**Name: Sakshi Kumari**

**Email:** [bhardwajsakshi027@gmail.com](mailto:bhardwajsakshi027@gmail.com)

**Phone no: 8102014145**

**DSA BOOTCAMP ASSIGNMENT**

Q1. Write a program to Swap to two numbers.

Code:

#include <iostream>

using namespace std;

int main()

{

int a = 5, b = 10, temp;

cout << "Before swapping." << endl;

cout << "a = " << a << ", b = " << b << endl;

temp = a;

a = b;

b = temp;

cout << "\nAfter swapping." << endl;

cout << "a = " << a << ", b = " << b << endl;

return 0;

}

Q2. Write a program to find the largest number among three numbers entered by the user.

Code:

#include<iostream>

using namespace std;

int main()

{

int numOne, numTwo, numThree, larg;

cout<<"Enter the Three Numbers: ";

cin>>numOne>>numTwo>>numThree;

if(numOne>numTwo)

{

if(numTwo>numThree)

larg = numOne;

else

{

if(numThree>numOne)

larg = numThree;

else

larg = numOne;

}

}

else

{

if(numTwo>numThree)

larg = numTwo;

else

larg = numThree;

}

cout<<"\nLargest Number = "<<larg;

cout<<endl;

return 0;

}

Q3. Write a program to check whether a year entered by a user is Leap year or not.

Code:

#include<iostream>

using namespace std;

int main()

{

int yr;

cout<<"Enter the Year: ";

cin>>yr;

if((yr%4==0) && (yr%100!=0))

cout<<"\nIt is a Leap Year";

else if(yr%400==0)

cout<<"\nIt is a Leap Year";

else

cout<<"\nIt is not a Leap Year";

cout<<endl;

return 0;

}

Q4. Write a program to display Fibonacci Series upto nth term. (Using loops)

Code:

// C++ program to print

// first n Fibonacci numbers

#include <bits/stdc++.h>

using namespace std;

// Function to print

// first n Fibonacci Numbers

void printFibonacciNumbers(int n)

{

int f1 = 0, f2 = 1, i;

if (n < 1)

return;

cout << f1 << " ";

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

cout << f2 << " ";

int next = f1 + f2;

f1 = f2;

f2 = next;

}

}

// Driver Code

int main()

{

printFibonacciNumbers(7);

return 0;

}

Q5. Write a program to check whether a number is Prime or Not.

Code:

#include <iostream>

using namespace std;

int main()

{

int num1, ctr = 0;

cout << "\n\n Check whether a number is prime or not:\n";

cout << "--------------------------------------------\n";

cout << " Input a number to check prime or not: ";

cin>> num1;

for (int a = 1; a <= num1; a++)

{

if (num1 % a == 0)

{

ctr++;

}

}

if (ctr == 2)

{

cout << " The entered number is a prime number. \n";

}

else {

cout << " The number you entered is not a prime number. \n";

}

}

Q6. Print this pattern using loops

For n=5

    \*

  \* \*

  \* \* \*

\* \* \* \*

\* \* \* \* \*

Code:

// C++ code to demonstrate star pattern

#include <iostream>

using namespace std;

// Function to demonstrate printing pattern

void triangle(int n)

{

// number of spaces

int k = 2 \* n - 2;

// Outer loop to handle number of rows

// n in this case

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

// Inner loop to handle number spaces

// values changing acc. to requirement

for (int j = 0; j < k; j++)

cout << " ";

// Decrementing k after each loop

k = k - 1;

// Inner loop to handle number of columns

// values changing acc. to outer loop

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

// Printing stars

cout << "\* ";

}

// Ending line after each row

cout << endl;

}

}

// Driver Code

int main()

{

int n = 5;

// Function Call

triangle(n);

return 0;

}

Q7.Write a program that takes n elements from the user and displays the second largest element of an array.

Code:

#include <iostream>

using namespace std;

int main(){

int n, num[50], largest, second;

cout<<"Enter number of elements: ";

cin>>n;

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

cout<<"Enter Array Element"<<(i+1)<<": ";

cin>>num[i];

}

/\* Here we are comparing first two elements of the

\* array, and storing the largest one in the variable

\* "largest" and the other one to "second" variable.

\*/

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

largest = num[1];

second = num[0];

}

else{

largest = num[0];

second = num[1];

}

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

/\* If the current array element is greater than largest

\* then the largest is copied to "second" and the element

\* is copied to the "largest" variable.

\*/

if (num[i] > largest) {

second = largest;

largest = num[i];

}

/\* If current array element is less than largest but greater

\* then second largest ("second" variable) then copy the

\* element to "second"

\*/

else if (num[i] > second && num[i] != largest) {

second = num[i];

}

}

cout<<"Second Largest Element in array is: "<<second;

return 0;

}

Q8. <https://www.hackerrank.com/challenges/array-left-rotation/problem>

Code:

// C++ program to rotate an array by

// d elements

#include <bits/stdc++.h>

using namespace std;

/\*Function to get gcd of a and b\*/

int gcd(int a, int b)

{

if (b == 0)

return a;

else

return gcd(b, a % b);

}

/\*Function to left rotate arr[] of siz n by d\*/

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

{

/\* To handle if d >= n \*/

d = d % n;

int g\_c\_d = gcd(d, n);

for (int i = 0; i < g\_c\_d; i++) {

/\* move i-th values of blocks \*/

int temp = arr[i];

int j = i;

while (1) {

int k = j + d;

if (k >= n)

k = k - n;

if (k == i)

break;

arr[j] = arr[k];

j = k;

}

arr[j] = temp;

}

}

// Function to print an array

void printArray(int arr[], int size)

{

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

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

}

/\* Driver program to test above functions \*/

int main()

{

int arr[] = { 1,2,3,4,5 };

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

// Function calling

leftRotate(arr, 2, n);

printArray(arr, n);

return 0;

}

Q9. <https://www.hackerrank.com/challenges/grading/problem>

Code:

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

/\*

\* Complete the 'gradingStudents' function below.

\*

\* The function is expected to return an INTEGER\_ARRAY.

\* The function accepts INTEGER\_ARRAY grades as parameter.

\*/

vector<int> gradingStudents(vector<int> grades) {

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

{

if((grades[i])<38)

break;

if((grades[i]%5) >= 3)

grades[i]=grades[i]+ (5-(grades[i]%5));

}

return grades;

}

int main()

{

ofstream fout(getenv("OUTPUT\_PATH"));

string grades\_count\_temp;

getline(cin, grades\_count\_temp);

int grades\_count = stoi(ltrim(rtrim(grades\_count\_temp)));

vector<int> grades(grades\_count);

for (int i = 0; i < grades\_count; i++) {

string grades\_item\_temp;

getline(cin, grades\_item\_temp);

int grades\_item = stoi(ltrim(rtrim(grades\_item\_temp)));

grades[i] = grades\_item;

}

vector<int> result = gradingStudents(grades);

for (size\_t i = 0; i < result.size(); i++) {

fout << result[i];

if (i != result.size() - 1) {

fout << "\n";

}

}

fout << "\n";

fout.close();

return 0;

}

string ltrim(const string &str) {

string s(str);

s.erase(

s.begin(),

find\_if(s.begin(), s.end(), not1(ptr\_fun<int, int>(isspace)))

);

return s;

}

string rtrim(const string &str) {

string s(str);

s.erase(

find\_if(s.rbegin(), s.rend(), not1(ptr\_fun<int, int>(isspace))).base(),

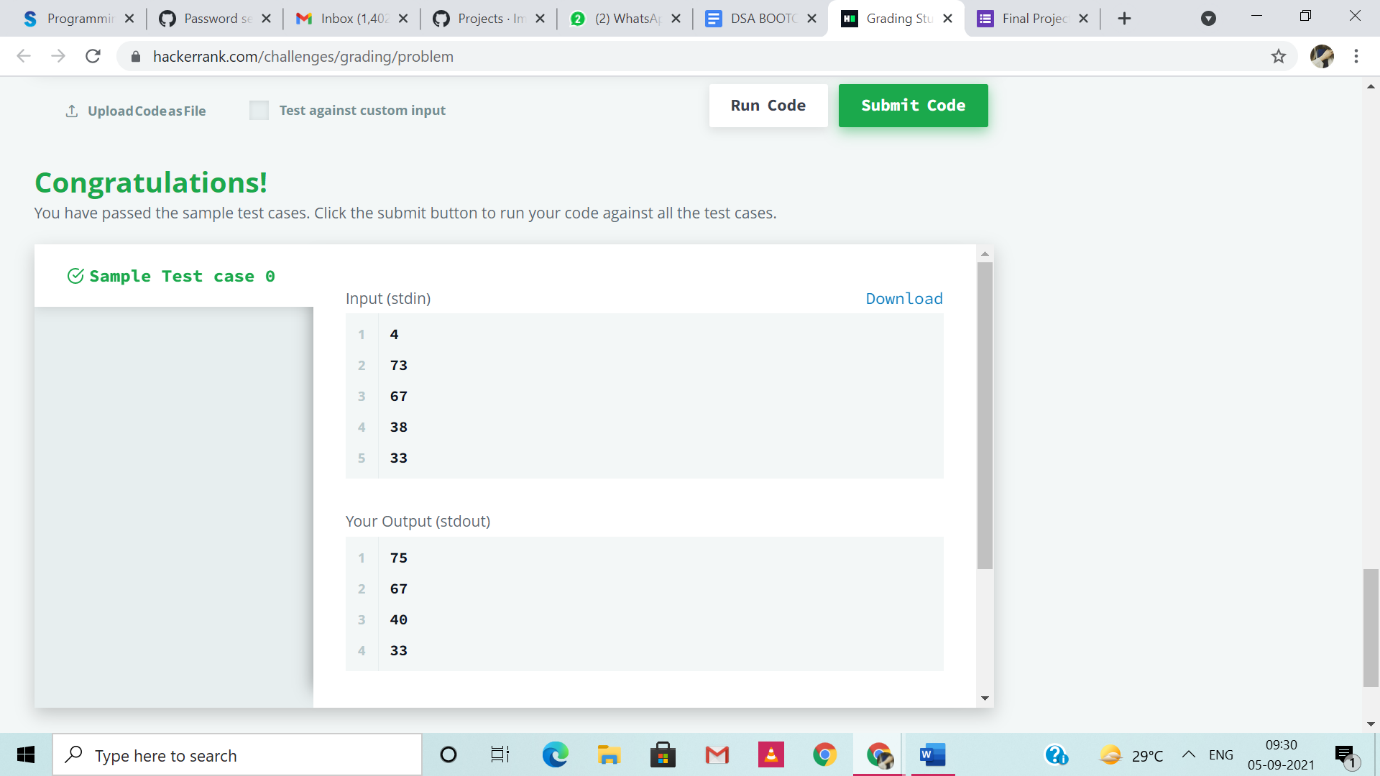
s.end()

);

return s;

}

Output:



Q10.  <https://www.hackerrank.com/challenges/camelcase/problem>

Code:

#include <bits/stdc++.h>

using namespace std;

/\*

 \* Complete the 'camelcase' function below.

 \*

 \* The function is expected to return an INTEGER.

 \* The function accepts STRING s as parameter.

 \*/

int camelcase(string s) {

int count = 0;

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

        if(s[i] >= 65 && s[i] <= 90) {

            count++;

        }

    }

    return count+1;

}

int main()

{

    ofstream fout(getenv("OUTPUT\_PATH"));

    string s;

    getline(cin, s);

    int result = camelcase(s);

    fout << result << "\n";

    fout.close();

    return 0;

}

Output:

