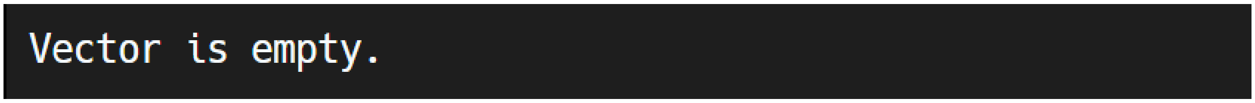
1- Write a program and Check if a vector is empty.

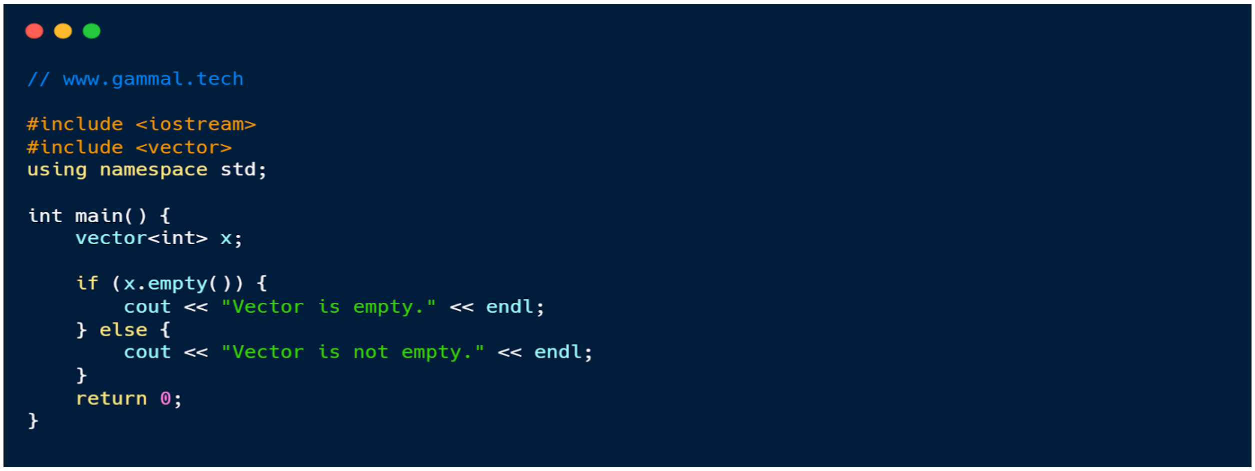
اكتب برنامجًا وتحقق مما إذا كان vector فارغًا.

Output



```
Vector is empty.
```

Solution

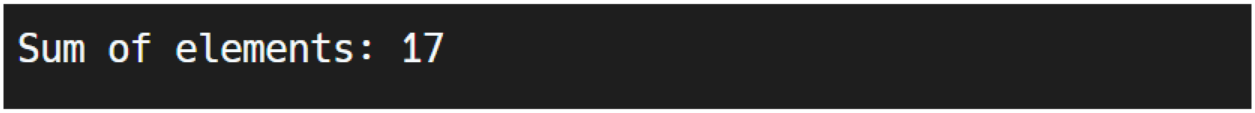


```
// www.gammal.tech

#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> x;

    if (x.empty()) {
        cout << "Vector is empty." << endl;
    } else {
        cout << "Vector is not empty." << endl;
    }
    return 0;
}
```

2- Create a vector and find the sum of its elements {3, 7, 2, 10, -5}

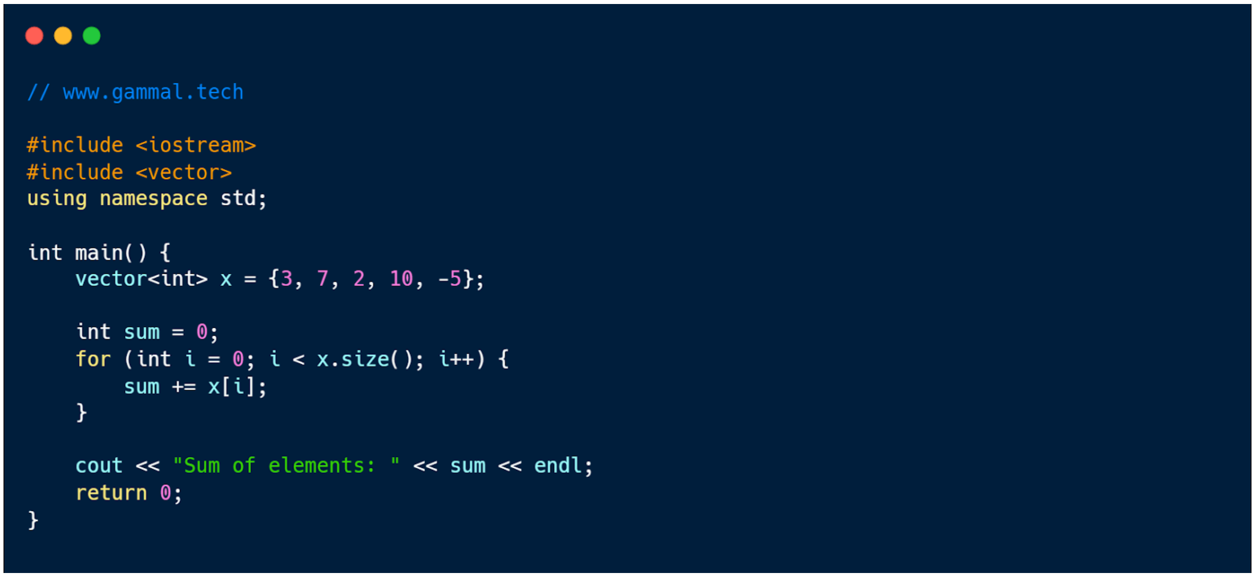
قم بإنشاء vector وإيجاد مجموع عناصره {3، 7، 2، 10، -5}

Output



```
Sum of elements: 17
```

Solution



```
// www.gammal.tech

#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> x = {3, 7, 2, 10, -5};

    int sum = 0;
    for (int i = 0; i < x.size(); i++) {
        sum += x[i];
    }

    cout << "Sum of elements: " << sum << endl;
    return 0;
}
```

3- Create two vectors and concatenate them into a third vector

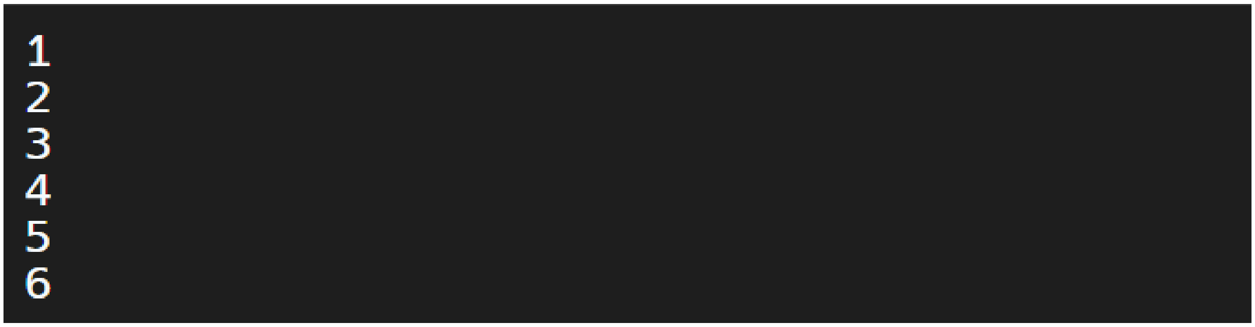
x = {1, 2, 3}; y = {4, 5, 6};

أنشئ two vectors وقم بربطهما في vector ثالث

x = {1, 2, 3}

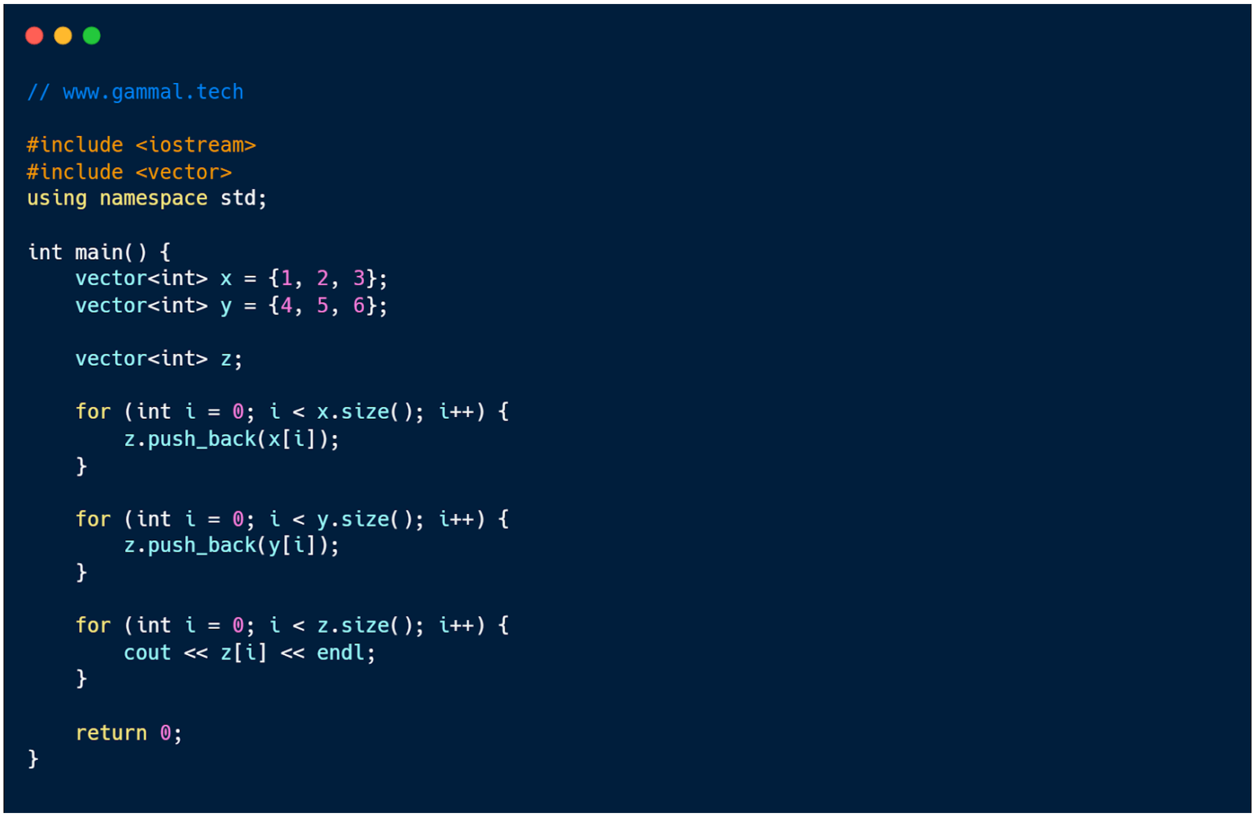
y= {4، 5، 6}

Output



```
1
2
3
4
5
6
```

Solution



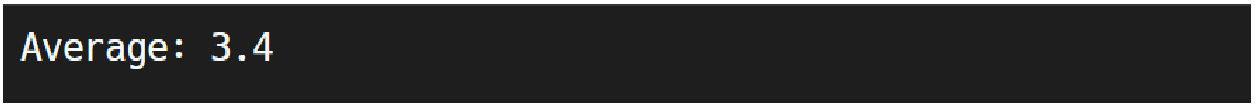
```
// www.gammal.tech

#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> x = {1, 2, 3};
    vector<int> y = {4, 5, 6};

    vector<int> z;

    for (int i = 0; i < x.size(); i++) {
        z.push_back(x[i]);
    }

    for (int i = 0; i < y.size(); i++) {
        z.push_back(y[i]);
    }

    for (int i = 0; i < z.size(); i++) {
        cout << z[i] << endl;
    }

    return 0;
}
```

4- Create a vector of floating-point numbers and find the average numbers = {2.5, 3.0, 5.5, 1.8, 4.2};

أنشئ متجهًا لأرقام floating-point وأوجد المتوسط

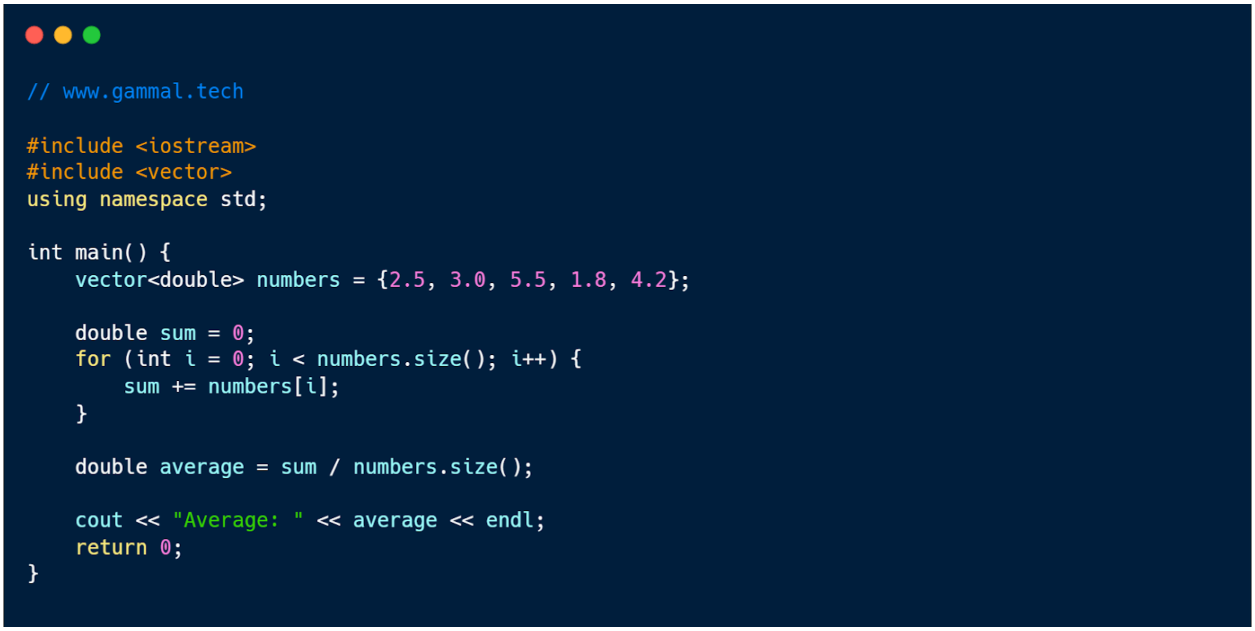
numbers= {2.5، 3.0، 5.5، 1.8، 4.2}؛

Output



```
Average: 3.4
```

Solution



```cpp
// www.gammal.tech

#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<double> numbers = {2.5, 3.0, 5.5, 1.8, 4.2};

    double sum = 0;
    for (int i = 0; i < numbers.size(); i++) {
        sum += numbers[i];
    }

    double average = sum / numbers.size();

    cout << "Average: " << average << endl;
    return 0;
}
```

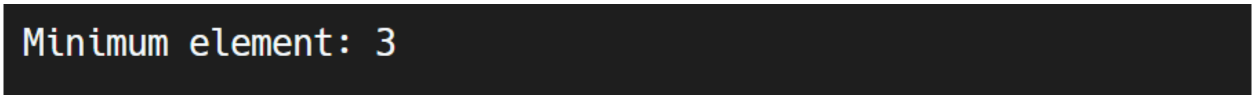
5- Create a vector and find the minimum element

x = {8, 3, 12, 5, 7}

قم بإنشاء vector وابحث عن الحد الأدنى للعنصر

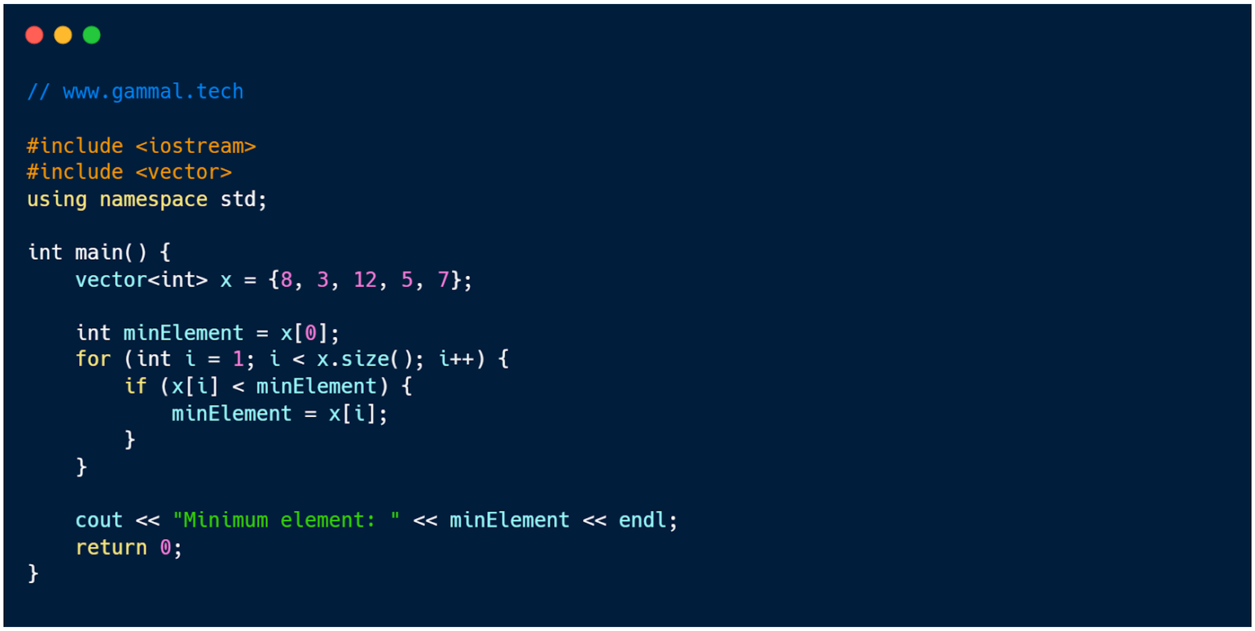
x = {8, 3, 12, 5, 7}

Output



```
Minimum element: 3
```

Solution

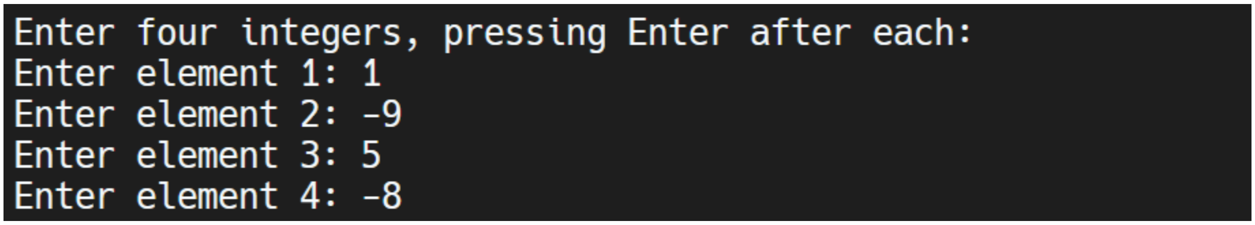


```cpp
// www.gammal.tech

#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> x = {8, 3, 12, 5, 7};

    int minElement = x[0];
    for (int i = 1; i < x.size(); i++) {
        if (x[i] < minElement) {
            minElement = x[i];
        }
    }

    cout << "Minimum element: " << minElement << endl;
    return 0;
}
```

6- Write a program to print only the negative integers entered by the user using a stack.

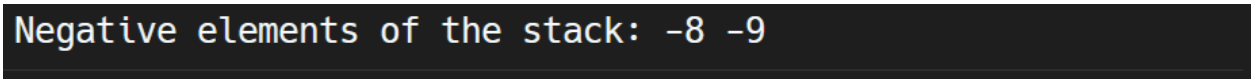
اكتب برنامجًا لطباعة الأعداد الصحيحة السالبة التي أدخلها المستخدم باستخدام stack فقط.

Input



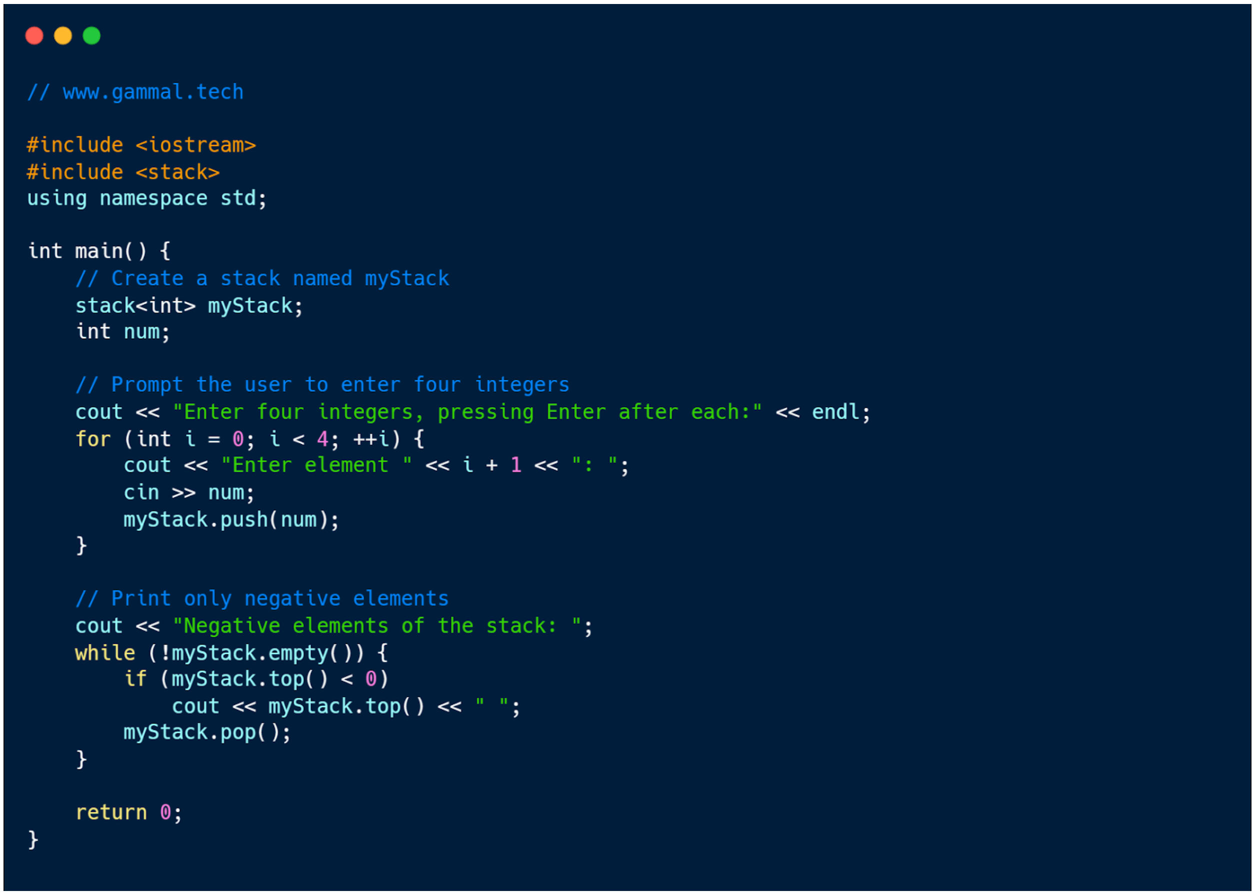
```
Enter four integers, pressing Enter after each:
Enter element 1: 1
Enter element 2: -9
Enter element 3: 5
Enter element 4: -8
```

Output



```
Negative elements of the stack: -8 -9
```

Solution

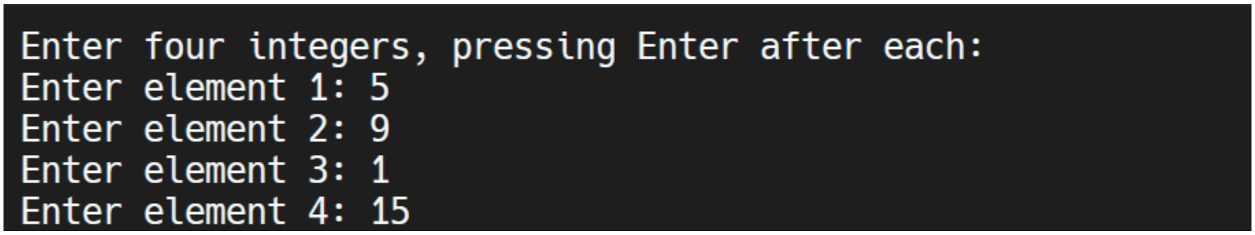


```
// www.gammal.tech

#include <iostream>
#include <stack>
using namespace std;

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;

    // Prompt the user to enter four integers
    cout << "Enter four integers, pressing Enter after each:" << endl;
    for (int i = 0; i < 4; ++i) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> num;
        myStack.push(num);
    }

    // Print only negative elements
    cout << "Negative elements of the stack: ";
    while (!myStack.empty()) {
        if (myStack.top() < 0)
            cout << myStack.top() << " ";
        myStack.pop();
    }

    return 0;
}
```

7- Write a program to print only the integers between 5 and 12 (inclusive) entered by the user using a stack.

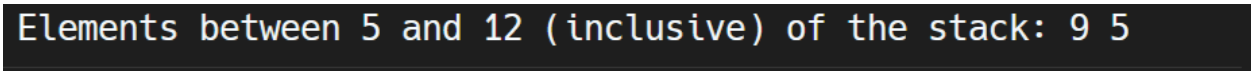
اكتب برنامجًا لطباعة الأعداد الصحيحة بين 5 و12 (inclusive) فقط التي أدخلها المستخدم باستخدام stack.

Input



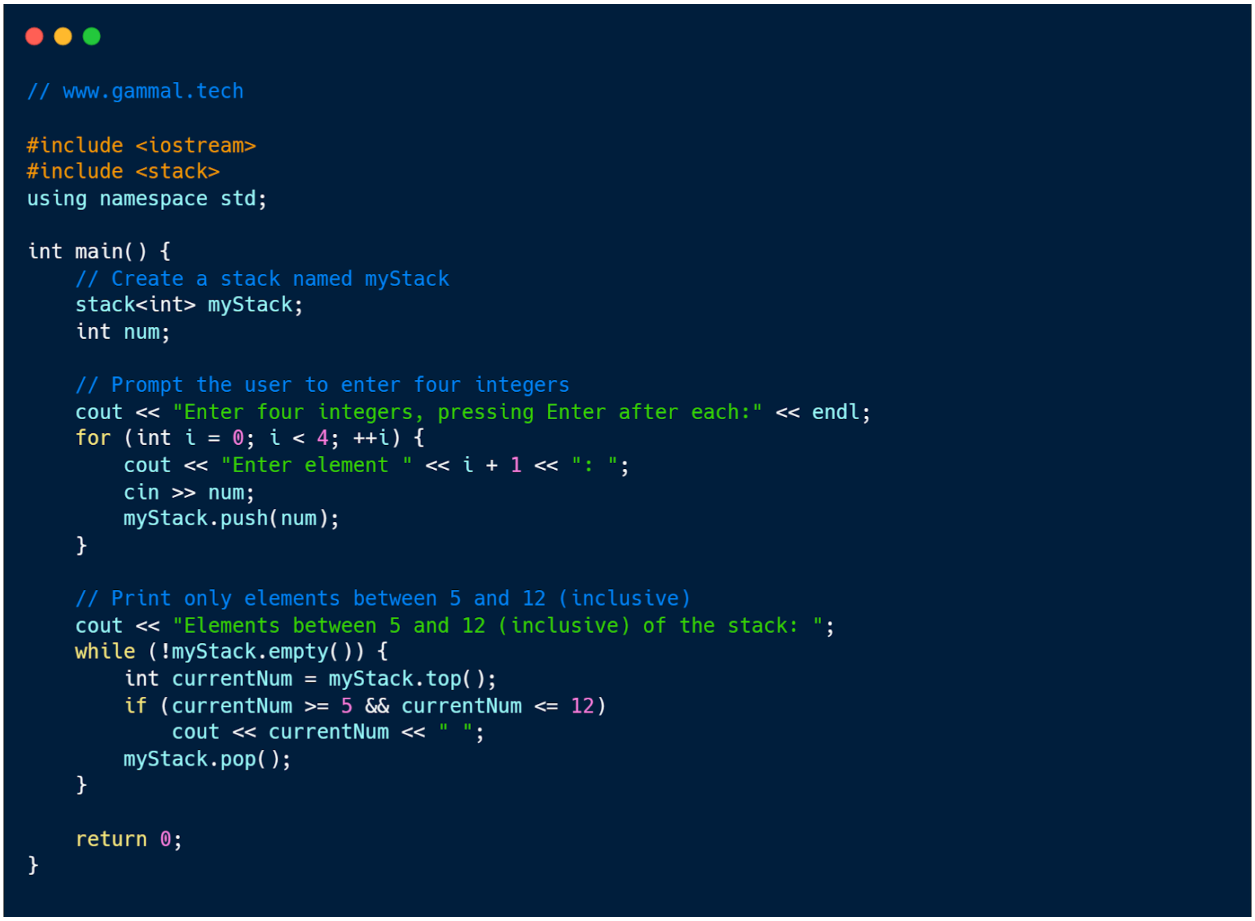
```
Enter four integers, pressing Enter after each:
Enter element 1: 5
Enter element 2: 9
Enter element 3: 1
Enter element 4: 15
```

Output



```
Elements between 5 and 12 (inclusive) of the stack: 9 5
```

Solution

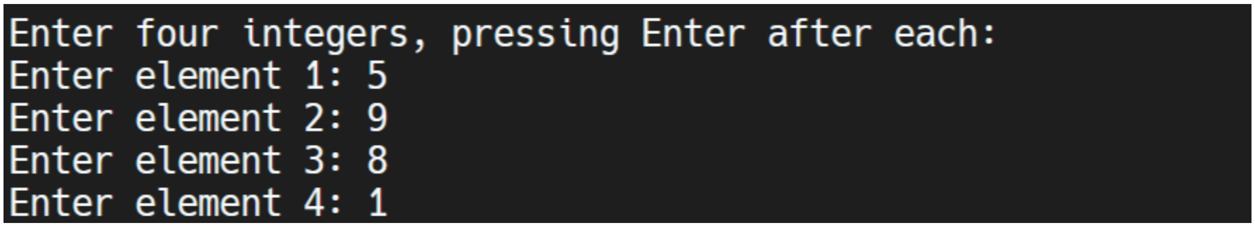


```
// www.gammal.tech

#include <iostream>
#include <stack>
using namespace std;

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;

    // Prompt the user to enter four integers
    cout << "Enter four integers, pressing Enter after each:" << endl;
    for (int i = 0; i < 4; ++i) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> num;
        myStack.push(num);
    }

    // Print only elements between 5 and 12 (inclusive)
    cout << "Elements between 5 and 12 (inclusive) of the stack: ";
    while (!myStack.empty()) {
        int currentNum = myStack.top();
        if (currentNum >= 5 && currentNum <= 12)
            cout << currentNum << " ";
        myStack.pop();
    }

    return 0;
}
```

8- Write a program to print the largest integer entered by the user using a stack.

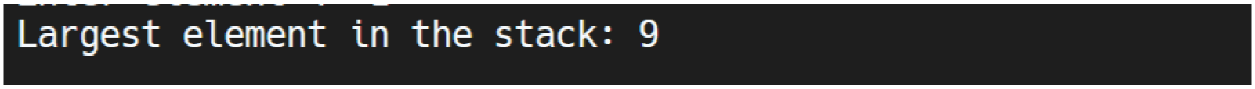
اكتب برنامجًا لطباعة أكبر عدد صحيح يدخله المستخدم باستخدام stack.

Input



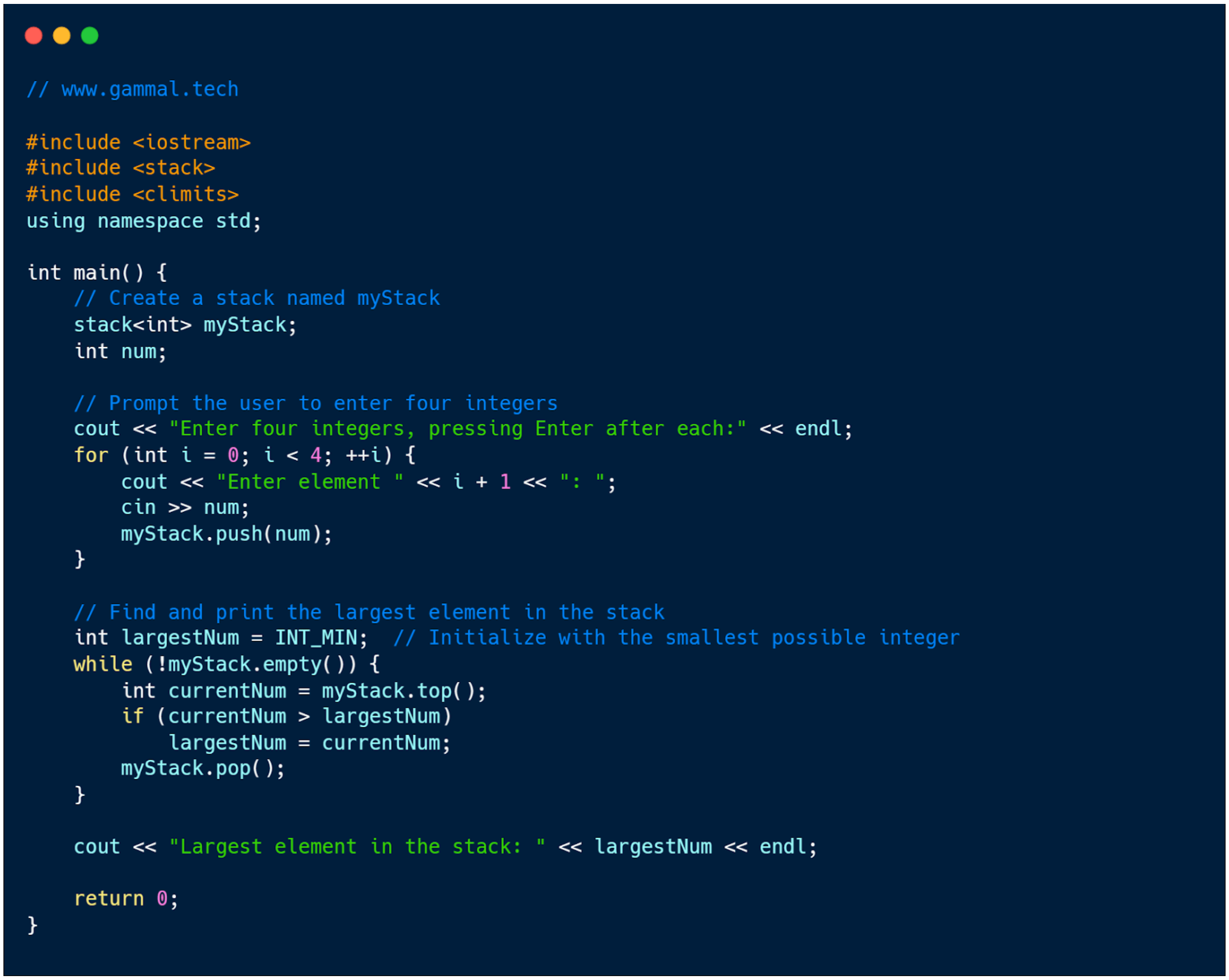
```
Enter four integers, pressing Enter after each:
Enter element 1: 5
Enter element 2: 9
Enter element 3: 8
Enter element 4: 1
```

Output



```
Largest element in the stack: 9
```

Solution

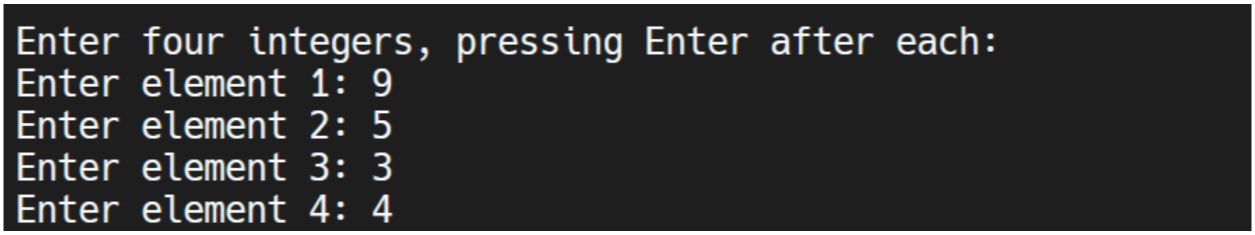


```cpp
// www.gammal.tech

#include <iostream>
#include <stack>
#include <climits>
using namespace std;

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;

    // Prompt the user to enter four integers
    cout << "Enter four integers, pressing Enter after each:" << endl;
    for (int i = 0; i < 4; ++i) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> num;
        myStack.push(num);
    }

    // Find and print the largest element in the stack
    int largestNum = INT_MIN;  // Initialize with the smallest possible integer
    while (!myStack.empty()) {
        int currentNum = myStack.top();
        if (currentNum > largestNum)
            largestNum = currentNum;
        myStack.pop();
    }

    cout << "Largest element in the stack: " << largestNum << endl;

    return 0;
}
```

9- Write a program to print the smallest integer entered by the user using a stack.

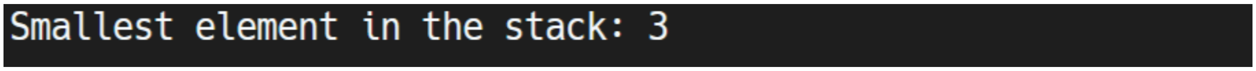
اكتب برنامجًا لطباعة أصغر عدد صحيح يدخله المستخدم باستخدام stack.

Input



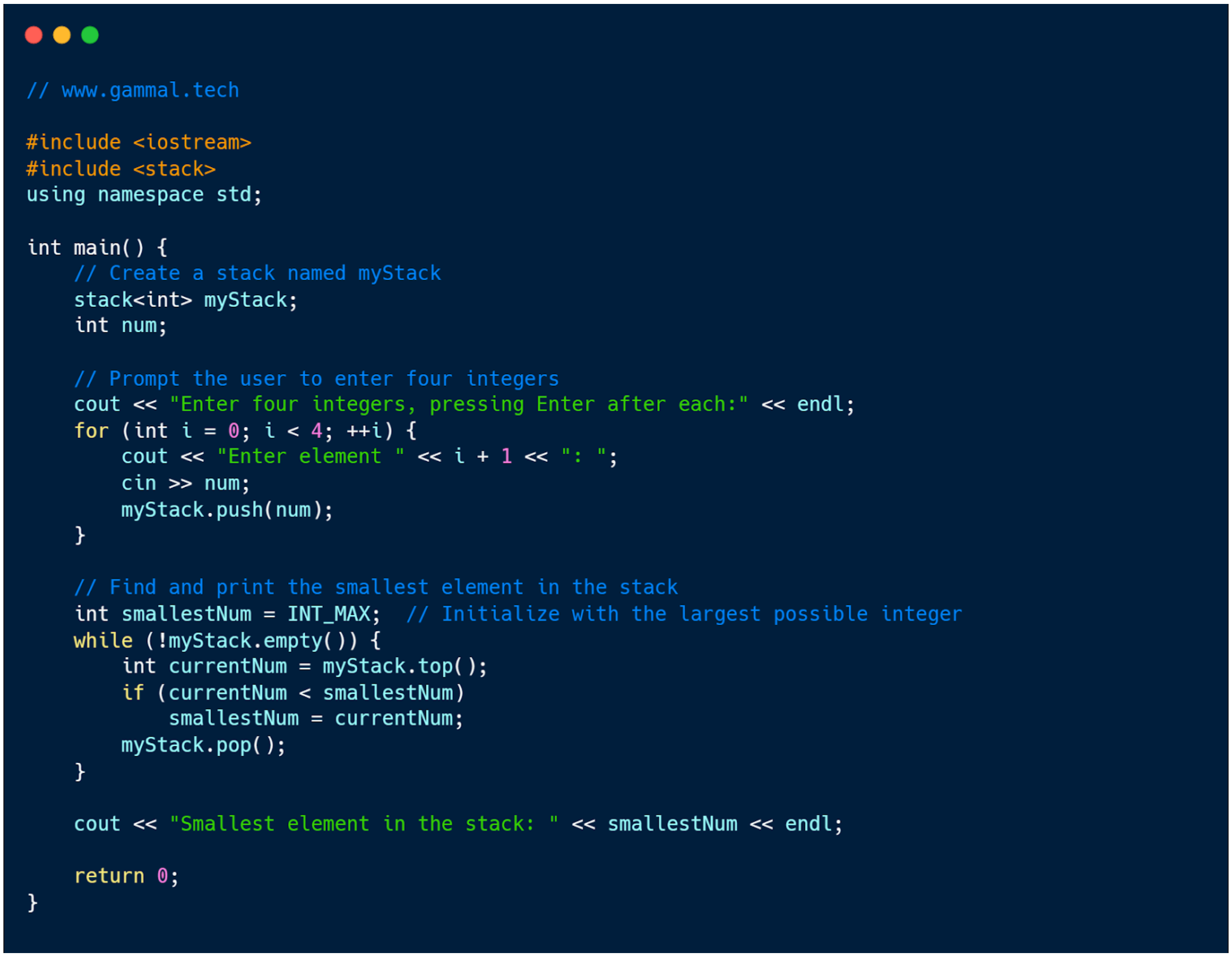
```
Enter four integers, pressing Enter after each:
Enter element 1: 9
Enter element 2: 5
Enter element 3: 3
Enter element 4: 4
```

Output



```
Smallest element in the stack: 3
```

Solution

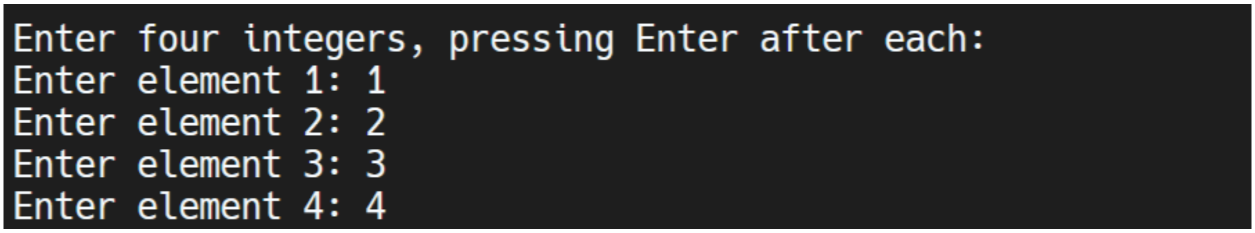


```cpp
// www.gammal.tech

#include <iostream>
#include <stack>
using namespace std;

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;

    // Prompt the user to enter four integers
    cout << "Enter four integers, pressing Enter after each:" << endl;
    for (int i = 0; i < 4; ++i) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> num;
        myStack.push(num);
    }

    // Find and print the smallest element in the stack
    int smallestNum = INT_MAX;  // Initialize with the largest possible integer
    while (!myStack.empty()) {
        int currentNum = myStack.top();
        if (currentNum < smallestNum)
            smallestNum = currentNum;
        myStack.pop();
    }

    cout << "Smallest element in the stack: " << smallestNum << endl;

    return 0;
}
```

10- Write a C++ program that takes four integers from the user, pushes them onto a stack, and then prints the sum of the even integers in the stack.

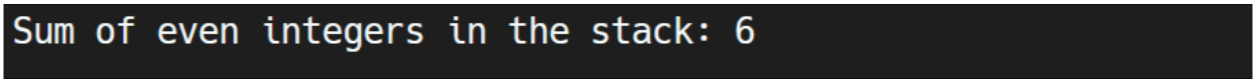
اكتب برنامج C++ يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack
ثم يطبع مجموع الأعداد الصحيحة الزوجية في stack .

Input



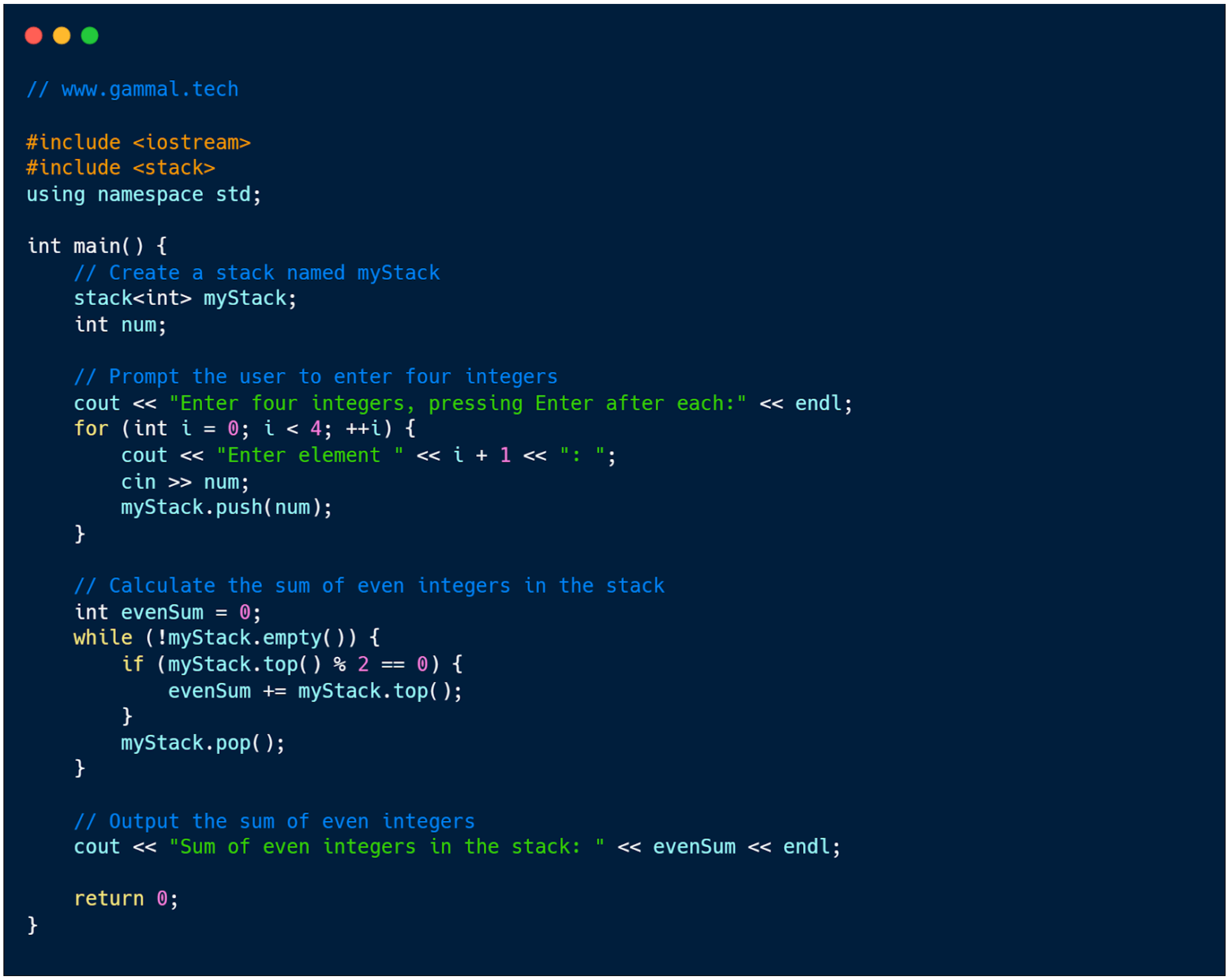
```
Enter four integers, pressing Enter after each:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
```

Output



```
Sum of even integers in the stack: 6
```

Solution



```cpp
// www.gammal.tech

#include <iostream>
#include <stack>
using namespace std;

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;

    // Prompt the user to enter four integers
    cout << "Enter four integers, pressing Enter after each:" << endl;
    for (int i = 0; i < 4; ++i) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> num;
        myStack.push(num);
    }

    // Calculate the sum of even integers in the stack
    int evenSum = 0;
    while (!myStack.empty()) {
        if (myStack.top() % 2 == 0) {
            evenSum += myStack.top();
        }
        myStack.pop();
    }

    // Output the sum of even integers
    cout << "Sum of even integers in the stack: " << evenSum << endl;

    return 0;
}
```