# Assignment ( week 0 ) [MUKUL AGGARWAL 2401030239](mailto:2401030239@mail.jiit.ac.in)

## Question 1)

#include <iostream>

#include<vector>

#include <unordered\_map>

using namespace std;

int main(){

vector<int> arr = {1, 2, 3, 5, 2, 9, 7, 3, 5};

unordered\_map<int, int> oho;

for(int num : arr){

oho[num]++;

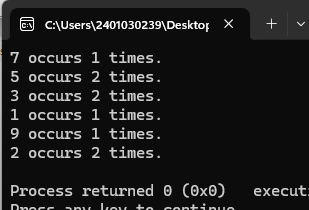
}

for(auto it : oho){

cout<<it.first<<" occurs "<<it.second<<" times.\n";

}

}



## Question 2)

#include <iostream>

#include<vector>

#include <unordered\_map>

using namespace std;

int main(){

vector<int> arr = {1, 2, 3, 5, 2, 9, 7, 3, 5};

int n = arr.size();

if(n==1){

cout<<"no change";

return 0;

}

cout<<"elements before rotation : ";

for(int num : arr){

cout<<num;

}

cout<<"\nelements after rotation : ";

int a = arr[0];

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

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

}

arr[n-1]=a;

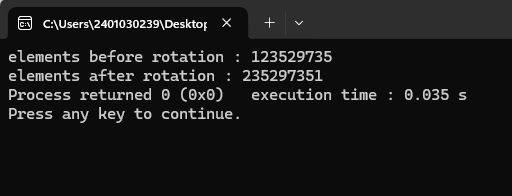
for(int num : arr){

cout<<num;

}

return 0;

}



## Question 3)

#include <iostream>

#include<vector>

#include <unordered\_map>

using namespace std;

int main(){

vector<int> arr = {32, 54, -6, -15};

int n = arr.size();

int small, big;

if(arr[0]<arr[1]){

small = arr[0]; big = arr[1];

}else{small = arr[1]; big = arr[0];}

for(int num : arr){

if(num<big && num>small){

big = num;

}else if(num<small){

big = small;

small = num;

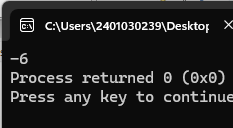
}

}

cout<<big;

return 0;

}



## Question 4)

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter number of elements: ";

cin >> n;

int\* arr = new int[n];

cout << "Enter " << n << " elements (odd and even integers): ";

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

cin >> arr[i];

}

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

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

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

}

}

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

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

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

}

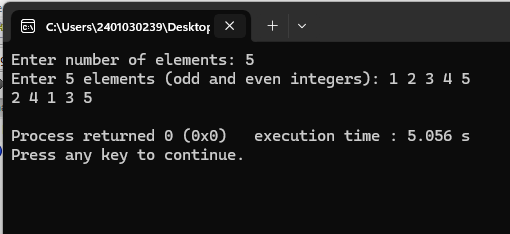
}

cout << endl;

delete[] arr;

return 0;

}



## Question 5)

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n;

cin >> n;

vector<int> socks(2\*n);

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

cin >> socks[i];

}

vector<bool> onTable(n+1, false);

int currentCount = 0;

int maxCount = 0;

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

int sock = socks[i];

if (!onTable[sock]) {

onTable[sock] = true;

currentCount++;

if (currentCount > maxCount) {

maxCount = currentCount;

}

} else {

onTable[sock] = false;

currentCount--;

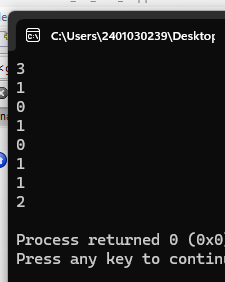
}

}

cout << maxCount << endl;

return 0;

}



## Question 6)