Name: Gaurav singh

Roll No: 70 Semester: 3 Sem.

Subject: Data structures and Algorithms.

Practical: 1 To study Array ADT and implementation various operations on an Array ADT.

CODE:

```
#include<iostream>
using namespace std ;
struct ArrayADT
{
 int *ptr;
 int capacity;
 int lastindex;
 };
 void createarray(struct ArrayADT *A,int C)
{
   A->ptr=(int*)malloc(C*sizeof(int));
   A->capacity=C;
   A->lastindex=-1;
 }
 void store(struct ArrayADT *A,int element,int index)
 {
  if(index<A->capacity&&index==A->lastindex+1)
  {
  A->ptr[index]=element;
  A->lastindex++;
   }
 else
  cout<<"plz"<<endl ;
```

```
 }
int retrive(struct ArrayADT *A,int index)
{if(index<A->capacity && index<=A->lastindex)
 {
 return(A->ptr[index]) ;
 }
 else
   cout<<"Enter proper index" ;
 return 0;
}
void modify(struct ArrayADT *A,int index,int element)
 { if(index<A->capacity && index<=A->lastindex)
  A->ptr[index]=element ;
 }
void remove(struct ArrayADT *A,int index)
  { int j=A->lastindex-index;
  for(int i=index;i<A->lastindex;i++)
 { A->ptr[index]=A->ptr[index+1] ;
}
A->lastindex--;
}
void indsert(struct ArrayADT *A,int element,int index)
  { if(A->lastindex+1<A->capacity&&index<=A->lastindex)
    {
      int j=A->lastindex;
      for(int i=j;i>=index;i--)
      {
        A->ptr[i+1]=A->ptr[i];
}A->ptr[index]=element;
```

```
      A->lastindex++ ;


    }
    else
      cout<<"Array Full hai dost"<<endl ;
  }

int search(struct ArrayADT *A,int element)
{
   for(int i=0;i<=A->lastindex;i++)
   {
     if(element==A->ptr[i])
     {
       return i;
       break;
     }
   }
   return -1;
}

void show(struct ArrayADT *A)
{
  for(int i=0;i<=A->lastindex;i++)
{
  cout<<A->ptr[i]<<endl;
}
}
void sort(struct ArrayADT *A){
    for(int i = 0; i<A->lastindex; i++){
```

```
        for(int j=i; j<A->lastindex; j++){
            for(int k = 0; k<j; k++){
                if(A->ptr[k]>A->ptr[k+1]){
                    int temp = A->ptr[k];
                    A->ptr[k]=A->ptr[k+1];
                    A->ptr[k+1]=temp;
                }
            }
        }
    }
}
int main()
{
 void createarray(struct ArrayADT *,int);
  void store(struct ArrayADT *,int,int);
  int search(struct ArrayADT *,int ) ;
    void remove(struct ArrayADT *,int );
    void modify(struct ArrayADT *,int ,int );
    int retrive(struct ArrayADT *,int );
struct ArrayADT B;
bool flag=true ;
cout<<"Press 1 to create an Array "<<endl ;
cout<<"Press 2 to store element in array "<<endl ;
cout<<"Press 3 to search element in tha array "<<endl ;
cout<<"Press 4 to remove element in that array "<<endl ;
cout<<"Press 5 to modify element in tha array "<<endl ;
cout<<"Press 6 to retrive element in tha array "<<endl ;
cout<<"Press 7 to insert element in tha array "<<endl ;
cout<<" press 8 to show all elements of that arraty"<<endl;
```

```
cout<<" press 9 to sort"<<endl ;
cout<<" press 10 to sort"<<endl ;

while(flag)
 { cout<<"Enter choice"<<endl ;
int choice;
cin>>choice;

switch(choice)
{ case 1:
  { int Ca;
    cout<<" Enter capacity of that array "<<endl ;
cin>>Ca;
    createarray(&B,Ca);
    cout<<"We Succesfully created an array"<<endl ;
  }
break ;
case 2 :
{ int N,INDEX;

 cout<<" Enter No of element that has to be store ";
cin>>N;
while(N-->0)
{int i;
  cin>>i;
  store(&B,i,B.lastindex+1);
}
break ;
case 3:
```

```
{int i;
  cout<<"Enter the element that has to search in array" ;
  cin>>i;
  int j=search(&B,i);
  if(j!=-1)
  {
    cout<<"element found at index"<<j<<endl ;

  }
  else
    cout<<"element not found"<<endl ;

}
break ;
case 4 :

{  int i;
  cout<<" Enter index no of that element that has to removed" ;
cin>>i;
remove(&B,i);

}
break ;
case 5 :
{int in,el;
  cout<<"Enter the index and element that has to modify" ;
cin>>el>>in ;
modify(&B,el,in);
}
```

```
break ;
case 6 :
{ int i;
 cout<<" Enter the index of which you want to retrive ";
cin>>i;
int ele=retrive(&B,i);
cout<<"Element :"<<ele<<endl ;
}
break ;
case 7:
{int el;
  int in;
  cout<<"Enter element and index that has to be inserted" ;
cin>>el>>in;
indsert(&B,el,in);
}
case 8:
{

show(&B);

}
case 9:
{
  sort(&B);
}
break ;
case 10 :
{
```

```
 flag=false ;
}
break ;
default : {
        cout<<"You have enterd an invalid choice\n";
      }
}
}
}
}
```

Output: