# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 5

1) Program to demonstrate Sorting algorithms

SELECTION SORT CODE

#include <stdio.h>

#include <stdlib.h>

#define size 20

int small(int a[], int k, int n);

void selection(int a[], int n);

int main()

{

    int a[size], n, i;

    printf("size:- ");

    scanf("%d", &n);

    printf("\nelements are :- \n");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &a[i]);

    }

    selection(a, n);

    printf("Sorted elements are:-\n");

    for (i = 0; i < n; i++)

    {

        printf("%d\t", a[i]);

    }

}

int small(int a[], int k, int n)

{

    int pos = k, s = a[k], i;

    for (i = k + 1; i < n; i++)

    {

        if (a[i] < s)

        {

            s = a[i];

            pos = i;

        }

    }

    return pos;

}

void selection(int a[], int n)

{

    int k, pos, temp;

    for (k = 0; k < n; k++)

    {

        pos = small(a, k, n);

        temp = a[k];

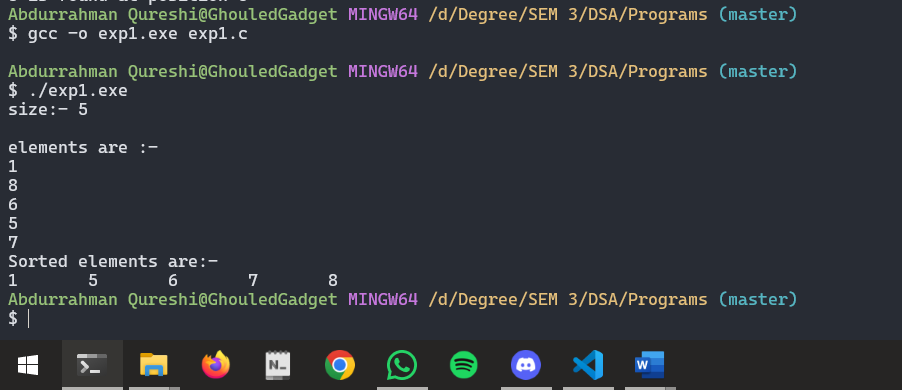
        a[k] = a[pos];

        a[pos] = temp;

    }

}

OUTPUT



BUBBLE SORT CODE

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

int main()

{

    int a[20];

    int i, j, temp, n;

    printf("Enter size of an array\n");

    scanf("%d", &n);

    printf("Enter elements\n");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &a[i]);

    }

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < n - 1; j++)

        {

            if (a[j] > a[j + 1])

            {

                temp = a[j];

                a[j] = a[j + 1];

                a[j + 1] = temp;

            }

        }

    }

    printf("sorted elements are\n");

    for (i = 0; i < n; i++)

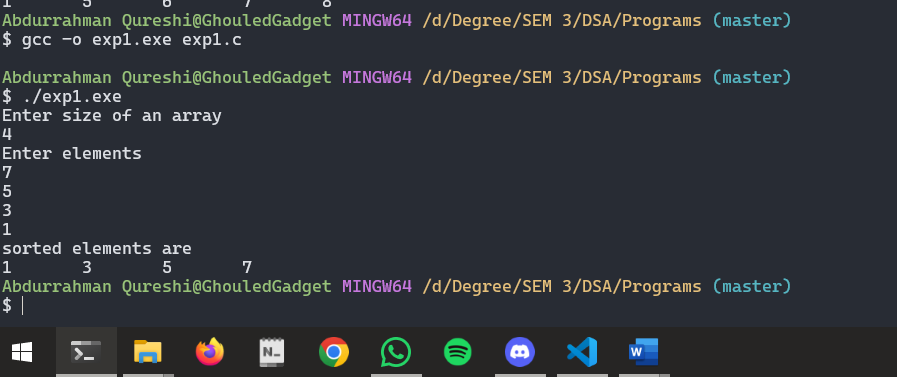
    {

        printf("%d\t", a[i]);

    }

}

OUTPUT



INSERTION SORT CODE

#include <stdio.h>

#include <stdlib.h>

#define size 10

void insertion\_sort(int a[], int n);

int main()

{

    int a[size], i, n;

    printf("\n Enter size of array:\n");

    scanf("%d", &n);

    printf("\n Enter elements:\n");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &a[i]);

    }

    insertion\_sort(a, n);

    printf("\n Sorted elements are:\n");

    for (i = 0; i < n; i++)

    {

        printf("%d\t", a[i]);

    }

}

void insertion\_sort(int a[], int n)

{

    int i, j, temp;

    for (i = 1; i < n; i++)

    {

        temp = a[i];

        j = i - 1;

        while ((temp < a[j]) && (j >= 0))

        {

            a[j + 1] = a[j];

            j--;

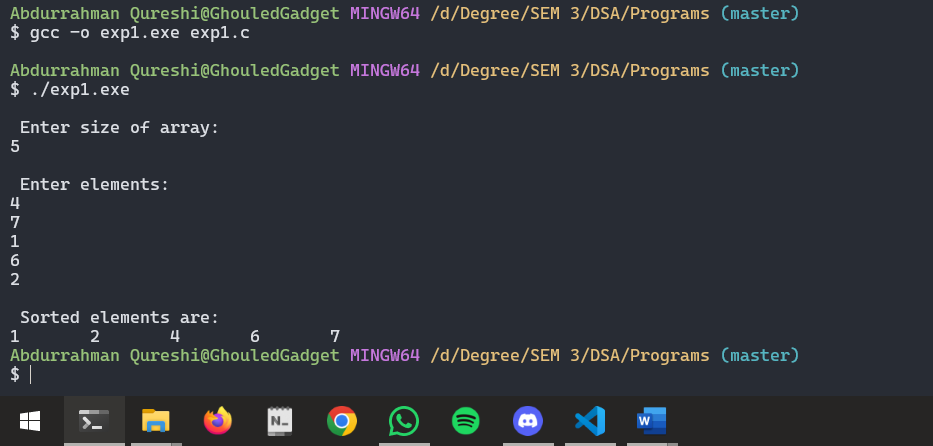
        }

        a[j + 1] = temp;

    }

}

OUTPUT



HEAP SORT CODE

#include <stdio.h>

void swap(int \*a, int \*b)

{

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void heapify(int arr[], int N, int i)

{

    int largest = i;

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    if (left < N && arr[left] > arr[largest])

        largest = left;

    if (right < N && arr[right] > arr[largest])

        largest = right;

    if (largest != i)

    {

        swap(&arr[i], &arr[largest]);

        heapify(arr, N, largest);

    }

}

void heapSort(int arr[], int N)

{

    for (int i = N / 2 - 1; i >= 0; i--)

        heapify(arr, N, i);

    for (int i = N - 1; i >= 0; i--)

    {

        swap(&arr[0], &arr[i]);

        heapify(arr, i, 0);

    }

}

void printArray(int arr[], int N)

{

    for (int i = 0; i < N; i++)

        printf("%d ", arr[i]);

    printf("\n");

}

int main()

{

    int arr[] = {12, 11, 13, 5, 6, 7};

    int N = sizeof(arr) / sizeof(arr[0]);

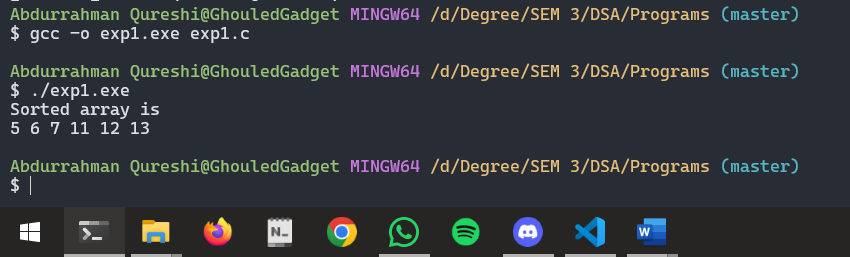
    heapSort(arr, N);

    printf("Sorted array is\n");

    printArray(arr, N);

}

OUTPUT



Tools used :

Software: Dev c++

Hardware: Lab Computers

References: Mam notes.

Conclusion

Heap sort, selection sort, insertion sort, and bubble sort are fundamental sorting algorithms. Heap sort leverages a binary heap for efficient sorting, while selection sort repeatedly finds the smallest element. Insertion sort builds the sorted array incrementally, and bubble sort repeatedly swaps adjacent elements. Each algorithm varies in efficiency and use cases.