1- Write a C++ program that demonstrates the usage of the stack container. The program should perform the following operations:

Create a stack named myStack.
Push the values 10, 20, and 30 onto the stack.
Print the top element of the stack.
Pop one element from the stack.
Print the top element of the stack after the pop operation.

Output

```
Top element: 30
After pop, top element: 20
```

Solution

```cpp
// www.gammal.tech

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

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

    // Push values 10, 20, and 30 onto the stack
    myStack.push(10);
    myStack.push(20);
    myStack.push(30);

    // Print the top element of the stack
    cout << "Top element: " << myStack.top() << endl;

    // Pop one element from the stack
    myStack.pop();

    // Print the top element of the stack after the pop operation
    cout << "After pop, top element: " << myStack.top() << endl;

    return 0;
}
```

2- Write a C++ program that demonstrates the usage of the stack container. The program should perform the following operations:

Create a stack named myStack.
Push the values 10, 20, 30, and 40 onto the stack.
Print all elements of the stack by popping and displaying them.

Output

```
40 30 20 10
```

Solution

```cpp
// www.gammal.tech

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

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

    // Push values 10, 20, 30, and 40 onto the stack
    myStack.push(10);
    myStack.push(20);
    myStack.push(30);
    myStack.push(40);

    // Print all elements of the stack by popping and displaying them
    while (!myStack.empty()) {
        cout << myStack.top() << " ";
        myStack.pop();
    }

    return 0;
}
```

3- Write a C++ program that prompts the user to enter four integers. The program should use a stack to store these integers, and then it should print the elements of the stack by popping and displaying them.

Input

```
Enter four integers, pressing Enter after each:
Enter element 1: 20
Enter element 2: 30
Enter element 3: 40
Enter element 4: 50
```

Output

```
Elements of the stack: 50 40 30 20
```

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);
          }

          // Print all elements of the stack by popping and displaying them
          cout << "Elements of the stack: ";
          while (!myStack.empty()) {
              cout << myStack.top() << " ";
              myStack.pop();
          }

          return 0;
      }
```

4- Write a C++ program that prompts the user to enter four integers. The program should use a stack to store these integers and print only the even elements of the stack.

اكتب برنامج C++ يطلب من المستخدم إدخال أربعة أعداد صحيحة. يجب أن يستخدم البرنامج stack لتخزين هذه الأعداد الصحيحة وطباعة العناصر الزوجية فقط stack.

Input

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

Output

```
Even elements of the stack: 4 2
```

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);
    }

    // Print even elements of the stack by popping and displaying them
    cout << "Even elements of the stack: ";
    while (!myStack.empty()) {
        if (myStack.top() % 2 == 0)
            cout << myStack.top() << " ";
        myStack.pop();
    }

    return 0;
}
```

4- Write a C++ program that prompts the user to enter four integers. The program should use a stack to store these integers and print only the odd elements of the stack.

اكتب برنامج C++ يطلب من المستخدم إدخال أربعة أعداد صحيحة. يجب أن يستخدم البرنامج stack لتخزين هذه الأعداد الصحيحة وطباعة العناصر الفردية فقط stack.

Input

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

Output

```
Odd elements of the stack: 5 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);
    }

    // Print odd elements of the stack by popping and displaying them
    cout << "Odd elements of the stack: ";
    while (!myStack.empty()) {
        if (myStack.top() % 2 != 0)
            cout << myStack.top() << " ";
        myStack.pop();
    }

    return 0;
}
```

5- Write a program C++ program to additionally print only the prime numbers among the entered integers using a stack.

اكتب برنامجًا C++ لطباعة الأعداد الأولية فقط بين الأعداد الصحيحة المدخلة باستخدام stack.

Input

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

Output

```
Prime elements of the stack: 5 3 2
```

Solution

```cpp
// www.gammal.tech

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

// Function to check if a number is prime
bool isPrime(int n) {
    if (n <= 1)
        return false;
    for (int i = 2; i <= sqrt(n); ++i) {
        if (n % i == 0)
            return false;
    }
    return true;
}

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 prime elements of the stack by popping and displaying them
    cout << "Prime elements of the stack: ";
    while (!myStack.empty()) {
        if (isPrime(myStack.top()))
            cout << myStack.top() << " ";
        myStack.pop();
    }

    return 0;
}
```

6- Write a program to count and print the number of even integers entered by the user using a stack.

اكتب برنامجًا لحساب وطباعة عدد الأعداد الصحيحة الزوجية التي أدخلها المستخدم باستخدام stack.

Input

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

Output

```
Number of even elements entered: 2
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    // Create a stack named myStack
    stack<int> myStack;
    int num;
    int evenCount = 0;

    // 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);

        // Count even integers
        if (num % 2 == 0) {
            evenCount++;
        }
    }

    // Print the count of even elements
    cout << "Number of even elements entered: " << evenCount << endl;

    return 0;
}
```

7- 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

```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);
    }

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

    return 0;
}
```

8- 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

```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);
    }

    // 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;
}
```

9- 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;
}
```

10- 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;
}
```