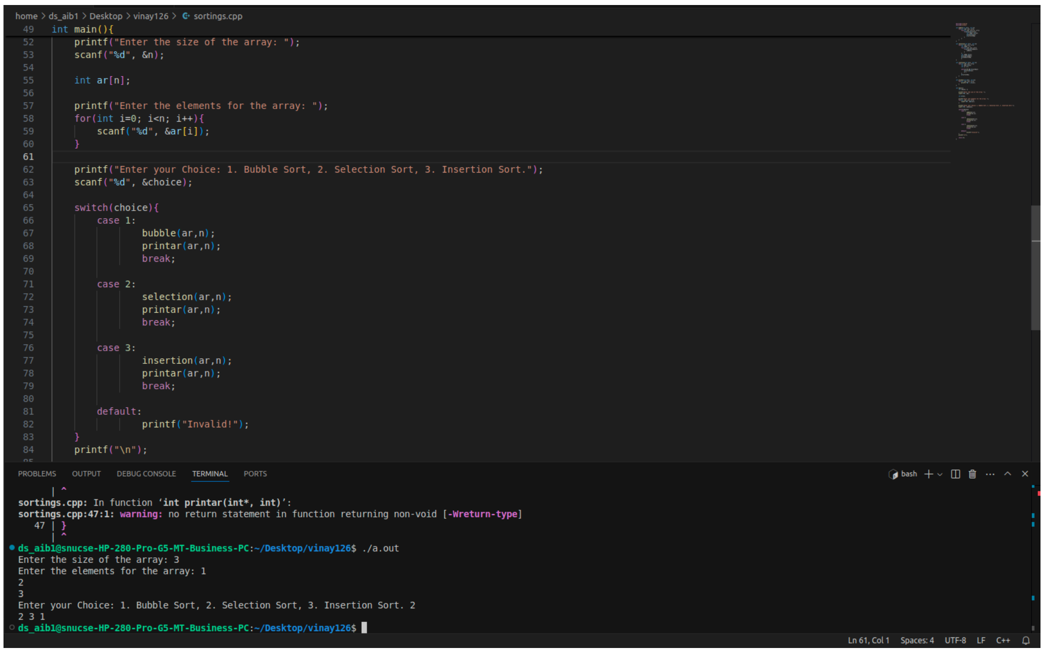
**AIM**: TO BUILD A MENU DRIVEN PROGRAM FOR SORTING THE ELEMENTS IN AN ARRAY.

**ALGORITHM**:

Start
Step 1: Use a function for Bubble, Selection and Insertion Sort.
Step 2: Open the main function defining the choice, size of the array and define the elements in the array
Step 3: Get the input from the user for the choice (which consists of 1. Bubble Sort 2. Selection Sort 3. Insertion Sort 4. Exiting the program).
Step 4: Open switch case, at each case, relating with the choice input from the user.
Step 5: Output Produced.
Stop.

**CODE AND OUTPUT**



home > ds_aib1 > Desktop > vinay126 > sortings.cpp

```
int main(){
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int ar[n];

    printf("Enter the elements for the array: ");
    for(int i=0; i<n; i++){
        scanf("%d", &ar[i]);
    }

    printf("Enter your Choice: 1. Bubble Sort, 2. Selection Sort, 3. Insertion Sort.");
    scanf("%d", &choice);

    switch(choice){
        case 1:
                bubble(ar,n);
                printar(ar,n);
                break;

        case 2:
                selection(ar,n);
                printar(ar,n);
                break;

        case 3:
                insertion(ar,n);
                printar(ar,n);
                break;

        default:
                printf("Invalid!");
    }
    printf("\n");
```

```
      | ^
sortings.cpp: In function ‘int printar(int*, int)’:
sortings.cpp:47:1: warning: no return statement in function returning non-void [-Wreturn-type]
   47 | }
      | ^
ds_aib1@snucse-HP-280-Pro-G5-MT-Business-PC:~/Desktop/vinay126$ ./a.out
Enter the size of the array: 3
Enter the elements for the array: 1
2
3
Enter your Choice: 1. Bubble Sort, 2. Selection Sort, 3. Insertion Sort. 2
2 3 1
ds_aib1@snucse-HP-280-Pro-G5-MT-Business-PC:~/Desktop/vinay126$
```

**AIM**: TO BUILD A MENU DRIVEN PROGRAM FOR SEARCHING AN ELEMENT IN A SORTED ARRAY BY CREATING A USER DEFINED HEADER FILE.

**ALGORITHM**

Start
Step 1: Create a header file named “sort.h”.
Step 2: In that file, different types of sortings are defined.
Step 3: Such header file is called to the main program.
Step 4: Functions for Linear and Binary Search are created.
Step 5: Opening the main function, variables such as choice, size of array, target (number to be found) and result (linear/binary search) are created.

Step 6: Size of the array is defined, values obtained from the user.
Step 7: Selection of Choices and execution of program though switch cases
Step 8: If Linear Search is chosen, the program shows the output of the sorted array (searched using linear sort) and the element present in the certain index.
Step 9: If Binary Search is chosen, the program shows the output of the sorted array(using bubble sort and searched using binary search) and the element present in the certain index.
Step 10: If Exit is chosen, The user gets out of the program.
Stop.

CODE:

```
//Program to search for an element in an array

#include <cstdio>
#include "sort.h"

//Function to search for a given element in an array using Linear Search


int linear(int ar[], int n, int target){
    for (int i=0;i<n;i++){
        if (ar[i]==target){
            return i;
        }
    }
    return -1;
}

//Function to search for a given element in an array using Binary Search

int binary(int ar[], int n, int target){
    int low=0, high=n-1;
    while (low <= high){
        int mid = (low+high)/2;
        if (ar[mid]==target){
            return mid;
        }
        else if (ar[mid]<target){
            low=mid+1;
        }
        else{
            high=mid-1;
        }
    }
    return -1;
}
```

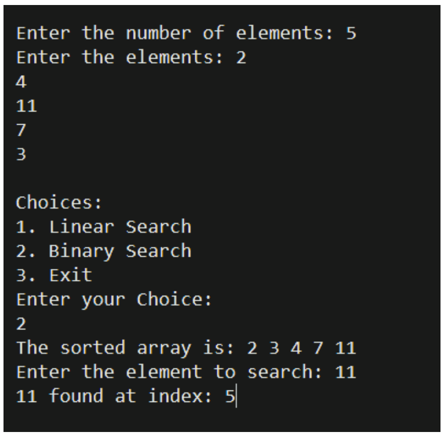
```c
int main(){
    int choice,n,target,result;
    printf("\n Enter the number of elements: ");
    scanf("%d",&n);

    int ar[n];

    printf("Enter the elements: ");
    for (int i=0;i<n;i++){
        scanf("%d",&ar[i]);
    }

    while(1){
        printf("\nChoices: \n");
        printf("1. Linear Search\n");
        printf("2. Binary Search\n");
        printf("3. Exit\n");
        printf("\nEnter your choice: \n");
        scanf("%d",&choice);
        printf("\n");

        switch(choice){
            case 1:
            printf("Enter the element to search: ");
            scanf("%d",&target);
            result = linear(ar,n,target);
            if (result != -1){
                printf("%d  found at index: %d\n",target, result+1);
            }
            else{
                printf("Target not found. \n");
            }
            break;

            case 2:
            bubble(ar,n);
            printar(ar,n);
            printf("Enter the element to search: ");
            scanf("%d",&target);
            result = binary(ar,n,target);
            if (result != -1){
                printf("%d found at index: %d\n",target, result+1 );
            }
            else{
                printf("Target not found. \n");
            }
            break;

            case 3:
            printf("Exited the program.\n");
            return -1;
            break;

            default:
            printf("Invalid choice");
        }
    }
    return 0;
}
```

Header File:
```
//Header File
```

```

#ifndef SORT_H
#define SORT_H

#include<stdio.h>
//Function to sort elements in the array using Bubble Sort

void bubble(int ar[], int n){
    for (int i=0;i<n-1;i++){
        for (int j=0;j<n-i-1;j++){
            if (ar[j]>ar[j+1]){
                int temp = ar[j];
                ar[j]=ar[j+1];
                ar[j+1]=temp;
            }
        }
    }
}

//Function to sort elements in the array using Selection Sort

void selection(int ar[], int n){
    for (int i=0;i<n-1;i++){
        int index=i;
        for (int j=i+1;j<n;j++){
            if (ar[j]<ar[index]){
                index=j;
```

```c
            }
        }
                int temp = ar[i];
                ar[i]=ar[index];
                ar[index]=temp;
    }
}

//Function to sort elements in the array using Insertion Sort

void insertion(int ar[], int n){
    for (int i=1;i<n;i++){
        int key=ar[i];
        int j=i-1;
        while (j >= 0 && ar[j]>key){
            ar[j+1]=ar[j];
            j--;
        }
        ar[j+1]=key;
    }
}

//Function to print the array

void printar(int ar[], int n){
    printf("The sorted array is: \n");
    for (int i=0; i<n; i++){
        printf("%d ",ar[i]);
    } printf("\n");
}

#endif
```

OUTPUT:



```
Enter the number of elements: 5
Enter the elements: 2
4
11
7
3

Choices:
1. Linear Search
2. Binary Search
3. Exit
Enter your Choice:
2
The sorted array is: 2 3 4 7 11
Enter the element to search: 11
11 found at index: 5
```