

PRACTICAL :-03

AIM:- : Create an array of size n and write a program to sort a given array by selection sort and bubble sort.

PROGRAM:- (selection sort)

```
#include<stdio.h>

void main()
{
    int arr[100],m,temp;
    printf("Enter the array size\n");
    scanf("%d",&m);
    printf("Enter the array elements\n");
    for(int i=0; i<m; i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<=m; j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    printf("Array after sorting \n");
```

```

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

```

[OUTPUT]

The screenshot shows a VS Code interface with a file explorer on the left and a terminal on the right. The file explorer shows a project named 'GURU012' with a folder 'Array' containing several files. The terminal shows the following commands and output:

```

PS C:\Users\dajig\OneDrive\Desktop\guru012> cd array
PS C:\Users\dajig\OneDrive\Desktop\guru012\array> gcc selection.c
PS C:\Users\dajig\OneDrive\Desktop\guru012\array> ./a.exe
Enter the array size
5
Enter the array elements
44
56
89
77
99
Array after sorting
99
89
77
56
44
PS C:\Users\dajig\OneDrive\Desktop\guru012\array>

```

The file explorer shows the following files in the 'Array' folder:

- a.exe
- allop.cpp
- array1.cpp
- binarySearch.c
- binarySearch.exe
- bubble.c
- dele.cpp
- delete.cpp
- delete.exe
- ex1.c
- ex2.c
- ex3.c
- ex4.cpp
- ex4.exe
- ex5.cpp
- ex6.cpp
- ex11.cpp
- ex12.cpp
- firstPos.cpp
- mid.cpp
- midPos.cpp
- new.cpp
- one.cpp
- pointer.cpp
- search.c
- selection.c
- zeroAndOnes.exe

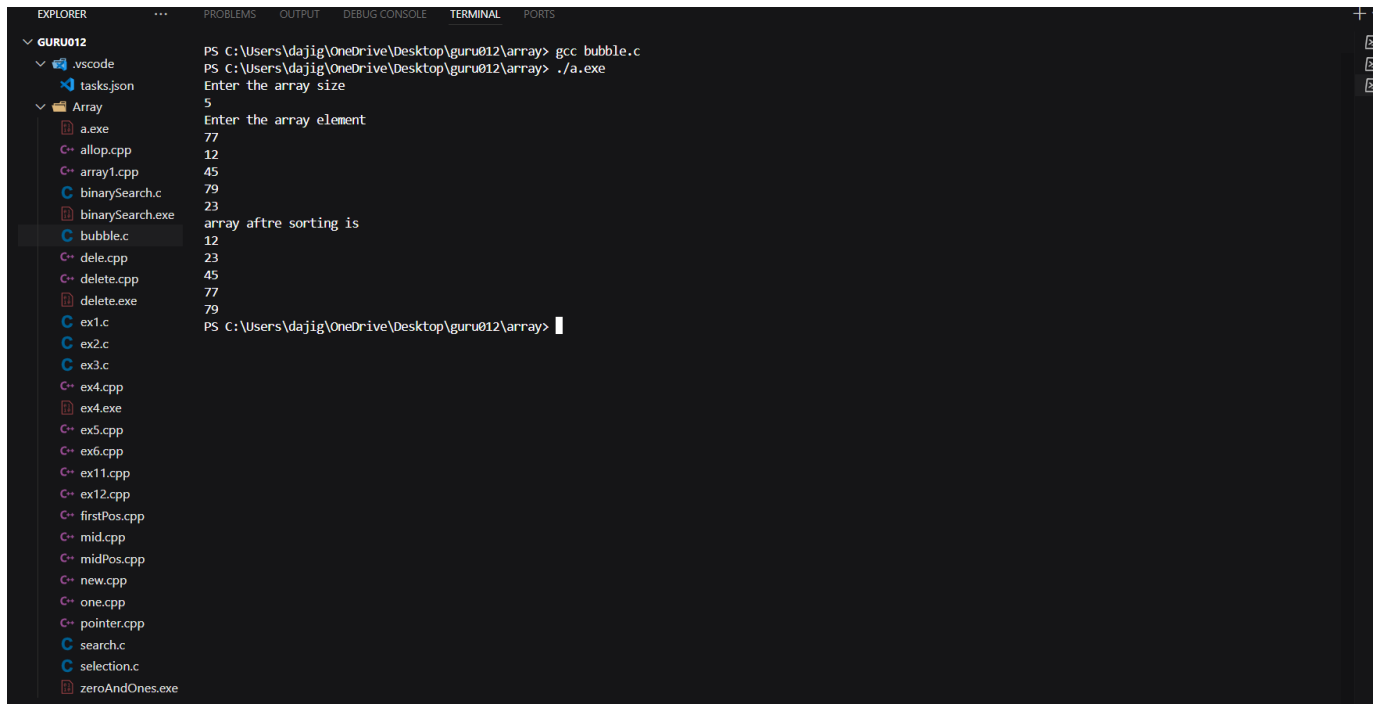
[Bubble Sort]

PROGRAM:-

```
#include<stdio.h>

void main()
{
    int a[100],n,temp;
    printf("Enter the array size\n");
    scanf("%d",&n);
    printf("Enter the array element\n");
    for(int i=0; i<n; i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<=n; j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    printf("array aftr sorting is\n");
    for(int i=0; i<n; i++)
    {
        printf("%d\n",a[i]);
    }
}
```

[OUTPUT]



```
PS c:\Users\dajig\OneDrive\Desktop\guru012\array> gcc bubble.c
PS c:\Users\dajig\OneDrive\Desktop\guru012\array> ./a.exe
Enter the array size
5
Enter the array element
77
12
45
79
23
array aftr sorting is
12
23
45
77
79
PS c:\Users\dajig\OneDrive\Desktop\guru012\array>
```

[B]: Write a program to search any integer in your array using binary search concept.

PROGRAM:-

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

```
void main()
```

```
{
```

```
    int i, j, n, mid, low, high, item, found = 0;
```

```
    printf("Enter the size of array\n");
```

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

```
    int a[n];
```

```
    printf("Enter %d elements:\n", n);
```

```

for (i = 0; i < n; i++)
{
    scanf("%d", &a[i]);
}

printf("Enter element you want to search: \n");
scanf("%d",&item);

low = 0,high = n-1;

for(i=0;i<(n+1)/2;i++){
    mid=(low + high)/2;
    if(item == a[mid]){
        printf("Element found at %d position\n",mid+1);
        found = 1;
        break;
    }
    else if(item < a[mid]){
        high = mid -1;
    }
    else if(item > a[mid]){
        low = mid + 1;
    }
}
if(found != 1){
    printf("Element not found :(\n");
}
}

```

[OUTPUT]

```
Array > C newsearch.c > main()
2 void main()
35 }
36 if(found != 1){
37     printf("Element not found :(\n");
38 }
39 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\dajig\OneDrive\Desktop\guru012\array> gcc newsearch.c
PS C:\Users\dajig\OneDrive\Desktop\guru012\array> ./a.exe
Enter the size of array
5
Enter 5 elements:
14
45
69
78
23
Enter element you want to search:
78
Element found at 4 position
PS C:\Users\dajig\OneDrive\Desktop\guru012\array> 
```

Github Link:- <https://github.com/guru24961/Data-Stracture-practical.git>