Questions for today.

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.

#include <bits/stdc++.h>

using namespace std;

vector<int>twosum(int n,vector<int>&arr,int target)

{

    int i=0;

    int j=n-1;

    vector<int>result(2,-1);

    while(i<j)

    {

        if(arr[i]+arr[j]==target){

            result[0]=i;

            result[1]=j;

            return result;

        }

         if(arr[i]+arr[j]>target)

        {

             j--;

        }

        else i++;

    }

    return result;

}

int main()

{

   int n=6;

   vector<int>arr={1,2,3,4,5,11};

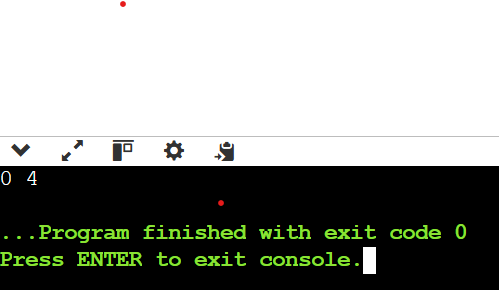
   int target=6;

   vector<int>ans=twosum(n,arr,target);

   if(ans[0]==-1) cout<<-1;

  else cout<<ans[0]<<" "<<ans[1];

}

  
  
  
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.

#include <bits/stdc++.h>

using namespace std;

int subarraysum(vector<int>arr,int sum)

{

    int n=arr.size();

    map<int,int>mpp;

    mpp[0]=1;

    int s=0;

    int count=0;

    for(int i=0;i<n;i++)

    {

        s=s+arr[i];

        if(mpp.find(s-sum)!=mpp.end())

        {

            count+=mpp[s-sum];

        }

        if(mpp.find(s)==mpp.end()){

            mpp[s]=1;

        }

        else mpp[s]++;

    }

    return count;

}

int main()

{

    int n=4;

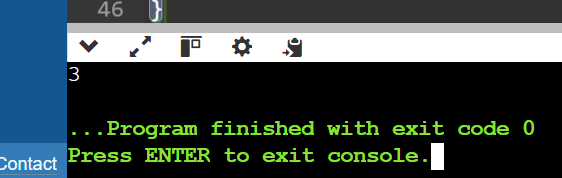
    vector<int>arr={0,1,2,3,6};

    int target=6;

    cout<<subarraysum(arr,target);

    return 0;

}

  
  
  
3. Next Greater Element  
Given a circular array, find the next greater number for every element.  
231 -> 312  
  
#include <bits/stdc++.h>

using namespace std;

int main()

{  int n=5;

    vector<int>a={2,3,1,4,2};

    next\_permutation(a.begin(),a.end());

    for(int i=0;i<a.size();i++)

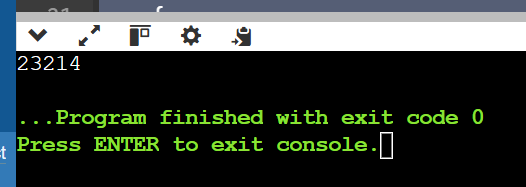
    {

        cout<<a[i];

    }

    return 0;

}

  
  
  
7. Strings  
Longest Substring Without Repeating Characters  
Given a string, find the length of the longest substring without repeating characters.  
  
#include <bits/stdc++.h>

using namespace std;

    int LongestSubstring(string s) {

      vector < int > mpp(256, -1);

      int left = 0, right = 0;

      int n = s.size();

      int len = 0;

      while (right < n) {

        if (mpp[s[right]] != -1)

          left = max(mpp[s[right]] + 1, left);

        mpp[s[right]] = right;

        len = max(len, right - left + 1);

        right++;

      }

      return len;

    }

int main() {

  string str = "sanskar";

  cout <<LongestSubstring(str);

  return 0;

}

