**DSA BOOTCAMP ASSIGNMENT**

Q1. Write a program to Swap to two numbers.

Code:

#include <iostream>

using namespace std;

int main()

{

int a=0,b=0;

cout<<"Enter Two Numbers :";

cin>>a>>b;

a=a+b;

b=a-b;

a=a-b;

cout<<"The numbers after swap are \n";

cout<<"A = "<<a<<"\n";

cout<<"B = "<<b<<"\n";

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 a,b,c;

cout<<"Enter Three Numbers :";

cin>>a>>b>>c;

if(a>b && a>c)

cout<<a<<" Is Largest";

else if(b>a && b>c)

cout<<b<<" Is Largest";

else

cout<<c<<" Is Largest";

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 year;

cout<<"Enter a Year :";

cin>>year;

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0)

cout << year << " is a leap year";

else

cout << year << " is not a leap year";

} else

cout << year << " is a leap year";

} else

cout << year << " is not a leap year";

return 0;

}

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

CODE:

#include <iostream>

using namespace std;

int main()

{

int i=0,n=0,n1=0,n2=1,n3=0;

cout<<"Enter the number of elements in the Fibonacci Series :";

cin>>n;

cout<<"\n Fibanacci Series :"<<n1<<" "<<n2;

for(i=3; i<=n; i++)

{

n3=n1+n2;

cout<<" "<<n3;

n1=n2;

n2=n3;

}

}

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

CODE:

#include <iostream>

using namespace std;

int main()

{ int i,a,count=0;

cout<<"Enter the number :";

cin>>a;

for(i=2; i<=a/2; i++){

if(a%i == 0){

count+=1; }

}

if(count==0)

cout<<a<<" is a Prime Number";

else

cout<<a<<" is Not Prime Number";

}

Q6. Print Pyramid pattern using loops,

Code:

#include<iostream>

using namespace std;

int main()

{

int i, s, j, n;

cout<<"Enter the Length of the Desired Pattern :";

cin>>n;

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

{

for(s=n; s>i; s--)

cout<<" ";

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

cout<<"\* ";

cout<<"\n";

}

cout<<"\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,i,j,x;

cout<<"Enter the Size of Array :";

cin>>n;

int arr[n];

cout<<"Enter the elements of the Array :";

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

cin>>arr[i];

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

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

if(arr[i] < arr[j]){

x = arr[i];

arr[i] = arr[j];

arr[j] = x;

}

}

}

cout<<"The Second Largest Number in the Array is :"<<arr[1];

return 0;

}

Q8. [Left Rotation](https://www.hackerrank.com/challenges/array-left-rotation/problem).

Code:

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

vector<string> split(const string &);

/\*

 \* Complete the 'rotateLeft' function below.

 \*

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

 \* The function accepts following parameters:

 \*  1. INTEGER d

 \*  2. INTEGER\_ARRAY arr

 \*/

vector<int> rotateLeft(int d, vector<int> arr) {

  for(int i = 0; i < d; ++i) {

        arr.push\_back(arr.front());

        arr.erase(arr.begin());

    }

    return arr;

}

int main()

{

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

    string first\_multiple\_input\_temp;

    getline(cin, first\_multiple\_input\_temp);

    vector<string> first\_multiple\_input = split(rtrim(first\_multiple\_input\_temp));

    int n = stoi(first\_multiple\_input[0]);

    int d = stoi(first\_multiple\_input[1]);

    string arr\_temp\_temp;

    getline(cin, arr\_temp\_temp);

    vector<string> arr\_temp = split(rtrim(arr\_temp\_temp));

    vector<int> arr(n);

    for (int i = 0; i < n; i++) {

        int arr\_item = stoi(arr\_temp[i]);

        arr[i] = arr\_item;

    }

    vector<int> result = rotateLeft(d, arr);

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

        fout << result[i];

        if (i != result.size() - 1) {

            fout << " ";

        }

    }

    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;

}

vector<string> split(const string &str) {

    vector<string> tokens;

    string::size\_type start = 0;

    string::size\_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {

        tokens.push\_back(str.substr(start, end - start));

        start = end + 1;

    }

    tokens.push\_back(str.substr(start));

    return tokens;

}

#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) {

 vector<int> ans;

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

        int temp = grades[i];

        if(grades[i] >= 38){

            if(grades[i]%5 >= 3){

                temp += 5 - grades[i]%5;

            }

        }

        ans.push\_back(temp);

    }

    return ans;

}

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);

Q10. [CamelCase](https://www.hackerrank.com/challenges/camelcase/problem)

Code:

    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;

}

#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 = 1;

    for (int i = 1; i < s.length() - 1; i++) {

        if (isupper(s[i]))

            count++;

    }

    return count;

}

int main()

{

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

    string s;

    getline(cin, s);

    int result = camelcase(s);

    fout << result << "\n";

    fout.close();

    return 0;

}