**Parth Ashra**

**S.Y IT**

**ROLL NO. 1**

**EXP 9**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

void insertionSort(int arr[], int n);

void main()

{

    int arr[100], i, n, x, choice, flag = 0;

    printf("\t WELCOME TO IMPLEMENTATION OF BINARY SEARCH\n");

    printf("\n Enter the number of elements of the array [maximum size = 100] : ");

    scanf("%d", &n);

    printf("\n Enter %d elements of the array : \n", n);

    for (i = 0; i < n; i++)

    {

        scanf(" %d", &arr[i]);

    }

    insertionSort(arr, n);

    do

    {

        printf("\n Operations available");

        printf("\n 1. Display Sorted List \t 2. Search a particular value \t 3. Exit");

        printf("\n Please Enter your choice : ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

        {

            printf("\n\n The sorted array is : \n");

            for (i = 0; i < n; i++)

            {

                printf(" %d \t", arr[i]);

            }

            break;

        }

        case 2:

        {

            printf("\n Enter the number to be searched : ");

            scanf("%d", &x);

            int flag = -1;

            for (int i = 0; i < n; i++)

            {

                if (arr[i] == x)

                {

                    flag = i;

                    break;

                }

            }

            if (flag != -1)

            {

                printf("\n %d is present in the sorted array at index : %d", x, flag);

            }

            else{

                printf("\n %d does not exist int the array", x);

            }

        }

        case 3:

        {

            printf("\n Program Finished !! Thank You");

            break;

        }

        default:

        {

            printf("\n Please enter a valid choice 1, 2, 3.");

        }

        }

    } while (choice != 3);

    getch();

}

void insertionSort(int arr[], int n)

{

    int i, j, temp;

    for (i = 1; i < n; i++)

    {

        temp = arr[i];

        j = i - 1;

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

        {

            arr[j + 1] = arr[j];

            j--;

        }

        arr[j + 1] = temp;

    }

}

Output

A screenshot of a computer program

Description automatically generated