#include <iostream>

using namespace std;

int count =0; //Global variable

////1-addelement in array.

void Addelement(int arr[], int size)

{

if(count<size)

{

for(int i=0; i<size; i++)

{

cout<<"Enter value "<<i<<" = ";

cin>>arr[i];

count++;

}

}

else

{

cout<<"List is full";

}

cout<<endl;

}

////2-Get any element.

void get(int arr[], int size)

{

if(count!=0)

{

int position;

cout<<"enter position whose value you want = ";

cin>>position;

int j = position;

for(int i=0; i<count; i++)

{

if(arr[i+1]==arr[j])

{

cout<<"At "<<j<<" position the number is "<<arr[i]<<endl;

}

}

}

else

{

cout<<"The Array is empty"<<endl;

}

}

////3-Copy array values.

void copylist (int arr[],int size)

{

int list[size];

for ( int i = 0; i<count; i++)

{

list[i]=arr[i];

}

cout<<" Elements copied in list from array "<<endl;

for ( int i = 0; i<count; i++)

{

cout<<" "<<list[i]<<" ";

}

cout <<endl;

}

////4-Update array.

void update (int arr[],int size)

{

int index,j;

cout << " Enter index of array ";

cin >> index;

cout << " Enter update value ";

int val;

cin >>val;

j=index;

for ( int i =0; i<size; i++)

{

if (i==j)

{

arr[j]=val;

}

}

size=size-1;

}

////5-Display array list.

void Traverse(int arr[])

{

if(count!=0)

{

for(int i=0; i<count; i++)

{

cout<<arr[i]<<endl;

}

}

else

{

cout<<"List is empty"<<endl;

}

}

////6-Clear array list.

int clear(int arr[])

{

count=0;

return count;

}

int main()

{

const int size=4;

int xyz[size];

while (true)

{

cout<<endl<<" ADT Operation List "<<endl<<endl;

cout<<"Press 1 \tto Add elements in the array"<<endl;

cout<<"Press 2 \tto get elements at specific place in the array"<<endl;

cout<<"Press 3 \tto copy any list to another list in the array"<<endl;

cout<<"Press 4 \tto replace the element and update the array list "<<endl;

cout<<"Press 5 \tto dispaly the elements of Array"<<endl;

cout<<"Press 6 \tto clear elements in the array"<<endl;

cout<<endl;

int num;

cout <<"Enter number: ";

cin>>num;

cout<<endl;

switch(num)

{

case 1:

{

Addelement(xyz,size);

break;

}

case 2:

{

get(xyz,size);

break;

}

case 3:

{

copylist(xyz,size);

break;

}

case 4:

{

update(xyz,size);

break;

}

case 5:

{

Traverse(xyz);

break;

}

case 6:

{

clear(xyz);

break;

}

default:

{

cout<<"invalid choice "<<endl;

}

}

}

return 0;

}