1.) Find maximum and minimum element in an array

```
 #include<stdio.h>
void    findMaxMin(int arr[],int n){
       int maxi=arr[0];
       int mini=arr[0];
       for(int i=1;i<n;i++){
              if(maxi<arr[i])maxi=arr[i];
              if(mini>arr[i])mini=arr[i];
       }
     printf("maximum element in array is : %d\n",maxi);
     printf("minimum element in array is : %d\n",mini);
}
int main(){
       int n;
     printf("Enter the size of the array");
        scanf("%d",&n);

        int arr[n];
printf("Enter the element of the array");
       for(int i=0;i<n;i++){
              scanf("%d",&arr[i]);
       }
       findMaxMin(arr,n);
       return 0;
}
```

Output :

Enter the size of the array : 5
Enter the element of the array : 1 4 2 3 2 5 9

maximum element in array is : 9
minimum element in array is : 1

2.) Write a program to search an element using Binary Search

```
    #include<stdio.h>
int Binarysearch(int arr[],int n,int target){
```

```
        int st=0,ed=n-1;
        while(st<=ed){
                int mid=ed-(ed-st)/2;
                if(arr[mid]==target)return mid;
                else if(arr[mid]>target)ed=mid-1;
                else st=mid+1;

        }
        return -1;
}
int main(){
        int n;
  printf("Enter the size of the array");
        scanf("%d",&n);
        int arr[n];
        printf("Enter the element of     the array in increasing order ")
        for(int i=0;i<n;i++){
                scanf("%d",&arr[i]);
        }
        int target;
        printf("Enter the element which you want to search");
        scanf("%d",&target);

        int idx=Binarysearch(arr,n,target);
        if(idx==-1){
                printf("Target element is not present in the array");
        }
        else{
                printf("Element present at the index : %d",idx);
        }

}
```

Output :

Enter the size of the array : 5
Enter the element fo the array in increasing order : 1 4 5 8 9
Enter the element which you want to search : 5

Element present at the index : 2

3.) Write a program to sort the array element using marge sort

```
#include<stdio.h>
void marge(int arr[],int st,int mid,int ed){
	int temp[ed-st+1];
	int i=st;
	int j=mid+1,idx=0;
	while(i<=mid&&j<=ed){
		if(arr[i]<=arr[j]){
			temp[idx++]=arr[i++];

		}
		else{
			temp[idx++]=arr[j++];
		}
	}
	while(i<=mid){
		temp[idx++]=arr[i++];
	}
	while(j<=ed){
		temp[idx++]=arr[j++];
	}
	for(int i=st;i<=ed;i++){
		arr[i]=temp[i-st];
	}
}
void margeSort(int arr[],int st,int ed){
	if(st>=ed)return;
	int mid=(st+ed)/2;
	margeSort(arr,st,mid);
	margeSort(arr,mid+1,ed);
	marge(arr,st,mid,ed);

}
void printArrayElement(int arr[],int n){
	printf("After sorting array element are     :")
	for(int i=0;i<n;i++){
		printf("%d\t",arr[i]);
	}
}
int main(){
	int n;
	printf("Enter the size of the array ");
```

```
        scanf("%d",&n);
        int arr[n];
        printf("Enter the array element ");
        for(int i=0;i<n;i++){
                scanf("%d",&arr[i]);
        }
        margeSort(arr,0,n-1);
        printArrayElement(arr,n);
}
```

Output :

Enter the size of the array : 5
Enter the array element : 1 4 3 2 6

After sorting array element are : 1 2 3 4 6

4.) Write a program to sort the array using Quick sort ?

```
#include<stdio.h>
int partition(int arr[],int st,int ed){
        int pivot=st;
        while(st<=ed){
                while(st<=ed&&arr[st]<=arr[pivot])st++;
                while(ed>=st&&arr[ed]>=arr[pivot])ed--;
                if(st<=ed){
                        int temp=arr[st];
                        arr[st]=arr[ed];
                        arr[ed]=temp;
                }
        }
        int temp=arr[pivot];
        arr[pivot]=arr[ed];
        arr[ed]=temp;
        return ed;
}
void QuickSort(int arr[],int st,int ed){
        if(st>=ed)return;

        int idx= partition(arr,st,ed);
        QuickSort(arr,st,idx-1);
```

```
        QuickSort(arr,idx+1,ed);
}
void printArrayElement(int arr[],int n){
printf("After sorting array element is : \t");
        for(int i=0;i<n;i++){
                printf("%d\t",arr[i]);
        }
}
int main(){
        int n;
        printf("Enter the size of the array");
        scanf("%d",&n);
        int arr[n];
        printf("Enter the element of the array");
        for(int i=0;i<n;i++){
                scanf("%d",&arr[i]);
        }
        QuickSort(arr,0,n-1);
        printArrayElement(arr,n);
return 0;
}
```

Output :

Enter the size of the array : 5
Enter the element of the array : 1 4 3 2 5
After sorting array element is : 1 2 3 4 5

5.) Write a program to sort the array using Insertion sort

```
        #include<stdio.h>
void InserttionSort(int arr[],int n){
        for(int i=1;i<n;i++){
                int j=i-1;
                int ele=arr[i];
                while(j>=0&&arr[j]>arr[j+1]){
                        arr[j+1]=arr[j];
                        j--;
                }
                arr[j+1]=ele;
        }
}
```

```
void printArrayElement(int arr[],int n){
        printf("After sorting array element are : \t");
        for(int i=0;i<n;i++){
                printf("%d\t",arr[i]);
        }
}
int main(){
        int n;
  printf("Enter the size of the array");
    scanf("%d",&n);
    int arr[n];
    printf("Enter the element of the array");
    for(int i=0;i<n;i++){
            scanf("%d",&arr[i]);
    }
    InserttionSort(arr,n);
    printArrayElement(arr,n);
    return 0;
}
```

Output :

Enter the size of the array : 5
Enter the element of the array : 1 3 2 5 4
After sorting array element is : 1 2 3 4 5

6.) write a program to sort the element using selection sort

```
#include<stdio.h>
void SelectionSort(int arr[],int n){
        for(int i=0;i<n;i++){
                int minimumIdx=i;
                for(int j=i+1;j<n;j++){
                        if(arr[j]<arr[minimumIdx]){
                                minimumIdx=j;
                        }
                }
                int temp=arr[minimumIdx];
                arr[minimumIdx]=arr[i];
                arr[i]=temp;
        }
}
void printArrayElement(int arr[],int n){
printf("After sorting array element are :");
```

```
        for(int i=0;i<n;i++){
                printf("%d\t",arr[i]);
        }
}
int main(){
        int n;
       printf("Enter the size of the array");
        scanf("%d",&n);
        printf("Enter the elment of the array")
        int arr[n];
        for(int i=0;i<n;i++){
                scanf("%d",&arr[i]);
        }
        SelectionSort(arr,n);
        printArrayElement(arr,n);
}
```

Output:

Enter the size of array : 4
Enter the array element : 1 5 3 2
After sorting array element are : 1 2 3 5

7.) Write a program to Implement stack using Array

```
#include<stdio.h>
void push(int stack[],int &top,int n,int ele){
        if(isFull(stack,top,n)){
                printf("stack is full ");
        }
       else{
                top=top+1;
        stack[top]=ele;
       }
}
void pop(int stack[],int    &top,int n){
        if(isEmpty(stack,top)){
                printf("Stack is empty");
        }
        else{
                int ele=stack[top];
        top=top-1;
        printf("The deleted element is : %d",ele);
        }
```

```
}
bool isEmpty(int stack[],int &top,int n){
	if(top==-1)return true;
	else return false;
}
bool isFull(int stack[],int &top,int n){
	if(top==n)return true;
	else return false;
}
void peek(int stack[],int &top,int n){
	if(isEmpty(stack,top,n)){
		printf("Stack is empty");
	}
	else{
		printf("The top most element of the stack is : %d",stack[top]);
	}
}
void printstack(int stack,int &top,int n){
	if(isEmpty(stack,top,n)){
		printf("Stack is empty")
	}
	else{
printf("element of stack are ");
		for(int i=top;i>=0;i--){
			printf("%d\t",stack[i]);
		}
	}
}
int main(){
     int n;
     printf("Enter the size of the Stack");
     scanf("%d",&n);
     int stack[n];
     int top=-1;
     push(stack,top,n,10);
     push(stack,top,n,40);
     push(stack,top,n,23);
     push(stack,top,n,45);
     pop(stack,top,n);
     pop(stack,top,n);
   if(isEmpty(stack,top,n)){
        printf("Stack is empty");
   }
   else{
        printf("Stack is not empty");
```

```
    }
    if(isFull(stack,top,n)){
          printf("Stack is Full");
    }
    else{
          printf("Stack is not Full");
    }

    peek(stack,top,n);
    printstack(stack,top,n);

         return 0;

}
```

Output :

Enter the size of the stack : 10
The deleted element is : 45
The deleted element is : 23

stack is not empty
stack is not full
Element of stack are : 40 10

8.) Write a program to create and traverse of linkedlist ?

```
#include<stdio.h>
#include<stddef.h>
struct Node{
       int data;
       struct Node *next;
};
  struct Node *head=NULL;
  void createList(){
         struct Node *t=head;
         while(1){
               int ele;
           printf("enter the elment which you want to insert or enter -1 if
you want to stop insertion \t");
               scanf("%d",&ele);
               if(ele==-1)return;
               struct Node *temp= (struct Node *)malloc(sizeof(struct
Node));
               if(temp==NULL)return;
```

```
            temp->data=ele;
            temp->next=NULL;

            if(!head){
                head=temp;
                t=temp;
            }
            else{
                t->next=temp;
                t=temp;
            }


        }
  }
  void printlist(){
        struct Node *node=head;
printf("Linkedlist is ");
        while(node!=NULL){
            printf("%d\t",node->data);
            node=node->next;
        }
  }

int main(){
     createList();
     printlist();
       return 0;
}
```

Output :

enter the elment which you want to insert or enter -1 if you want to stop insertion : 3
enter the elment which you want to insert or enter -1 if you want to stop insertion : 4
enter the elment which you want to insert or enter -1 if you want to stop insertion : 6
enter the elment which you want to insert or enter -1 if you want to stop insertion : -1

linkedlist is : 3 4 6

9.) write a program to find the maximum and minimum element using recursion

```
#include<stdio.h>
#include<stdlib.h>
int * findmaxmin(int arr[],int st,int ed){
	int *ans= (int *)malloc(2*sizeof(int));
	if(st==ed){
		ans[0]=arr[st];
		ans[1]=arr[ed];
		return ans;
	}
	else if(ed-st==1){
		if(arr[st]>arr[ed]){
			ans[0]=arr[ed];
			ans[1]=arr[st];
		}
		else {
			ans[0]=arr[st];
			ans[1]=arr[ed];
		}
		return ans;

	}
	int mid=(st+ed)/2;
	int *left=findmaxmin(arr,st,mid);
	int *right=findmaxmin(arr,mid+1,ed);
	   if(left[0]<right[0]){
			ans[0]=left[0];
	   }
	   else ans[0]=right[0];
	   if(left[1]>right[1]){
			ans[1]=left[1];
	   }
	   else ans[1]=right[1];
	   free(left);
	   free(right);
	   return ans;
}
int main(){
	int n;
      printf("Enter the size of the array");
	scanf("%d",&n);
```

```
    int arr[n];
printf("Enter the element of the array ");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
   int *ans= findmaxmin(arr,0,n-1);
   printf("minimum element in array is : %d\n",ans[0]);
   printf("maximum element in array is : %d\n",ans[1]);
   return 0;
}
```

Output :

Enter the size of the array : 5
Enter the element of the array : 1 4 3 2 9
mimimum element in array is : 1
maximum element in array is : 9

10.) write a program to print all subset of a set ?

```
 #include<stdio.h>
void printsubsetsum(int arr[],int i,int n,int sum){
    if(i==n){
        printf("%d\t",sum);
        return;
    }
    printsubsetsum(arr,i+1,n,sum+arr[i]);
    printsubsetsum(arr,i+1,n,sum);
}
int main(){
    int n;
    printf("Enter the size of the array");
    scanf("%d",&n);
    printf("Enter the elment of the array")
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    printsubsetsum(arr,0,n,0);
}
```

Output :

Enter the size of the array : 4
Enter the element of the array : 1 4 3 2

10 8 7 5 6 4 3 1 9 7 6
4 5 3 2 0