (num < 10)
    {
        printDigit(num);
    }
    else if (num >= 10 && num <= 19)
    {
        const char *teens[] = {"Ten", "Eleven", "Twelve", "Thirteen",
"Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};
        printf("%s ", teens[num - 10]);
    }
    else
    {
        const char *tens[] = {"", "", "Twenty", "Thirty", "Forty",
"Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};
        printf("%s ", tens[num / 10]);
        if (num % 10 > 0)
        {
            printDigit(num % 10);
        }
    }
}

// Function to convert a three-digit number into words
void convertThreeDigits(int num)
{
    if (num >= 100)
    {
        printDigit(num / 100);
        printf("Hundred ");
        num %= 100;
        if (num > 0)
        {
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
            printf("and ");
        }
    }
    convertTwoDigits(num);
}

int main()
{
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);

    if (number < 0)
    {
        printf("Negative ");
        number = -number;
    }

    if (number == 0)
    {
        printf("Zero\n");
    }
    else
    {
        if (number >= 1000)
        {
            convertThreeDigits(number / 1000);
            printf("Thousand ");
            number %= 1000;
        }
        convertThreeDigits(number);
        printf("\n");
    }

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-6> gcc .\Question6.c
PS D:\MCA\MCA-DSA\LAB-6> .\a.exe
Enter an integer: 5678
Five Thousand Six Hundred and Seventy Eight
PS D:\MCA\MCA-DSA\LAB-6>
```

Data Structure

Lab-7

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Exp no 1: WAP to demonstrate the union's effectiveness over structure. You can use any previously given structure program to depict the idea.

```c
#include <stdio.h>

// Define a structure for a rectangle
struct Rectangle
{
    char shape_type; // 'R' for rectangle
    float length;
    float width;
};

// Define a structure for a circle
struct Circle
{
    char shape_type; // 'C' for circle
    float radius;
};

// Define a union to store either a rectangle or a circle
union Shape
{
    char shape_type; // To determine the type of shape ('R' for
rectangle, 'C' for circle)
    struct Rectangle rectangle;
    struct Circle circle;
};

int main()
{
    union Shape shape;

    // Store a rectangle in the union
    shape.shape_type = 'R';
    shape.rectangle.length = 5.0;
    shape.rectangle.width = 3.0;

    printf("Shape Type: %c\n", shape.shape_type);
    printf("Rectangle Length: %.2f\n", shape.rectangle.length);
    printf("Rectangle Width: %.2f\n", shape.rectangle.width);

    // Store a circle in the same union
    shape.shape_type = 'C';
    shape.circle.radius = 2.5;

    printf("Shape Type: %c\n", shape.shape_type);
    printf("Circle Radius: %.2f\n", shape.circle.radius);
```

```
    return 0;
}
```

```
PS D:\MCA\MCA-DSA> cd .\LAB-7\
PS D:\MCA\MCA-DSA\LAB-7> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-7> .\a.exe
Shape Type: R
Rectangle Length: 5.00
Rectangle Width: 3.00
Shape Type: C
Circle Radius: 2.50
```

Experiment no 2: WAP to demonstrate the various run-time memory allocation approaches like

a. Malloc

b. Calloc

c. Free

d. Realloc

For implementing this, make use of array, function, and wherever necessary pointer.

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

// Function to allocate memory using malloc
int *allocateWithMalloc(int size)
{
    int *arr = (int *)malloc(size * sizeof(int));
    if (arr == NULL)
    {
        printf("Memory allocation with malloc failed.\n");
        exit(1);
    }
    return arr;
}

// Function to allocate memory using calloc
int *allocateWithCalloc(int size)
{
    int *arr = (int *)calloc(size, sizeof(int));
    if (arr == NULL)
    {
```

```c
        printf("Memory allocation with calloc failed.\n");
        exit(1);
    }
    return arr;
}

// Function to reallocate memory using realloc
int *reallocate(int *arr, int newSize)
{
    int *newArr = (int *)realloc(arr, newSize * sizeof(int));
    if (newArr == NULL)
    {
        printf("Memory reallocation with realloc failed.\n");
        free(arr); // Release the original memory
        exit(1);
    }
    return newArr;
}

// Function to print an array
void printArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main()
{
    int *dynamicArray = NULL;
    int size = 5;

    // Allocate memory using malloc
    dynamicArray = allocateWithMalloc(size);

    // Initialize the array
    for (int i = 0; i < size; i++)
    {
        dynamicArray[i] = i + 1;
    }

    printf("Array allocated with malloc: ");
    printArray(dynamicArray, size);

    // Reallocate memory using realloc
    size = 10;
```

```c
    dynamicArray = reallocate(dynamicArray, size);

    // Initialize the additional elements
    for (int i = 5; i < size; i++)
    {
        dynamicArray[i] = i + 1;
    }

    printf("Array reallocated with realloc: ");
    printArray(dynamicArray, size);

    // Deallocate memory using free
    free(dynamicArray);
    dynamicArray = NULL;

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-7> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-7> .\a.exe
Array allocated with malloc: 1 2 3 4 5
Array reallocated with realloc: 1 2 3 4 5 6 7 8 9 10
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-8

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

Write code to implement Stack Data structures. Implement the Push and Pop operation in the stack.

```c
#include <stdio.h>

#include <stdlib.h>

#define SIZE 4

int top = -1, stack[SIZE];
void push();
void pop();
void display();
void peek();

int main()
{
    int choice;

    while (1)
    {
        printf("\nPerform operations on the stack:");
        printf("\n1.Push the element\n2.Pop the
element\n3.Display\n4.Peek\n5.Exit");
        printf("\n\nEnter the choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            push();
            break;
        case 2:
            pop();
            break;
        case 3:
            display();
            break;
        case 4:
            peek();
            break;
        case 5:
            exit(0);

        default:
            printf("\nInvalid choice!!");
        }
    }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
}

void push()
{
    int x;

    if (top == SIZE - 1)
    {
        printf("\nOverflow!!");
    }
    else
    {
        printf("\nEnter the element to be added onto the stack: ");
        scanf("%d", &x);
        top = top + 1;
        stack[top] = x;
    }
}

void pop()
{
    if (top == -1)
    {
        printf("\nUnderflow!!");
    }
    else
    {
        printf("\nPopped element: %d", stack[top]);
        top = top - 1;
    }
}

void display()
{
    if (top == -1)
    {
        printf("\nUnderflow!!");
    }
    else
    {
        printf("\nElements present in the stack: \n");
        for (int i = top; i >= 0; --i)
            printf("%d\n", stack[i]);
    }
}
void peek()
{
    if (top == -1)
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    {
        printf("Stack is empty!");
    }
    else
    {
        printf("Topmost element of the stack is: %d", stack[top]);
    }
}
```

```
PS D:\MCA\MCA-DSA\LAB-8> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-8> .\a.exe

Perform operations on the stack:
1.Push the element
2.Pop the element
3.Display
4.Peek
5.Exit

Enter the choice: 1

Enter the element to be added onto the stack: 10

Perform operations on the stack:
1.Push the element
2.Pop the element
3.Display
4.Peek
5.Exit

Enter the choice: 1

Enter the element to be added onto the stack: 20
```

Kindly use the stack data structure to reverse a number/string and if the number is a palindrome, print that number/string.

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

#define MAX_SIZE 100

// Define the stack data structure
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
struct Stack
{
    char items[MAX_SIZE];
    int top;
};

// Initialize the stack
void initialize(struct Stack *stack)
{
    stack->top = -1;
}

// Check if the stack is empty
int isEmpty(struct Stack *stack)
{
    return stack->top == -1;
}

// Check if the stack is full
int isFull(struct Stack *stack)
{
    return stack->top == MAX_SIZE - 1;
}

// Push a character onto the stack
void push(struct Stack *stack, char value)
{
    if (isFull(stack))
    {
        printf("Stack is full. Cannot push %c.\n", value);
        return;
    }
    stack->items[++stack->top] = value;
}

// Pop a character from the stack
char pop(struct Stack *stack)
{
    if (isEmpty(stack))
    {
        printf("Stack is empty. Cannot pop.\n");
        exit(1);
    }
    return stack->items[stack->top--];
}

int main()
{
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
    struct Stack stack;
    initialize(&stack);

    char input[MAX_SIZE];
    printf("Enter a string: ");
    scanf("%s", input);

    int length = strlen(input);
    int i;

    // Push each character onto the stack
    for (i = 0; i < length; i++)
    {
        push(&stack, input[i]);
    }

    char reversed[MAX_SIZE];
    int j = 0;

    // Pop characters from the stack to reverse the input
    while (!isEmpty(&stack))
    {
        reversed[j++] = pop(&stack);
    }

    reversed[j] = '\0'; // Null-terminate the reversed string

    printf("Reversed: %s\n", reversed);

    // Check if the input is a palindrome
    if (strcmp(input, reversed) == 0)
    {
        printf("%s is a palindrome!\n", input);
    }
    else
    {
        printf("%s is not a palindrome.\n", input);
    }

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-8> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-8> .\a.exe
Enter a string: anshu
Reversed: uhsna
anshu is not a palindrome.
PS D:\MCA\MCA-DSA\LAB-8> .\a.exe
Enter a string: hllh
Reversed: hllh
hllh is a palindrome!
PS D:\MCA\MCA-DSA\LAB-8>
```

For the given string 2+3*5+8/2+6, convert this into postfix and eventually solve this using Stack.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h> // Include ctype.h for isdigit function

#define MAX_SIZE 100

// Define a stack data structure for characters
struct Stack
{
    char items[MAX_SIZE];
    int top;
};

// Initialize the stack
void initialize(struct Stack *stack)
{
    stack->top = -1;
}

// Check if the stack is empty
int isEmpty(struct Stack *stack)
{
    return stack->top == -1;
}

// Check if the stack is full
int isFull(struct Stack *stack)
{
    return stack->top == MAX_SIZE - 1;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```c
// Push a character onto the stack
void push(struct Stack *stack, char value)
{
    if (isFull(stack))
    {
        printf("Stack is full. Cannot push %c.\n", value);
        return;
    }
    stack->items[++stack->top] = value;
}

// Pop a character from the stack
char pop(struct Stack *stack)
{
    if (isEmpty(stack))
    {
        printf("Stack is empty. Cannot pop.\n");
        exit(1);
    }
    return stack->items[stack->top--];
}

// Get the precedence of an operator
int getPrecedence(char operator)
{
    if (operator== '+' || operator== '-')
        return 1;
    if (operator== '*' || operator== '/')
        return 2;
    return 0; // Lower precedence for other characters (operands,
parentheses)
}

// Convert infix expression to postfix
void infixToPostfix(char *infix, char *postfix)
{
    struct Stack operatorStack;
    initialize(&operatorStack);

    int infixLength = strlen(infix);
    int postfixIndex = 0;

    for (int i = 0; i < infixLength; i++)
    {
        char currentChar = infix[i];

        if (isdigit(currentChar))
        {
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
            postfix[postfixIndex++] = currentChar; // Operand, add to
postfix
        }
        else if (currentChar == '(')
        {
            push(&operatorStack, currentChar);
        }
        else if (currentChar == ')')
        {
            while (!isEmpty(&operatorStack) &&
operatorStack.items[operatorStack.top] != '(')
            {
                postfix[postfixIndex++] = pop(&operatorStack);
            }
            pop(&operatorStack); // Pop and discard '('
        }
        else
        {
            // Operator
            while (!isEmpty(&operatorStack) && getPrecedence(currentChar)
<= getPrecedence(operatorStack.items[operatorStack.top]))
            {
                postfix[postfixIndex++] = pop(&operatorStack);
            }
            push(&operatorStack, currentChar);
        }
    }

    // Pop any remaining operators from the stack
    while (!isEmpty(&operatorStack))
    {
        postfix[postfixIndex++] = pop(&operatorStack);
    }

    postfix[postfixIndex] = '\0'; // Null-terminate the postfix string
}

// Evaluate a postfix expression
int evaluatePostfix(char *postfix)
{
    struct Stack operandStack;
    initialize(&operandStack);

    int postfixLength = strlen(postfix);

    for (int i = 0; i < postfixLength; i++)
    {
        char currentChar = postfix[i];
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        if (isdigit(currentChar))
        {
            push(&operandStack, currentChar - '0'); // Convert char to
int and push as operand
        }
        else
        {
            int operand2 = pop(&operandStack);
            int operand1 = pop(&operandStack);

            switch (currentChar)
            {
            case '+':
                push(&operandStack, operand1 + operand2);
                break;
            case '-':
                push(&operandStack, operand1 - operand2);
                break;
            case '*':
                push(&operandStack, operand1 * operand2);
                break;
            case '/':
                push(&operandStack, operand1 / operand2);
                break;
            }
        }
    }

    return pop(&operandStack); // Result is on top of the operand stack
}

int main()
{
    char infixExpression[] = "2+3*5+8/2+6";
    char postfixExpression[MAX_SIZE];

    infixToPostfix(infixExpression, postfixExpression);
    printf("Infix Expression: %s\n", infixExpression);
    printf("Postfix Expression: %s\n", postfixExpression);

    int result = evaluatePostfix(postfixExpression);
    printf("Result: %d\n", result);

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-8> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-8> .\a.exe
Infix Expression: 2+3*5+8/2+6
Postfix Expression: 235*+82/+6+
Result: 27
PS D:\MCA\MCA-DSA\LAB-8>
```

Implement a functionality where a dedicated memory which was allocated using Malloc is about to get filled, reallocate the memory if the available memory is less than 10 %.

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *ptr;
    int i, n;
    printf("Enter the number of elements\n");
    scanf("%d", &n);
    printf("Entered no is %d", n);
    ptr = (int *)malloc(n * sizeof(int));
    if (ptr == NULL)
    {
        printf("Memeory not allocated\n");
        exit(0);
    }
    else if ()
        else
        {
            printf("Memory is allocated\n");
            for (i = 0; i < n; i++)
            {
                ptr[i] = i;
            }
            printf("Elemets in the array are:\n");
            for (i = 0; i < n; i++)
            {
                printf("%d", ptr[i]);
            }
        }
    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-9

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

1. Create Queue
2. Perform Enqueue and Deque operations on Queue.
3. Traverse the queue and print its element.
4. Print underflow and overflow when desired conditions are not met.

Reverse the elements of Queue using recursion

```c
#include <stdio.h>
#define N 5

int front = -1, rear = -1;
int queue[N];

void enqueue(int x)
{
    if (rear == N - 1)
    {
        printf("Overflow\n");
    }
    else if (front == -1 && rear == -1)
    {
        front = rear = 0;
        queue[rear] = x;
    }
    else
    {
        rear++;
        queue[rear] = x;
    }
}

void dequeue()
{
    if (front == -1 && rear == -1)
    {
        printf("Underflow\n");
    }
    else if (front == rear)
    {
        printf("The dequeue element is %d\n", queue[front]);
        front = rear = -1;
    }
    else
    {
        printf("The dequeue element is %d\n", queue[front]);
        front++;
    }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
}

void peek()
{
    if (front == -1 && rear == -1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        printf("%d\n", queue[front]);
    }
}

// Recursive function to reverse the queue
void reverseQueue(int i)
{
    if (i <= rear)
    {
        reverseQueue(i + 1);
        printf("%d ", queue[i]);
    }
}

void display()
{
    if (front == -1 && rear == -1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        for (int i = front; i <= rear; i++)
        {
            printf("%d ", queue[i]);
        }
        printf("\n");
    }
}

int main()
{
    int choice, element;

    do
    {
        printf("Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to
display, 5 to reverse display, and 0 to exit: ");
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter the element to enqueue: ");
            scanf("%d", &element);
            enqueue(element);
            break;

        case 2:
            dequeue();
            break;

        case 3:
            peek();
            break;

        case 4:
            display();
            break;

        case 5:
            printf("Reversed queue elements: ");
            reverseQueue(front);
            printf("\n");
            break;

        case 0:
            printf("Exiting the program\n");
            break;

        default:
            printf("Invalid choice\n");
        }
    } while (choice != 0);

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-9> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-9> .\a.exe
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 2
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 4
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 6
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 8
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 4
2 4 6 8
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 5
Reversed queue elements: 8 6 4 2
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 0
Exiting the program
```

Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-10

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

WAP to populate an array of 'n' elements using a random function. Share the time complexity for all the experiments, n should be large enough to see the difference in execution.

1. Implement Insertion sort in the above data set.

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

// Function to generate random elements in the array
void populateArrayRandomly(int arr[], int n)
{
    srand(time(NULL)); // Seed for random number generation
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand() % 100; // Generate random numbers between 0 and
99
    }
}

void insertionSort(int arr[], int n)
{
    int i, j;
    for (i = 1; i < n; i++)
    {
        int temp = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > temp)
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = temp;
    }
}

int main()
{
    int n = 10; // Choose a sufficiently large value for n
    int arr[n];

    // Populate array with random elements
    populateArrayRandomly(arr, n);

    printf("Original array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
    }

    // Perform insertion sort
    insertionSort(arr, n);

    printf("\nSorted array using Insertion Sort algorithm: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    printf("\n");

    return 0;
}
```

```
PS D:\MCA\MCA-DSA\LAB-10> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-10> .\a.exe
Original array: 1 17 26 50 87 82 63 83 42 45
Sorted array using Insertion Sort algorithm: 1 17 26 42 45 50 63 82 83 87
PS D:\MCA\MCA-DSA\LAB-10>
```

Implement Selection sort in the above data set.

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

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void selectionSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n - 1; i++)
    {
        int min = i;
        for (j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[min])
            {
                min = j;
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
            }
        }
        // Swap the elements
        if (min != i)
        {
            swap(&arr[i], &arr[min]);
        }
    }
}

// Function to generate random elements in the array
void populateArrayRandomly(int arr[], int n)
{
    srand(time(NULL)); // Seed for random number generation
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand() % 100; // Generate random numbers between 0 and
99
    }
}

int main()
{
    int n = 10; // Choose a sufficiently large value for n
    int arr[n];

    // Populate array with random elements
    populateArrayRandomly(arr, n);

    printf("Original array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    // Perform selection sort
    selectionSort(arr, n);

    printf("\nSorted array using Selection Sort algorithm: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    printf("\n");

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-10> gcc .\Question2.c
PS D:\MCA\MCA-DSA\LAB-10> .\a.exe
Original array: 20 64 10 60 35 29 21 63 6 8
Sorted array using Selection Sort algorithm: 6 8 10 20 21 29 35 60 63 64
PS D:\MCA\MCA-DSA\LAB-10>
```

Implement Quicksort in the above data set.

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

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int partition(int a[], int low, int high)
{
    int pivot = a[low];
    int i, j;
    i = low;
    j = high;
    while (i < j)
    {
        do
        {
            i++;
        } while (a[i] <= pivot);

        do
        {
            j--;
        } while (a[j] > pivot);

        if (i < j)
        {
            swap(&a[i], &a[j]);
        }
    }
    swap(&a[low], &a[j]);
    return j;
}

void quicksort(int a[], int low, int high)
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
{
    if (low < high)
    {
        int j = partition(a, low, high);
        quicksort(a, low, j);
        quicksort(a, j + 1, high);
    }
}

// Function to generate random elements in the array
void populateArrayRandomly(int a[], int n)
{
    srand(time(NULL)); // Seed for random number generation
    for (int i = 0; i < n; i++)
    {
        a[i] = rand() % 100; // Generate random numbers between 0 and 99
    }
}

int main()
{
    int n = 10; // Choose a sufficiently large value for n
    int arr[n];

    // Populate array with random elements
    populateArrayRandomly(arr, n);

    printf("Original array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    // Perform quick sort
    quicksort(arr, 0, n - 1);

    printf("\nSorted array using Quick Sort algorithm: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    printf("\n");

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-10> gcc .\Question3.c
PS D:\MCA\MCA-DSA\LAB-10> .\a.exe
Original array: 4 17 0 17 36 18 94 69 61 48
Sorted array using Quick Sort algorithm: 0 4 17 17 18 36 61 69 94 48
PS D:\MCA\MCA-DSA\LAB-10>
```

Implement Merge Sort using the above dataset.

```c
#include <stdio.h>

void merge(int a[], int lb, int mid, int ub)
{
    int i, j, k;
    i = lb;
    j = mid + 1;
    k = lb;
    int b[ub + 1];

    while (i <= mid && j <= ub)
    {
        if (a[i] < a[j])
        {
            b[k] = a[i];
            i++;
        }
        else
        {
            b[k] = a[j];
            j++;
        }
        k++;
    }

    if (i > mid)
    {
        while (j <= ub)
        {
            b[k] = a[j];
            j++;
            k++;
        }
    }
    else
    {
        while (i <= mid)
        {
            b[k] = a[i];
            i++;
            k++;
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        }
    }

    for (k = lb; k <= ub; k++)
    {
        a[k] = b[k];
    }
}
void mergeSort(int a[], int lb, int ub)
{
    if (lb < ub)
    {
        int mid = (lb + ub) / 2;
        mergeSort(a, lb, mid);
        mergeSort(a, mid + 1, ub);
        merge(a, lb, mid, ub);
    }
}

int main()
{
    int arr[] = {15, 5, 24, 8, 1, 3, 16, 10, 20};
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Original array is: \n");
    for (int i = 0; i < n; i++)
    {
        printf("%d, ", arr[i]);
    }
    mergeSort(arr, 0, n - 1);

    printf("\n\nSorted array using Merge Sort algorithm is: \n");
    for (int i = 0; i < n; i++)
    {
        printf("%d, ", arr[i]);
    }
    return 0;
}
```

```
PS D:\MCA\MCA-DSA\SearchingAndSorting> gcc .\mergeSort.c
PS D:\MCA\MCA-DSA\SearchingAndSorting> .\a.exe
Original array is:
15, 5, 24, 8, 1, 3, 16, 10, 20,

Sorted array using Merge Sort algorithm is:
1, 3, 5, 8, 10, 15, 16, 20, 24,
```

Name:-Aakash Bhatt
Sap-id:- 500124633