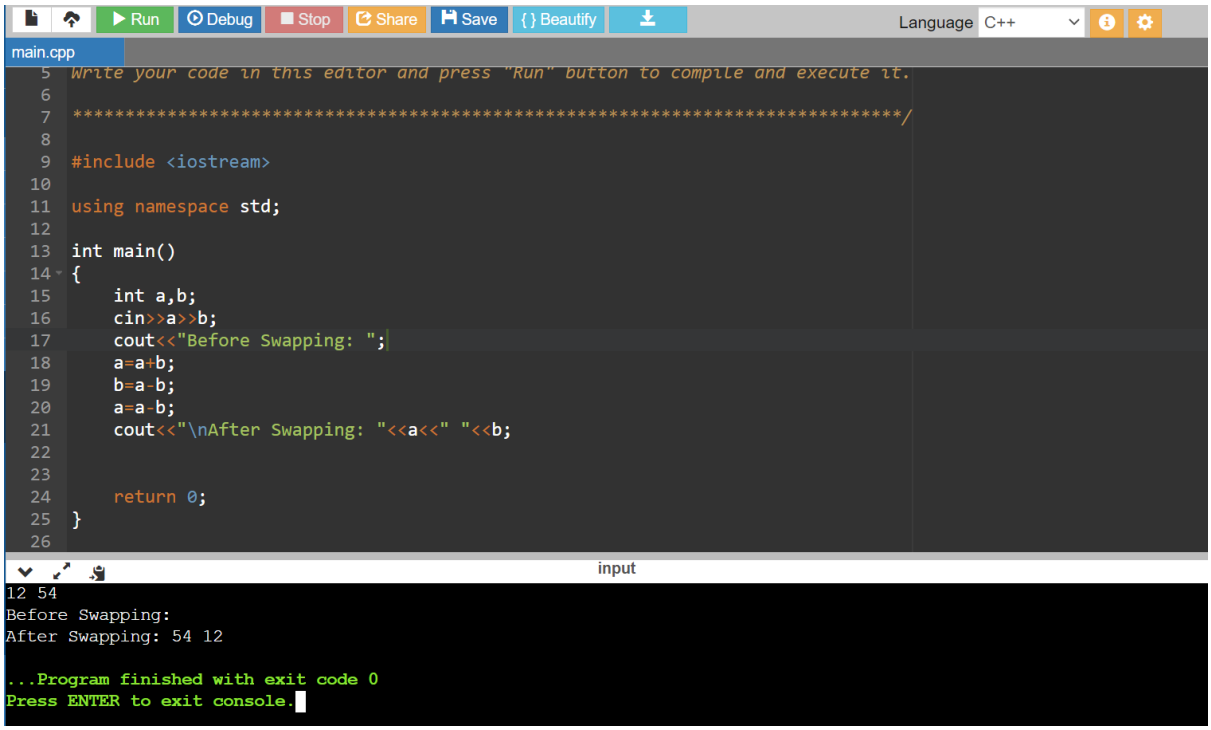
DSA BOOTCAMP ASSIGNMENT

Submitted By Dharm Vashisth (USICT, GGSIPU)

Q1. Write a program to Swap to two numbers.



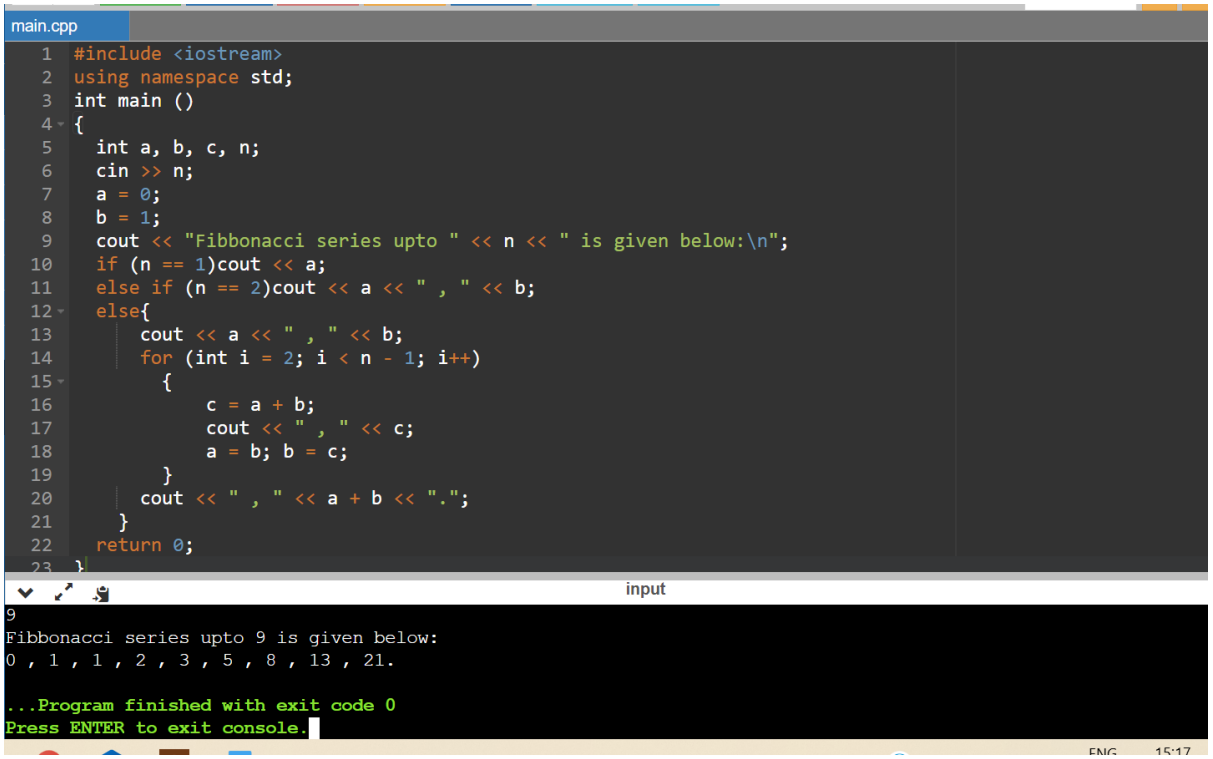
```
Run  Debug  Stop  Share  Save  { } Beautify                Language C++
main.cpp
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <iostream>

using namespace std;

int main()
{
    int a,b;
    cin>>a>>b;
    cout<<"Before Swapping: ";
    a=a+b;
    b=a-b;
    a=a-b;
    cout<<"\nAfter Swapping: "<<a<<" "<<b;


    return 0;
}

input
12 54
Before Swapping:
After Swapping: 54 12

...Program finished with exit code 0
Press ENTER to exit console.
```

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

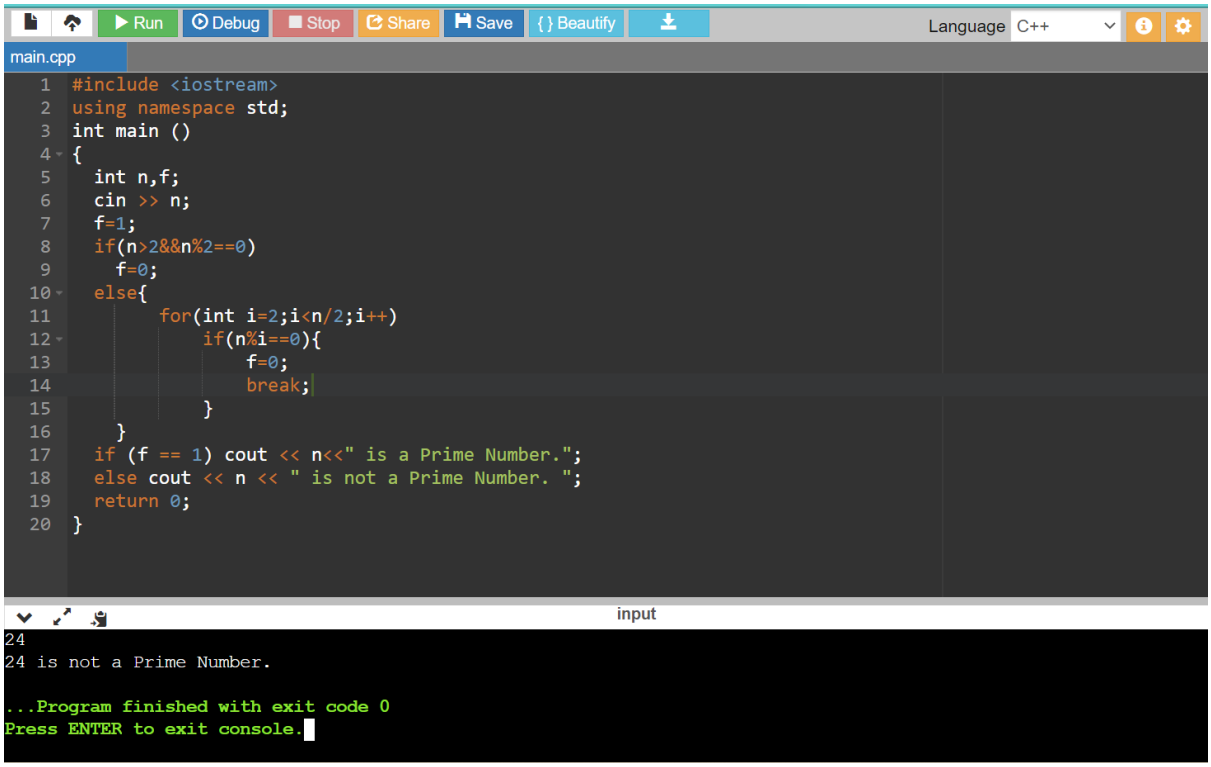
```
Run  Debug  Stop  Share  Save  { } Beautify                Language C++
main.cpp
Code, Compile, Run and Debug C++ program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <iostream>

using namespace std;

int main()
{
    int a,b,c,largest;
    cin>>a>>b>>c;
    if(a>b && a>c)
        largest=a;
    else if(b>a && b>c)
        largest=b;
    else
        largest=c;

    cout<<"\nLargest Number among "<<a<<" , "<<b<<" and "<<c<<" is "<<largest<<endl;

input
43 -53 12

Largest Number among 43 , -53 and 12 is 43


...Program finished with exit code 0
Press ENTER to exit console.
```

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

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



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



Q6. Print this pattern using loops
For n=5

```
    *
   * *
  * * *
 * * * *
* * * * *
```

main.cpp

```
#include <iostream>
using namespace std;
int main ()
{
  int n=5;
    for(int i=0;i<n;i++){
        for(int k=n-i-1;k>0;k--)
        cout<<" ";
        for(int j=0;j<=i;j++)
            cout<<"* ";
        cout<<endl;
    }
  return 0;
}
```

input

```
    *
   * *
  * * *
 * * * *
* * * * *


...Program finished with exit code 0
Press ENTER to exit console.
```

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

main.cpp

```cpp
#include <iostream>
using namespace std;
int main ()
{
  int ar[5],m1,m2;
  cout<<"Enter 5 integer elements ";
    for(int i=0;i<5;i++)
        cin>>ar[i];
    m1=ar[0];
    m2=ar[1];
    for(int i=2;i<5;i++){
        if(ar[i]>m1){
            m2=m1;
            m1=ar[i];
        }
    }
        cout<<"Second Largest Element is "<<m2;
  return 0;
}
```

input

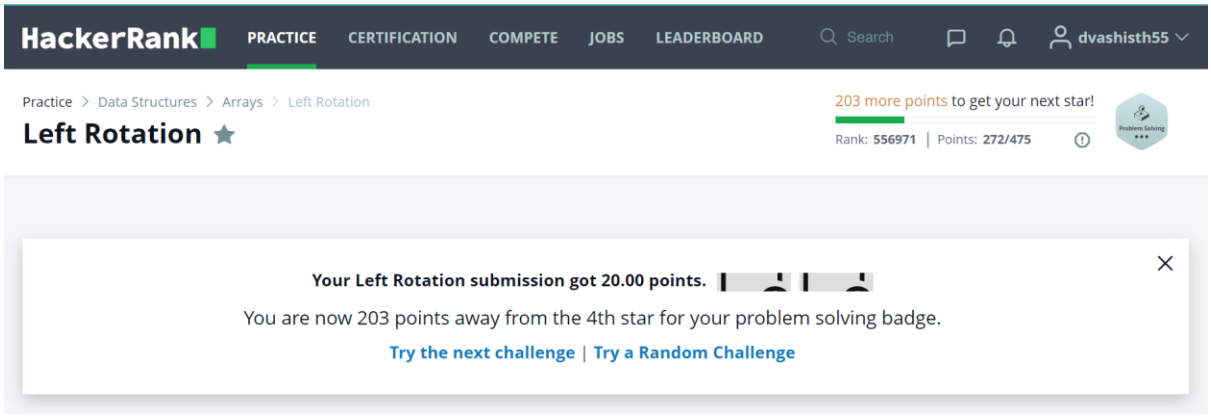
```
Enter 5 integer elements 234 -3455 23 54 877
Second Largest Element is 234

...Program finished with exit code 0
Press ENTER to exit console.
```

Q8. Left Rotation

```
#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) {
    int n=arr.size();
    int f,j;
    for(int i=0;i<d;i++){
        f=arr[0];
        for(j=0;j<n-1;j++){
            arr[j]=arr[j+1];
        }
        arr[n-1]=f;
    }
    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_in
put_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);
```

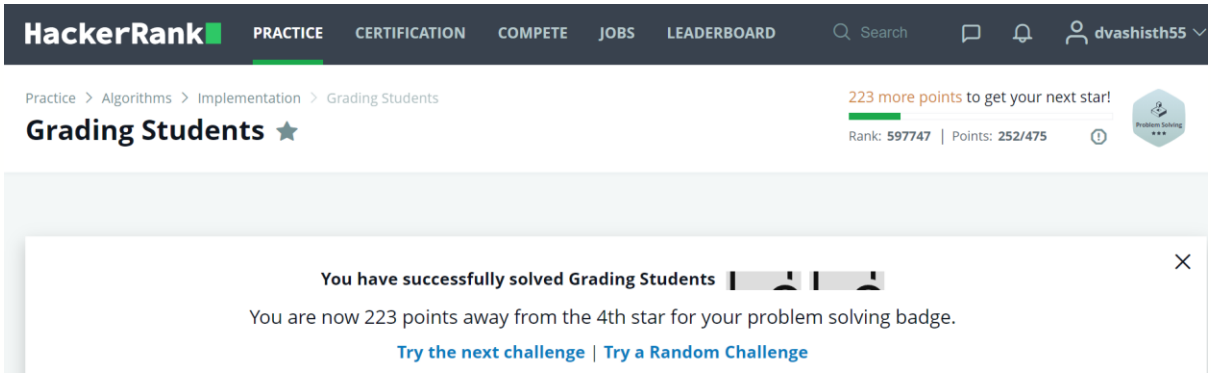
```
    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace)))
.base(),
        s.end()
    );

    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;
}
```



Q9. Grading Students

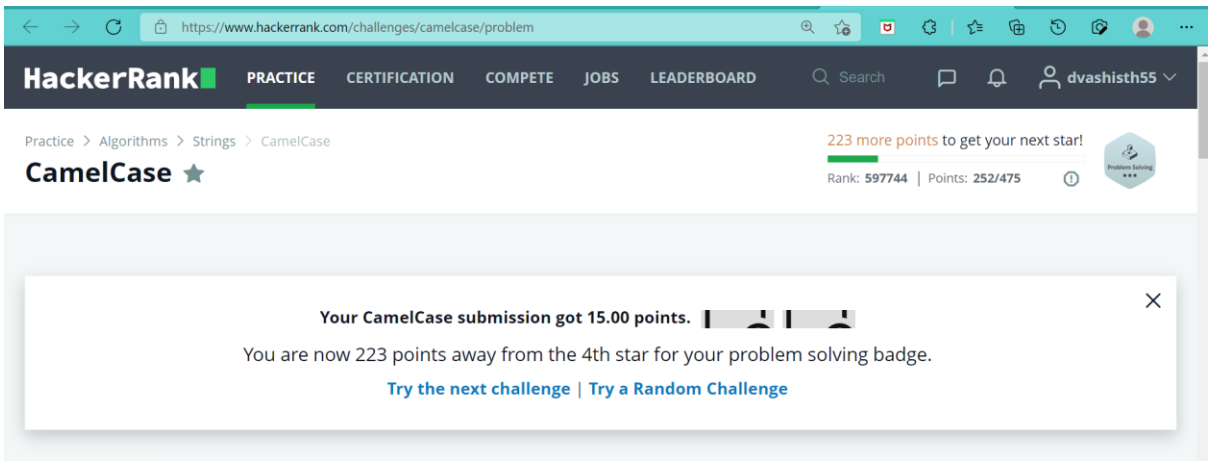
```
#include <bits/stdc++.h>

using namespace std;
string ltrim(const string &);
string rtrim(const string &);

vector<int> gradingStudents(vector<int> grades) {
    int n=grades.size();
    for(int i=0;i<n;i++){

        if(grades[i]>=38){
            if(grades[i]%5>=3){
                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;
}
```



Q10. CamelCase

```
#include <bits/stdc++.h>

using namespace std;

int camelcase(string s) {
    int n=1,i=0;
    while (s[i]!='\0') {
        if(s[i]>='A'&&s[i]<='Z')
            n++;
        i++;
    }
    return n;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string s;
    getline(cin, s);

    int result = camelcase(s);

    fout << result << "\n";

    fout.close();

    return 0;
}
```



The End