1. Write a program to swap to two nos.

#include<iostream>

using namespace std;

int main(){

    int a,b;

    cout<<"Enter value of a";

    cin>>a;

    cout<<"Enter value of b";

    cin>>b;

    cout<<"Value of a before swapping: "<<a<<endl;

    cout<<"Value of b before swapping: "<<b<<endl;

    swap(a,b);

    cout<<"Value of a after swapping: "<<a<<endl;

    cout<<"Value of b after swapping: "<<b<<endl;

    return 0;

}

2) Write a program to find the largest no. among three nos. entered by the user.

#include<iostream>

using namespace std;

int main(){

    int a,b,c;

    cout<<"Enter the value of a, b, c;"<<endl;

    cin>>a>>b>>c;

    if(a>b && a>c){

        cout<<a<<" is greater"<<endl;

    }

    else if(b>a && b>c){

        cout<<b<<" is greater"<<endl;

    }

    else{

        cout<<c<<" is greater"<<endl;

    }

1. Write a program to check wheather a year entered by a user is leap year or not.

#include<iostream>

using namespace std;

int main(){

    int year;

    cout<<"Enter any year"<<endl;

    cin>>year;

    if(((year%4==0) && (year%100!=0)) || (year%400==0)){

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

    }

    else{

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

    }

    return 0;

}

1. Write a program to display Fibonacci series upto nth term.

#include<iostream>

using namespace std;

int main(){

    int f1,f2,f3,i,terms;

    cout<<"Enter no. of terms"<<endl;

    cin>>terms;

    f1=0;

    f2=1;

    cout<<"Fibbonaci Series"<<endl;

    cout<<f1<<endl;

    cout<<f2<<endl;

    i=3;

    while(i<=terms)

    {

        f3=f1+f2;

        cout<<f3<<endl;

        f1=f2;

        f2=f3;

        i++;

    }

return 0;

}

1. Write a program to check wheather a number is prime or not.

#include<iostream>

using namespace std;

int main(){

int n,i;

cout<<"Enter the value of n"<<endl;

cin>>n;

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

{

    if(n%i==0)

        break;

}

if(i==n)

    cout<<n<<" is prime no. "<<endl;

else

    cout<<n<<" is not a prime no. "<<endl;

return 0;

}

1. Print the pattern using loops.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

#include <iostream>

#include<conio.h>

using namespace std;

int main() {

    int rows, i, j;

    cout << "Enter the number of rows : ";

    cin>>rows;

    for (i = 0; i < rows; i++) {

        for (j = rows; j > i; j--) {

            cout << " ";

        }

        for (int k = 1; k <= i + 1; k++) {

            cout << " \* ";

        }

        cout << "\n";

    }

    getch();

    return 0;

}

1. Write a program that take n elements from the user and displays the second largest elements of an array.

#include <iostream>

using namespace std;

 int main(){

  int arr1[50],n,i,j=0,lrg,lrg2nd;

       cout<<"Input the size of array : ";

       cin>>n;

    /\* Stored values into the array\*/

       cout<<"Input" <<n<<" elements in the array :";

       for(i=0;i<n;i++)

            {

          cout<<"element - : "<<i;

          cin>>arr1[i];

        }

/\* find location of the largest element in the array \*/

//   lrg=arr1[0];

   lrg=0;

  for(i=0;i<n;i++)

  {

      if(lrg<arr1[i])

      {

           lrg=arr1[i];

           j = i;

      }

  }

/\* ignore the largest element and find the 2nd largest element in the array \*/

   lrg2nd=0;

  for(i=0;i<n;i++)

  {

     if(i==j)

        {

          i++;  /\* ignoring the largest element \*/

          i--;

        }

      else

        {

          if(lrg2nd<arr1[i])

         {

               lrg2nd=arr1[i];

             }

        }

  }

  cout<<"The Second largest element in the array is : "<<lrg2nd;

}

8)

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

vector<string> split(const string &);

vector<int> rotateLeft(int d, vector<int> 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;

}

9)

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

vector<int> gradingStudents(vector<int> 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;

}

10)

#include <bits/stdc++.h>

using namespace std;

int camelcase(string s) {

}

int main()

{

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

    string s;

    getline(cin, s);

    int result = camelcase(s);

    fout << result << "\n";

    fout.close();

    return 0;

}