ASSIGNMENT-5

1. Write a menu-driven program to perform a Linear Search and Binary Search on an Array

SOLUTION:

```
#include<stdio.h>
#include<stdlib.h>
int binary_search(int n,int arr[],int low,int high,int num){
	int mid=0;
	if(high>=low){
		mid=(low+high)/2;
		if(arr[mid]==num)
		return mid;
		else if(arr[mid]>num)
		return binary_search(n,arr,low,mid-1,num);
		else
		return binary_search(n,arr,mid+1,high,num);
	}
	else
	return -1;
}
void linear_search(int n,int arr[],int num){
	int i,c=0;
	for(i=0;i<n;i++){
		if(arr[i]==num){
			c=1;
			printf("Element Found at Index: %d",i+1);
			break;
		}
	}
	if(c==0)
	printf("Element Not Found in Array!\n");
}
int main(){
	int i,j,n,num,ch,result,temp;
	printf("Enter No. of Elements:");
	scanf("%d",&n);
	int low=0,high=n-1,arr[n];
	printf("Enter Values of Array:\n");
	for(i=0;i<n;i++){
		printf("Array[%d]:",i);
		scanf("%d",&arr[i]);
	}
	printf("Enter Value to be Searched:");
```

```
    scanf("%d",&num);
    printf("\n1. Linear Search\n2. Binary Search\n3. EXIT\nEnter your Choice: ");
    scanf("%d",&ch);
    switch(ch){
        case 1:linear_search(n,arr,num);
                break;
        case 2:for(i=0;i<n-1;i++){
                    for(j=0;j<n-i-1;j++){
                        if(arr[j]>arr[j+1]){
                            int temp=arr[j];
                            arr[j]=arr[j+1];
                            arr[j+1]=temp;
                        }
                    }
                }
                printf("Array after Sorting:\n");
                for(i=0;i<n;i++)
                printf("Array[%d]:%d\n",i,arr[i]);
                result=binary_search(n,arr,low,high,num);
                if(result!=-1)
                printf("Element is found at Position:%d",result+1);
                else
                printf("Element Not Found in Aarray!\n");
                break;
        case 3:exit(0);
        default:printf("Invalid Choice!");
                break;
    }
    return 0;
}
```

OUTPUT:

```
Enter No. of Elements:5
Enter Values of Array:
Array[0]:10
Array[1]:90
Array[2]:70
Array[3]:-55
Array[4]:30
Enter Value to be Searched:70

1. Linear Search
2. Binary Search
3. EXIT
Enter your Choice: 2
```

```
Array after Sorting:
Array[0]:-55
Array[1]:10
Array[2]:30
Array[3]:70
Array[4]:90
Element is found at Position:4
```

2. Write a menu-driven program to perform a Bubble, Insertion, or Selection Sort on an Array

SOLUTION:

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

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

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;
	}
}

void SelectionSort(int arr[], int n){
	int i,j,temp,small;
	for(i=0;i<n-1;i++){
		small=i;
		for(j=i+1;j<n;j++){
			if(arr[j]<arr[small]){
```

```
                small=j;
            }
        }
        temp=arr[i];
        arr[i]=arr[small];
        arr[small]=temp;
    }
}

void Display(int arr[], int n){
    int i;
    printf("Sorted Array:\n");
    for(i=0;i<n;i++){
        printf("Array[%d]: %d\n",i+1,arr[i]);
    }
}
int main() {
    int i,n,ch;
    printf("Enter the Number of Elements you want in your Array: ");
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Array[%d]: ",i+1);
        scanf("%d",&arr[i]);
    }
    printf("-----MENU-----\n1. Bubble Sort\n2. Insertion Sort\n3. Selection Sort\n4.
EXIT\nEnter Your Choice: ");
    scanf("%d",&ch);
    switch(ch){
        case 1:BubbleSort(arr,n);
                Display(arr,n);
                break;

        case 2:InsertionSort(arr,n);
                Display(arr,n);
                break;

        case 3:SelectionSort(arr,n);
                Display(arr,n);
                break;

        case 4:exit(0);

        default:printf("Invalid Input!");
                break;
    }
    return 0;
}
```

OUTPUT:

```
Enter the Number of Elements you want in your Array: 5
Array[1]: 10
Array[2]: 90
Array[3]: 50
Array[4]: -25
Array[5]: 30
-----MENU-----
1. Bubble Sort
2. Insertion Sort
3. Selection Sort
4. EXIT
Enter Your Choice: 3
Sorted Array:
Array[1]: -25
Array[2]: 10
Array[3]: 30
Array[4]: 50
Array[5]: 90
```

3. Write a program to perform Quick Sort on an Array

SOLUTION:

```
#include<stdio.h>

int Partition(int arr[], int beg, int end){
        int left,right,temp,loc,flag;
        loc=left=beg;
        right=end;
        flag=0;
        while(flag!=1){
                while ((arr[loc]<=arr[right]) && (loc!=right))
                right--;
                if(loc==right)
                flag=1;
                else if(arr[loc]>arr[right]){
                        temp=arr[loc];
                        arr[loc]=arr[right];
                        arr[right]=temp;
                        loc=right;
                }
                if(flag!=1){
                        while((arr[loc]>=arr[left]) && (loc!=left))
                        left++;
                        if(loc==left)
                        flag=1;
                        else if(arr[loc]<arr[left]){
```

```
                temp=arr[loc];
                arr[loc]=arr[left];
                arr[left]=temp;
                loc=left;
            }
        }
    }
    return loc;
}
void QuickSort(int arr[], int beg, int end){
    int loc;
    if(beg<end){
        loc=Partition(arr,beg,end);
        QuickSort(arr,beg,loc-1);
        QuickSort(arr,loc+1,end);
    }
}
int main() {
    int i,n,ch;
    printf("Enter the Number of Elements you want in your Array: ");
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Array[%d]: ",i+1);
        scanf("%d",&arr[i]);
    }
    QuickSort(arr,0,n-1);
    printf("Sorted Array:\n");
    for(i=0;i<n;i++){
        printf("Array[%d]: %d\n",i+1,arr[i]);
    }
    return 0;
}
```

OUTPUT:

```
Enter the Number of Elements you want in your Array: 5
Array[1]: 10
Array[2]: 90
Array[3]: 50
Array[4]: -75
Array[5]: 30
Sorted Array:
Array[1]: -75
Array[2]: 10
Array[3]: 30
Array[4]: 50
Array[5]: 90
```