DSA LAB – 01

1. Take n numbers as input in to an 1-d dynamic array A from user; write a C program to compute and
display the minimum, maximum, sum, and average of the elements in A.
Sample Input:

$$n = 5, A[\,] = 10, 20, 30, 40, 50$$

Sample Output:

$$Minimum = 10$$
$$Maximum = 50$$
$$Sum = 150$$
$$Average = 30$$

Code:-

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


int main(){
    int n,sum,minimum=0,maximum=0;
    printf("Enter the number of elements:-");
    scanf("%d",&n);
    int *arr = (int*)calloc(n,sizeof(int));
    printf("Enter all the elements spearated by space:- ");
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);
    maximum = arr[0];
    minimum = arr[0];
    for(int j=0;j<n;j++){
        sum+=arr[j];
        if (arr[j]<minimum) minimum = arr[j];
        else if (arr[j]>maximum) maximum = arr[j];
    }
    printf("Minimum = %d \n Maximum = %d \n Sum = %d \n Average = %d
\n",minimum,maximum,sum,sum/n);


}
```

Output:-

```
Enter the number of elements:-5
Enter all the elements spearated by space:- 1 2 3 4 5
Minimum = 1
 Maximum = 5
 Sum = 15
 Average = 3
```

2. WAP to reverse the contents of a dynamic array of n elements.

Code:-

```c
#include<stdio.h>

#include<stdlib.h>

int main(){
    int n;
    printf("Enter the number of elements:-");
    scanf("%d",&n);
    int *arr = (int*)calloc(n,sizeof(int));
    printf("Enter all the elements spearated by space:- ");
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);

    int last = n-1;
    for (int j=0;j<n/2;j++){
        int temp = arr[j];
        arr[j] = arr[last];
        arr[last] = temp;
        last--;
    }

    printf("Reversed array is:- ");
    for(int i=0;i<n;i++) printf("%d ",arr[i]);
}
```

Output:-

```
Enter the number of elements:-5

Enter all the elements spearated by space:- 56 44 99 33 22
Reversed array is:- 22 33 99 44 56
```

3 . WAP to search an element in a dynamic array of n numbers.

Code:-

```c
#include<stdio.h>

#include <stdlib.h>

int main(){
    int n;
    printf("Enter the number of elements:-");
    scanf("%d",&n);
    int *arr = (int*)calloc(n,sizeof(int));
    printf("Enter all the elements spearated by space:- ");
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);
    int search;
    printf("Enter the element to search:- ");
    scanf("%d",&search);
    printf("%d is present in index:- ",search);
    for(int j=0;j<n;j++){
        if (arr[j]==search) printf("%d,",j);
    }
}
```

Output:-

```
Enter the number of elements:-5

Enter all the elements spearated by space:- 1 65 42 22 23
Enter the element to search:- 22
22 is present in index:- 3,
```

4. Given an unsorted dynamic array of size n, WAP to find and display the number of elements between two elements a and b (both inclusive).
   Sample Input:

$$arr = [1, 2, 2, 7, 5, 4], a = 2 \; and \; b = 5,$$

   Sample Output:

$$4 \; and \; the \; numbers \; are: 2, 2, 5, 4.$$

Code:-

```c
#include<stdio.h>

#include<stdlib.h>

int main(){
    int n;
    printf("Enter the number of elements:-");
    scanf("%d",&n);
    int *arr = (int*)calloc(n,sizeof(int));
    printf("Enter all the elements spearated by space:- ");
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);
    int search1,search2;
    printf("Enter starting ending elemnts:- ");
    scanf("%d %d",&search1,&search2);
    printf("Elements between %d and %d are ",search1,search2);
    for(int j=0;j<n;j++){
        if (arr[j]>=search1 && arr[j]<=search2) printf("%d,",arr[j]);
    }
}
```

Output:-

```
Enter the number of elements:-5

Enter all the elements spearated by space:- 10 20 30 40 50
Enter starting ending elemnts:- 20 40
Elements between 20 and 40 are 20,30,40,
```

5. Given a dynamic array, WAP to print the next greater element (NGE) for every element. The next greater element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1. E.g. For the input array [2, 5, 3, 9, 7], the next greater elements for each elements are as follows.

Element	NGE
2	5
5	9
3	9
9	-1
7	-1

Code:-

```c
#include <stdio.h>

#include<stdlib.h>
int main()
{
    int n;
    printf("Enter the number of elements:-");
    scanf("%d", &n);
    int *arr = (int*)calloc(n,sizeof(int));
    printf("Enter all the elements spearated by space:- ");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("Element | NGE\n");
    int nge, i, j;
    for (i = 0; i < n; i++)
    {
        nge = -1;
        for (j = i + 1; j < n; j++)
        {
            if (arr[i] < arr[j])
            {
                nge = arr[j];
                break;
            }
        }
        printf("%d\t -- %d \n", arr[i], nge);
    }
}
```

Output:-

```
Enter the number of elements:-5

Enter all the elements spearated by space:- 2 5 3 9 7
Element | NGE
2         -- 5
5         -- 9
3         -- 9
9         -- -1
7         -- -1
```

6. Let A be $n \times n$ square dynamic matrix. WAP by using appropriate user defined functions for the following:

a) Find the number of nonzero elements in A
b) Find the sum of the elements above the leading diagonal.
c) Display the elements below the minor diagonal.
d) Find the product of the diagonal elements.

Code:-

```c
#include<stdio.h>


void non_zero(int n,int (*p)[n]){
    int count = 0;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if (p[i][j]==0) count++;
        }
    }
    printf("Number of non-zero elements is %d\n",count);
}

void sum_diag(int n,int (*p)[n]){
    int sum = 0;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if (j>i) sum+=p[i][j];
        }
    }
    printf("Sum of upper diagonal elements is %d\n",sum);
}


void eb_mindiag(int n,int (*p)[n]){
    printf("The elements below the minor diagonal are:- ");
    for(int i=1;i<n;i++){
        for(int j=n-1;j>0;j--){
```

```
            if (i+j>=n)
            printf("%d,",p[i][j]);
        }
    }
}

void prod_diage(int n,int (*p)[n]){
    int prod=1;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if (i=j)
            prod *= p[i][j];
        }
    }
    printf("\nThe product of the diagonal elements is %d\n",prod);
}


int main(){
    int n;
    printf("Enter n of nxn square matrix:- ");
    scanf("%d",&n);
    int arr[n][n];
    // int *arr = (int*)calloc(n,sizeof(int));
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            scanf("%d",&arr[i][j]);
        }
    }

    printf("The matrix entered by the user is:- \n");
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%d ",arr[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    non_zero(n,arr);
    sum_diag(n,arr);
    eb_mindiag(n,arr);
    prod_diage(n,arr);
}
```

Output:-

```
Enter n of nxn square matrix:- 3

1 2 3 4 5 6 7 8 9
The matrix entered by the user is:-
1 2 3
4 5 6
7 8 9

Number of non-zero elements is 0
Sum of upper diagonal elements is 11
The elements below the minor diagonal are:- 6,9,8,
The product of the diagonal elements is 45
```

7. Define a structure for representing a point in two-dimensional Cartesian co-ordinate system. Now, write a C program to perform the following tasks:
    a. Compute the distance between two given points.

Compute the area of a triangle, given the co-ordinates of its three vertices.

Code:-

```c
#include <stdio.h>

#include <math.h>

float comp_dist(float x1,float y1, float x2, float y2){
    return pow(pow(x2-x1,2)+pow(y2-y1,2),0.5);
}

float comp_area(float x1,float y1, float x2, float y2,float x3, float y3){
    return 0.5*( ( (x1*y2) + (x2*y3) + (x3*y1) ) - ( (x2*y1) + (x3*y2) +
(x1*y3) ) );
}


int main(){
    int choice;
    printf("Enter\n 1 to Compute distance b/w 2 points \n 2 to compute area of
a triangle\n 3 to exit\n Enter your choice:- ");
    scanf("%d",&choice);
    if (choice==1){
        float x1,y1,x2,y2;
        printf("Enter x1 y1 x2 y2:- ");
        scanf("%f %f %f %f",&x1,&y1,&x2,&y2);
        float k = comp_dist(x1,y1,x2,y2);
        printf("The distance between the 2 points is %.2f",k);
    }
    else if (choice==2){
        float x1,y1,x2,y2,x3,y3;
```

```
        printf("Enter x1 y1 x2 y2 x3 y3:- ");
        scanf("%f %f %f %f",&x1,&y1,&x2,&y2,&x3,&y3);
        float k = comp_area(x1,y1,x2,y2,x3,y3);
        printf("The area of the triangle is %.2f",k);
    }
}
```

Output:-

```
Enter

 1 to Compute distance b/w 2 points
 2 to compute area of a triangle
 3 to exit
 Enter your choice:- 1
Enter x1 y1 x2 y2:- 1 3 8 9
The distance between the 2 points is 9.22
```

8. Write a C program that:
    a. Uses Structure to store name, roll no., marks, and address of 5 students in C programming subject;
    b. Displays the stored information.

Code:-

```
#include<stdio.h>

#include<stdlib.h>

struct data{
    char name[20];
    int roll;
    int marks_cs;
    char add[20];
};

int main(){
    int n;
    printf("n:- ");
    scanf("%d",&n);

    struct data* info = (struct data*)malloc(n*sizeof(struct data));

    for(int i=0; i<n;i++){
        printf("Enter details of student %d\n",i+1);
        printf("Enter your name:- ");
        scanf("%s",info[i].name);

        printf("Roll number:- ");
```

```
        scanf("%d",&info[i].roll);

        printf("Enter marks in CS:- ");
        scanf("%d",&info[i].marks_cs);

        printf("Enter address:- ");
        scanf("%s",info[i].add);
    }

    for(int i=0; i<n;i++){
        printf("Details of student Entered %d\n",i+1);
        printf("Name :- %s \n ",info[i].name);

        printf("Roll number:- %d\n",info[i].roll);

        printf("marks:- %d\n",info[i].marks_cs);

        printf("address:- %s\n",info[i].add);
    }



}
```

Output:-

```
n:- 1

Enter details of student 1
Enter your name:- Aditya
Roll number:- 2129011
Enter marks in CS:- 99
Enter address:- Bhubaneswar
Details of student Entered 1
Name :- Aditya
 Roll number:- 2129011
marks:- 99
address:- Bhubaneswar
```