Question-1 Operations on Array

#include<iostream>

using namespace std;

void deletion(int a[],int n);

void insertion(int a[],int n);

void transverse(int a[],int n);

void location(int a[],int n);

int main()

{

int a[100],n,i,choice;

char action;

do{

cout<<"Enter size of Array:";

cin>>n;

cout<<"Enter Element of size "<<n<<":";

for(i=0;i<n;i++)

{

cin>>a[i];

}

cout<<endl<<endl<<"\t"<<"Enter choice by typing numeric code:";

cout<<endl<<"\t1:Transverse of an Array"<<endl<<"\t2:Insertion of Array"<<endl<<"\t3:Deletion of Array"<<endl<<"\t4:Find Location of Array"<<endl;

cin>>choice;

switch(choice)

{

case 1:

transverse(a,n);

break;

case 2:

insertion(a,n);

break;

case 3:

deletion(a,n);

break;

case 4:

location(a,n);

break;

default:

cout<<"Wrong Input...";

break;

}

cout<<endl<<endl<<"\t"<<"If you want to Perform Action again then Press Y:";

cin>>action;

}while(action=='Y' || action=='y');

}

void deletion(int a[],int n)

{

int pos,element,i;

cout<<"Enter the Element to be Deleted:";

cin>>element;

cout<<"Enter position of Element:";

cin>>pos;

for(i=pos-1;i<n-1;i++)

{

a[i]=a[i+1];

}

n--;

cout<<"After Deletion:";

for(i=0;i<n;i++)

{

cout<<a[i]<<"\t";

}

}

void transverse(int a[],int n)

{

cout<<"Transverse of An Array:";

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

{

cout<<a[i]<<"\n";

}

}

void insertion(int a[],int n)

{ int pos,i,element;

cout<<"Enter the Element to be Inserted:";

cin>>element;

cout<<"Enter position of Element:";

cin>>pos;

for(i=n;i>=pos;i--)

{

a[i]=a[i-1];

}

a[pos-1]=element;

n++;

cout<<"After Insertion:";

for(i=0;i<n;i++)

{

cout<<a[i]<<"\n";

}

}

void location(int a[],int n){

int loc,counter=0,i;

cout<<"Enter the Element which you want to know position:";

cin>>loc;

for(i=0;i<n;i++){

if(a[i]==loc)

cout<<loc<<" found at the position of:"<<i+1<<endl;

counter++;

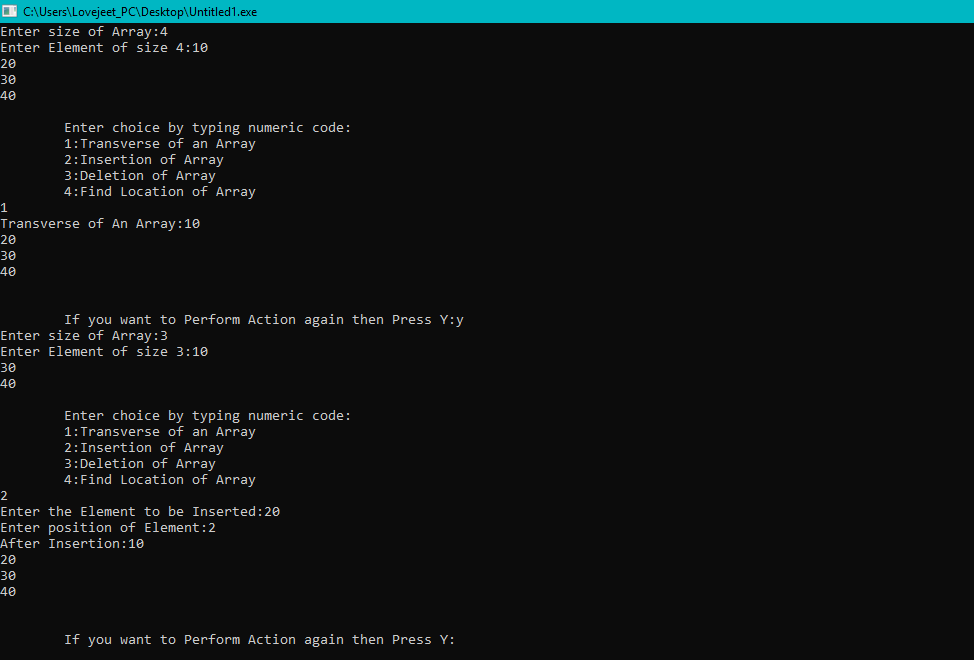
}

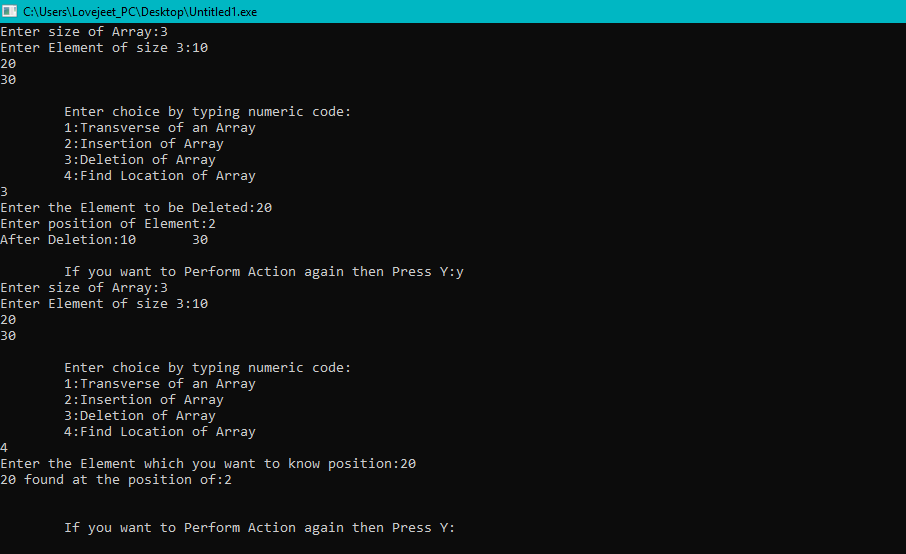
if(counter==0){

cout<<"Entered Element is not found";

}

}





Question-2 Array Typical

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

int i,n,j,counter=0,temp=0;

cout<<"Initialize the Array:";

cin>>n;

int arr[n+1];

cout<<"Enter "<<n<<" Integers Number:";

for(i=0;i<n;i++){

cin>>arr[i];

}

for(i=0;i<n;i++){

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

{

arr[i]=0;

arr[i+1]=2\*arr[i+1];

}

}

for(i=0;i<n;i++){

for(j=i+1;j<n;j++){

if(arr[i]==0){

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

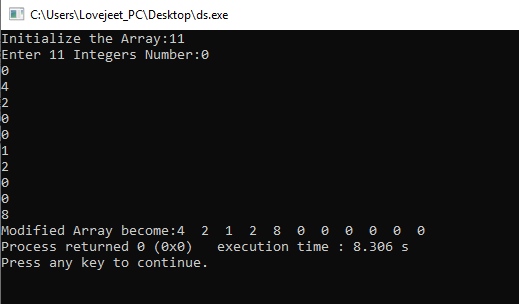
cout<<"Modified Array become:";

for(i=0;i<n;i++){

cout<<arr[i]<<setw(3);

}

}



# Question-3 Two elements whose sum is closest to zero

# include <bits/stdc++.h>

# include <stdlib.h>

# include <math.h>

using namespace std;

void minAbsSumPair(int arr[], int arr\_size)

{

int inv\_count = 0;

int l, r, min\_sum, sum, min\_l, min\_r;

if(arr\_size < 2)

{

cout << "Invalid Input";

return;

}

min\_l = 0;

min\_r = 1;

min\_sum = arr[0] + arr[1];

for(l = 0; l < arr\_size - 1; l++)

{

for(r = l + 1; r < arr\_size; r++)

{

sum = arr[l] + arr[r];

if(abs(min\_sum) > abs(sum))

{

min\_sum = sum;

min\_l = l;

min\_r = r;

}

}

}

cout << "The two elements whose sum is minimum are " << arr[min\_l] << " and " << arr[min\_r];

}

int main()

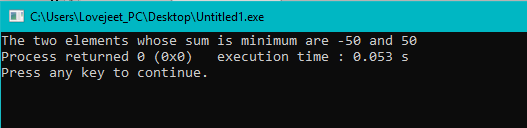
{

int arr[] = {1, 60, -10, 70, -50,50};

minAbsSumPair(arr, 6);

return 0;

}



Question-4 Pattern Making

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int value=n;

int space=0;

int row=1;

int decvalue=n;

while(row<=2\*n+1)

{

int col=1;

while(col<=space)

{

cout<<" ";

col++;

}

col=1;

while(col<=decvalue+1)

{

cout<<value<<" ";

value--;

col++;

}

value=value+2;

col=1;

while(col<=decvalue)

{

cout<<value<<" ";

value++;

col++;

}

if(row<=n)

{

value=value-2;

decvalue--;

space++;

}

else{

decvalue++;

space--;

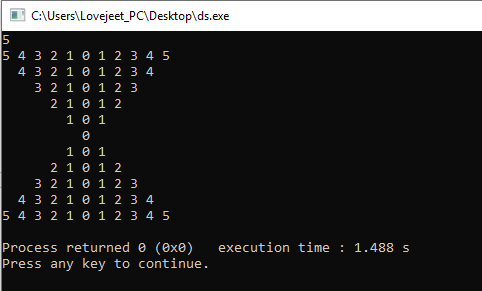
}

row++;

cout<<endl;

}

}



Question-5 Hour & Minute

#include<iostream>

#include<string>

#include<stdio.h>

using namespace std;

int main()

{

int hour,minute;

cout<<"Enter time in Hour Format:";

cin>>hour;

cout<<"Enter time in Minute Format:";

cin>>minute;

string h[] = {"One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten",

"Eleven", "Twelve"};

string m[] = { "one", "two", "three", "four", "five", "six", "seven",

"eight", "nine", "ten", "eleven", "twelve",

"thirteen", "fourteen" };

if(hour!=0&&minute==0){

cout<<h[hour-1]<< " o' clock";

}

else

if(hour!=0&&minute==10)

{

cout<<m[minute-1]<< " minutes "<<"past "<<h[hour-1];

}

else if(hour!=0&&minute==30)

{

cout<<"half past "<<h[hour-1];

}

else if(hour!=0&&minute==45){

cout<<"quarter to "<<h[hour];

}

else if(hour!=0&&minute>45){

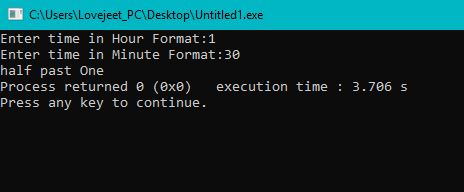
int outer;

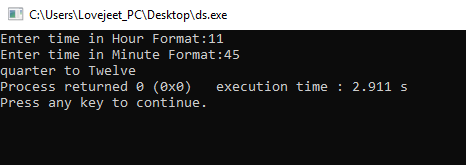
outer=60-minute;

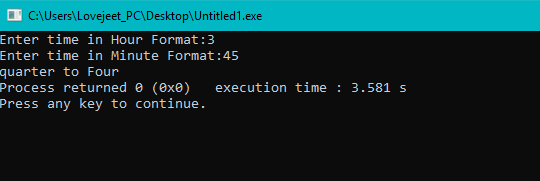
cout<<m[outer-1]<<" minutes to "<<h[hour];

}

}







Question-6 Sorting Algorithm (Selection)

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

int n,i,j,counter=0;

int arr[] = {20,15,214,152,1,451,485};

n = sizeof(arr)/sizeof(arr[0]);

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(arr[i]>arr[j])

{

counter=arr[i];

arr[i]=arr[j];

arr[j]=counter;

}

}

}

cout<<"Sorted List of Above Array is:"<<endl;

for(i=0;i<n;i++)

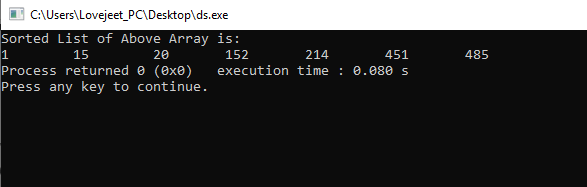
{

cout<<arr[i];

cout<<setw(10);

}

}



Question-7 Array Rotation

#include<iostream>

using namespace std;

void leftRotatebyOne(int arr[], int n)

{

int temp = arr[0], i;

for (i = 0; i < n - 1; i++)

arr[i] = arr[i + 1];

arr[i] = temp;

}

void leftRotate(int arr[], int d, int n)

{

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

leftRotatebyOne(arr, n);

}

void printArray(int arr[], int n)

{

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

cout << arr[i] << " ";

}

int main()

{

int arr[] = { 1, 3, 5, 7, 9, 11, 13 };

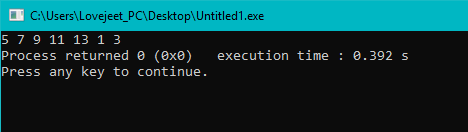
int n = sizeof(arr) / sizeof(arr[0]);

leftRotate(arr, 2, n);

printArray(arr, n);

return 0;

}



Question-8 Splitting of Array

#include <bits/stdc++.h>

using namespace std;

void splitArr(int arr[], int n, int k)

{

for (int i = 0; i < k; i++) {

int x = arr[0];

for (int j = 0; j < n - 1; ++j)

arr[j] = arr[j + 1];

arr[n - 1] = x;

}

}

int main()

{

int arr[] = { 12, 10, 5, 6, 52, 36 };

int n = sizeof(arr) / sizeof(arr[0]);

int position = 2;

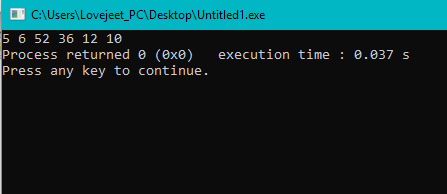
splitArr(arr, 6, position);

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

printf("%d ", arr[i]);

return 0;

}



Question-9 Reverse Array using Pointers

#include <iostream>

using namespace std;

void swap(int\* a, int\* b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void reverse(int array[], int array\_size)

{

int \*pointer1 = array,

\*pointer2 = array + array\_size - 1;

while (pointer1 < pointer2) {

swap(pointer1, pointer2);

pointer1++;

pointer2--;

}

}

void print(int\* array, int array\_size)

{

int \*length = array + array\_size,

\*position = array;

cout << "Array = ";

for (position = array; position < length; position++)

cout << \*position << " ";

}

int main()

{

int array[] = { 1,10,100,1000,2000,3000};

cout << "Original ";

print(array, 6);

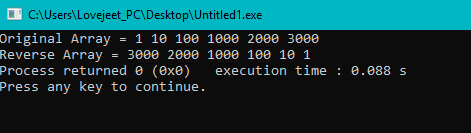
cout <<endl<< "Reverse ";

reverse(array, 6);

print(array, 6);

return 0;

}



Question-10 Maximum consecutive one’s (or zeros) in a binary circular array

#include <bits/stdc++.h>

using namespace std;

int getMaxLength(bool arr[], int n)

{

int count = 0;

int result = 0;

for (int i = 0; i < 2 \* n; i++) {

if (arr[i % n] == 0) {

count = 0;

if (i >= n)

break;

}

else {

count++;

result = max(result, count);

}

}

return result;

}

int main()

{

bool arr[] = { 1, 1, 0, 0, 1, 0, 1, 0,

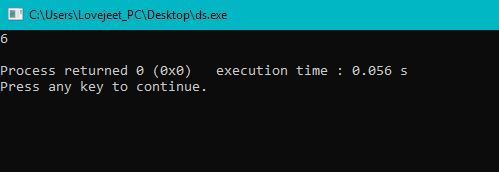
1, 1, 1, 1 };

int n = sizeof(arr) / sizeof(arr[0]);

cout << getMaxLength(arr, n) << endl;

return 0;

}



Question-11 Pointer(c)

#include<iostream>

using namespace std;

main(){

int stud[5][2]={

{1234,56},

{1212,33},

{1434,80},

{1312,78},

{1203,75}

};

int i,j;

for(i=0;i<5;i++){

cout<<"\n";

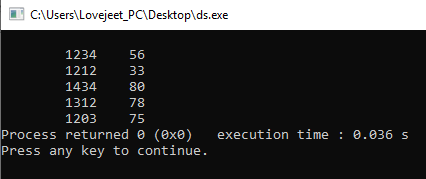
for(j=0;j<=1;j++){

cout<<"\t"<<\*(\*(stud+i)+j);

}

}

}



Question-12 Simple Inheritance

#include <iostream>

using namespace std;

class A

{

private:

int a;

protected:

int x;

public:

void setVal(int v)

{

x=v;

}

};

class B:private A

{

public:

void printVal(void)

{

setVal(10);

cout << "value of x: " << x << endl;

}

};

int main()

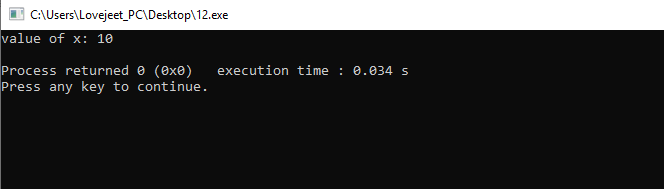
{

B objB;

objB.printVal();

return 0;

}



Question-13 Replace array elements by sum of next two consecutive elements

#include <bits/stdc++.h>

using namespace std;

void printArr(int arr[], int n)

{

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

cout << arr[i] << " ";

}

void updateArr(int arr[], int n)

{

if (n < 3)

return;

int first = arr[0];

int second = arr[1];

for (int i = 0; i < n - 2; i++)

arr[i] = arr[i + 1] + arr[i + 2];

arr[n - 2] = arr[n - 1] + first;

arr[n - 1] = first + second;

printArr(arr, n);

}

int main()

{

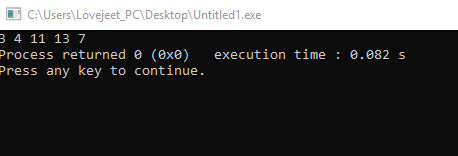
int arr[] = {5, 2, 1, 3, 8};

int n = sizeof(arr) / sizeof(arr[0]);

updateArr(arr, n);

return 0;

}



Question-14 Hierarchy of Class

#include <iostream>

using namespace std;

class BaseClass

{

int i;

public:

void setInt(int n);

int getInt();

};

class DerivedClass : public BaseClass

{

int j;

public:

void setJ(int n);

int mul();

};

void BaseClass::setInt(int n)

{

i = n;

}

int BaseClass::getInt()

{

return i;

}

void DerivedClass::setJ(int n)

{

j = n;

}

int DerivedClass::mul()

{

return j \* getInt();

}

int main()

{

DerivedClass ob;

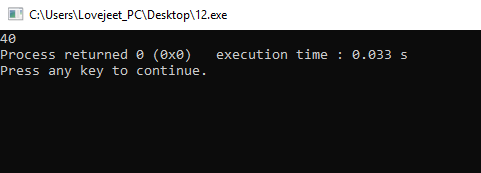
ob.setInt(10);

ob.setJ(4);

cout << ob.mul();

return 0;

}



Question-15 Count Different Element in Array

#include <iostream>

using namespace std;

int countDistinct(int arr[], int n)

{

int res = 1;

for (int i = 1; i < n; i++) {

int j = 0;

for (j = 0; j < i; j++)

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

break;

if (i == j)

res++;

}

return res;

}

int main()

{

int arr[] = { 12, 10, 9, 45, 2, 10, 10, 45 };

int n = sizeof(arr) / sizeof(arr[0]);

cout << countDistinct(arr, n);

return 0;

}

