**15B17CI371 – Data Structures Lab**

**ODD 2024**

**Week 1-LAB B**

**Practice Lab**

1.Write a C/C++ program to find the average of n numbers using arrays.

#include<iostream>

using namespace std;

int main()

{

int n;

cout<<"Enter the no. of elements in the array:";

cin>>n;

int arr[n];

int sum=0;

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

{

cin>>arr[i];

sum=sum+arr[i];

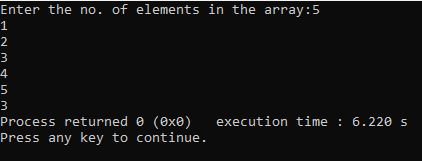
}

float avg;

avg=sum/n;

cout<<avg;

}



2. Write a C/C++ program to find the frequency of each element in an array.

Example:

Input: array = {1, 2, 3, 5, 2, 9, 7, 3, 5}

Output:

1 occurs 1 times

2 occurs 2 times

3 occurs 2 times

5 occurs 2 times

7 occurs 1 times

9 occurs 1 times

#include <iostream>

using namespace std;

int main() {

int array[] = {1, 2, 3, 5, 2, 9, 7, 3, 5};

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

int unique[n];

int freq[n];

int c = 0;

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

freq[i] = 0;

}

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

int j;

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

if (array[i] == unique[j]) {

freq[j]++;

break;

}

}

if (j == c) {

unique[c] = array[i];

freq[c] = 1;

c++;

}

}

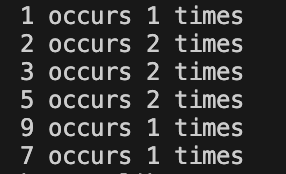
for (int i = 0; i < c; ++i) {

cout << unique[i] << " occurs " << freq[i] << " times" << endl;

}

return 0;

}



3.Given an array, write a program in C/C++to left rotate the elements of the array by one.

Example: Array Elements before rotating: 1 2 3 4 5 6 7

Array Elements after rotating: 2 3 4 5 6 7 1

#include<iostream>

using namespace std;

int main()

{

int n;

cout<<"Enter the no. of elements in the array:";

cin>>n;

int arr[n];

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

{

cin>>arr[i];

}

cout<<"Array elements before rotating:"<<endl;

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

{

cout<<arr[i];

cout<<" ";

}

int temp;

temp=arr[0];

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

{

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

}

arr[n-1]=temp;

cout<<endl;

cout<<"Array elements after rotating:"<<endl;

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

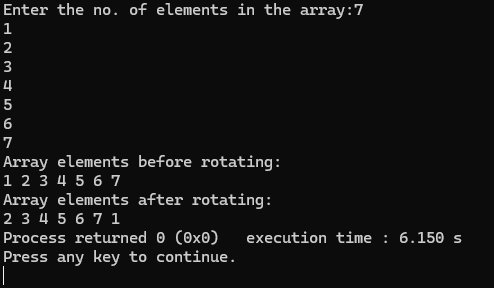
{

cout<<arr[i];

cout<<" ";

}

}



4.Write a C/C++ program to find the second smallest element in a one-dimensional array.

Example: Input:

Array size: 4

Elements: 32 54 -6 -15

Output: -6

#include<iostream>

using namespace std;

int main()

{

int n;

cout<<"Array size:";

cin>>n;

int arr[n];

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

{

cin>>arr[i];

}

int temp,temp1;

temp=arr[0];

temp1=arr[1];

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

{

if(arr[i]<temp)

{ temp1=temp;

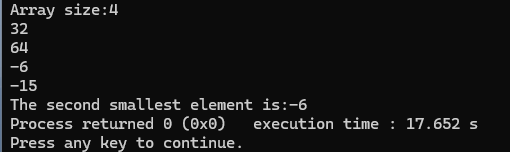
temp=arr[i];

}

}

cout<<"The second smallest element is:"<<temp1;

}



5.A dynamically created array stores following integer elements (odd and even integers). It

is desired to print/display the elements of this array in such manner that it first prints all

the even elements then it prints all the odd elements.

Example:

Input: 2 8 3 6 7 9 5 4

Output: 2 8 6 4 3 7 9 5

#include<iostream>

using namespace std;

int main()

{

int a;

cout<<"Enter number of inputs: ";

cin>>a;

int \*arr=new int[a];

cout<<"Enter the numbers:\n";

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

cin>>arr[i];

}

cout<<"Input: ";

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

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

}

cout<<endl<<"Output: ";

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

if(arr[i]%2==0)

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

}

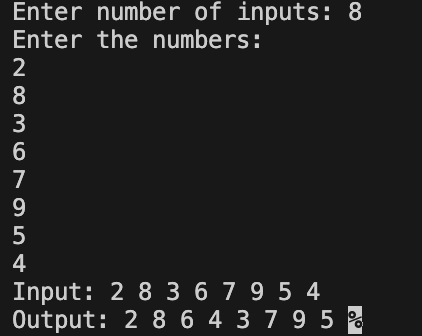
for(int i=0;i<a;i++){

if(arr[i]%2!=0)

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

}

}



6. Write a program without STL to create the dynamic array of user inputted length (n),

assign values at different indices of the array, and as presented in above example,

display the elements of this array.

(Note: don’t enter the elements manually, rather use following statement in loop to

randomly assign elements (in range between 0 and 99) in the array:

A[i] = rand()%100, where A is an array).

#include<iostream>

using namespace std;

int main()

{

int a;

cout<<"Enter number of inputs: ";

cin>>a;

int \*A= new int[a];

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

{

A[i]=rand()%100;

}

cout<<"The randomly generated array of length "<<a<<":"<<endl;

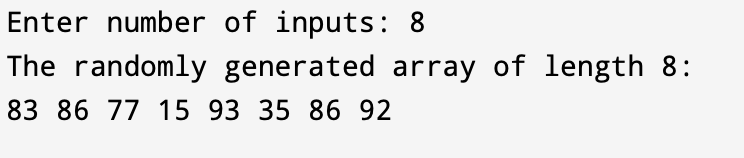
for(int i=0;i<a;i++){

cout<<A[i]<<" ";

}

return 0;

}

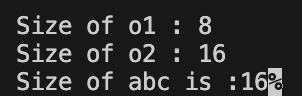


7.Considering that an integer variable, a float variable, a double variable, a character

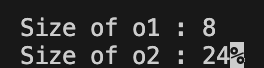
variable, and a pointer variable need 4, 4, 8, 1, and 8 bytes memory space respectively,

what will be the output of following C++ programs.

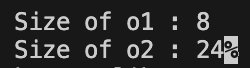
a)



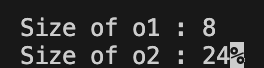
b)



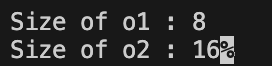
c)



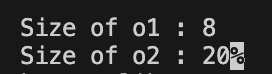
d)



e)



f)



8. Analyze the correctness and output of following programs

(a)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

float \*a;

a = (float \*)malloc(sizeof(int));

a[0] = 4.5;

cout<<a[0];

return 0;

}

Ans : the code is correct and the output will be 4.5

(b)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a;

a = (int \*)malloc(sizeof(float));

a[0] = 5;

cout<<a[0];

return 0;

}

Ans : the code is correct and the output will be 5

(c)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a, \*b;

a = (int \*)malloc(sizeof(int));

b = (int \*)malloc(5\*sizeof(int));

cout<<sizeof(a)<< sizeof(b);

return 0;

}

Ans : the code is correct and the output will be 8 8.

(d)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a;

a[0] = (int \*)malloc(sizeof(int));

a[0] = 5;

cout<<a[0];

return 0;

}

Ans : here the a is an pointer to an integer but it is uninitialized henceforth no memory can be allocated , hence accessing a[0] will lead to the termination of the code while running it.

(e)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a[5];

a[0] = (int \*)malloc(sizeof(int));

a[0][0] = 5;

cout<<a[0][0];

return 0;

}

Ans : the code is correct and the output will be 5

(f)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

struct node{int a[10];};

struct node \*n;

n = (struct node \*)malloc(sizeof(struct

node)); cout<<sizeof(n);

return 0;

}

Ans : int this code the part size of(struct node) calculates the size of ‘n’ rather than calculating the size of memory allocated for the struct node.

(g)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a[5];

a[0] = (int

\*)malloc(2\*sizeof(int)); a[0][1]

= 5;

cout<<a[0][1];

return 0;

}

Ans : the code is correct and the output will be 5

(h)

#include <iostream>

#include <malloc.h>

using namespace std;

int main() {

int \*a = (int \*)malloc(5\*sizeof(int));

a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4;

a[4] = 5; delete(a);

cout<<a[0]<<a[1]<<a[2]<<a[3]<<a[4];

return 0;

}

Ans : this program will print garbage values due to the use of delete() instead of free(a).