1. Two Sum II - Input array is sorted

Given a sorted array of integers, return the indices of the two numbers such that they add up to a specific target.

Code –

#include<bits/stdc++.h>

using namespace std;

vector<int> twosum(vector<int> arr,int k){

int left=0,right=arr.size()-1;

while(left<right){

if(arr[left]+arr[right]==k){

return {left,right};

}

if(arr[left]+arr[right]<k){

right--;

}

if(arr[left]+arr[right]>k){

left++;

}

}

return {};

}

int main(){

int n,k;

cin>>k;

cin>>n;

vector<int> arr(n);

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

cin>>arr[i];

}

sort(arr.begin(),arr.end());

cout<<”Sorted Array: “;

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

cout<<arr[i]<<” “;

}

vector<int> res = twosum(arr,k);

for(int i=0;i<res.size();i++){

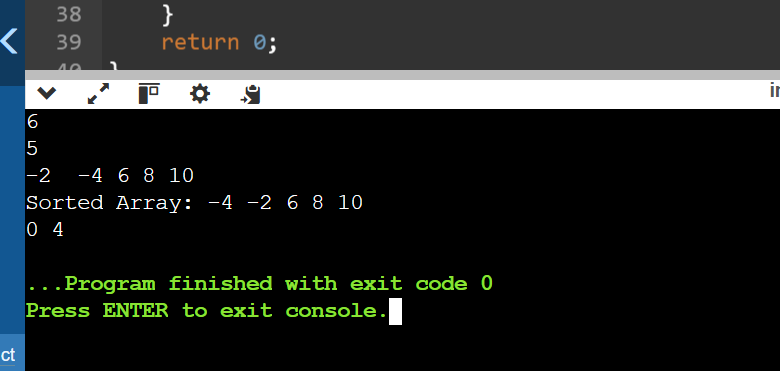
cout<<res[i]<<" ";

}

return 0;

}

Output –



2. Subarray Sum Equals K

Given an array of integers and a target sum k, return the total number of continuous subarrays whose sum equals to k.

Code –

#include<bits/stdc++.h>

using namespace std;

int countsum(vector<int> arr, int k){

unordered\_map<int,int> count\_map;

int sum=0;

int count=0;

int n=arr.size();

count\_map[0]=1;

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

sum+=arr[i];

if(count\_map.find(sum-k)!=count\_map.end()){

count+=count\_map[sum-k];

}

count\_map[sum]++;

}

return count;

}

int main(){

int n,k;

cin>>n;

vector<int> arr(n);

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

cin>>arr[i];

}

cout<<"Target: ";

cin>>k;

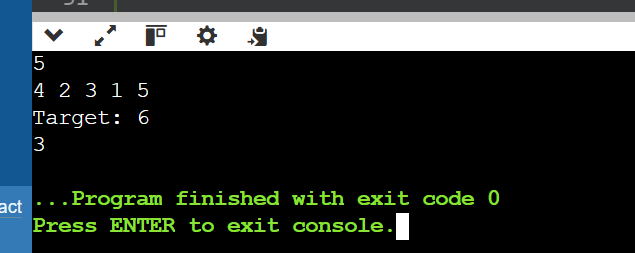
int res=countsum(arr,k);

cout<<res;

return 0;

}

Output –



7. Strings

Longest Substring Without Repeating Characters

Given a string, find the length of the longest substring without repeating characters.

Code –

#include<bits/stdc++.h>

using namespace std;

int larglen(string s){

int n=s.length();

int maxlen=0;

unordered\_set<char> cset;

int left=0;

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

if(cset.count(s[right])==0){

cset.insert(s[right]);

maxlen=max(maxlen,right-left);

}else{

while(cset.count(s[right])){

cset.erase(s[left]);

left++;

}

cset.insert(s[right]);

}

}

return maxlen;

}

int main(){

string s;

cin>>s;

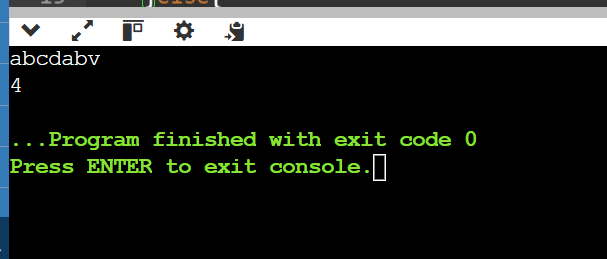
int out=larglen(s);

cout<<out;

return 0;

}

Output –



9. Rearrange a no to find min possible no in o(n) and constant space.

Code –

#include<bits/stdc++.h>

using namespace std;

string nextgreter(string s){

int n=s.length();

string ans="";

int arr[10]={0};

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

arr[s[i]-48]++;

}

for(int i=0;i<10;i++){

for(int j=0;j<arr[i];j++)ans=ans+to\_string(i);

}

return ans;

}

int main(){

string s;

cin>>s;

string res=nextgreter(s);

cout<<res;

return 0;

}

Output –

