1- Write a C++ program that takes a string as input, reverses the string using a stack, and then outputs the reversed string.

اكتب برنامج C++ الذي يأخذ string كمدخل، ويعكس string باستخدام stack ثم يخرج string المعكوسة.

Input

```
Enter a string: Gammal
```

Output

```
Reversed string: lammaG
```

Solution

```cpp
// www.gammal.tech

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

// Function to reverse a string using a stack
string reverseString(string input) {
    stack<char> charStack;

    // Push each character onto the stack
    for (char c : input) {
        charStack.push(c);
    }

    string reversed = "";

    // Pop each character from the stack to construct the reversed string
    while (!charStack.empty()) {
        reversed += charStack.top();
        charStack.pop();
    }

    return reversed;
}

int main() {
    string original, reversed;

    // Get input from the user
    cout << "Enter a string: ";
    cin >> original;

    // Call the reverseString function to reverse the string
    reversed = reverseString(original);

    // Output the reversed string
    cout << "Reversed string: " << reversed << endl;

    return 0;
}
```

2- Write a C++ program that evaluates a postfix expression using a stack. The program should take a postfix expression as input, perform the evaluation, and output the result.

اكتب برنامج C++ الذي يقوم بتقييم تعبير postfix باستخدام stack. يجب أن يأخذ البرنامج تعبير postfix كمدخل، ويقوم بإجراء التقييم، ويخرج النتيجة.

Example

Solution

```cpp
// www.gammal.tech

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

// Function to evaluate a postfix expression
int evaluatePostfix(string expression) {
    stack<int> operands;

    for (char c : expression) {
        if (isdigit(c)) {
            // If the character is a digit, push it onto the stack
            operands.push(c - '0');
        } else {
            // If the character is an operator, perform the corresponding operation
            int operand2 = operands.top();
            operands.pop();
            int operand1 = operands.top();
            operands.pop();

            switch (c) {
                case '+':
                    operands.push(operand1 + operand2);
                    break;
                case '-':
                    operands.push(operand1 - operand2);
                    break;
                case '*':
                    operands.push(operand1 * operand2);
                    break;
                case '/':
                    operands.push(operand1 / operand2);
                    break;
                case '^':
                    operands.push(pow(operand1, operand2));
                    break;
            }
        }
    }

    // The final result is at the top of the stack
    return operands.top();
}

int main() {
    string postfix;

    // Get input from the user
    cout << "Enter a postfix expression: ";
    cin >> postfix;

    // Call the evaluatePostfix function to evaluate the expression
    cout << "Result: " << evaluatePostfix(postfix) << endl;

    return 0;
}
```

3- Write a program that implements a minimum stack. The program should define a class MinStack with methods to push, pop, get the top element, and get the minimum element in constant time.

اكتب برنامجًا ينفذ الحد الأدنى من stack. يجب أن يحدد البرنامج فئة MinStack مع طرق push, pop, get the top element العلوي والحصول على الحد الأدنى من العنصر في وقت ثابت.

Output

```
Minimum element: 2
Top element: 2
Minimum element: 2
```

Solution

```cpp
// www.gammal.tech

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

// Class definition for MinStack
class MinStack {
private:
    stack<int> mainStack; // Stack to store elements
    stack<int> minStack;  // Stack to store minimum elements

public:
    // Method to push an element onto the stack
    void push(int value) {
        mainStack.push(value); // Push the value onto the main stack
        // If the minStack is empty or the new value is less than or equal to the current minimum, push it onto the minStack
        if (minStack.empty() || value <= minStack.top()) {
            minStack.push(value);
        }
    }

    // Method to pop the top element from the stack
    void pop() {
        if (!mainStack.empty()) {
            // If the top element of the main stack is equal to the top element of the minStack, pop from both stacks
            if (mainStack.top() == minStack.top()) {
                minStack.pop();
            }
            mainStack.pop();
        }
    }

    // Method to get the top element of the stack
    int top() {
        return mainStack.top();
    }

    // Method to get the minimum element of the stack
    int getMin() {
        return minStack.top();
    }
};

int main() {
    // Create an instance of the MinStack
    MinStack myMinStack;

    // Push elements onto the stack
    myMinStack.push(3);
    myMinStack.push(2);
    myMinStack.push(5);

    // Display the minimum element
    cout << "Minimum element: " << myMinStack.getMin() << endl;

    // Pop an element from the stack
    myMinStack.pop();

    // Display the top and minimum elements after popping
    cout << "Top element: " << myMinStack.top() << endl;
    cout << "Minimum element: " << myMinStack.getMin() << endl;

    return 0;
}
```

4- Write a C++ program that implements a stack using a linked list. Define a class Node to represent each node in the linked list and a class Stack to implement stack operations (push, pop, and getTop).

Output

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

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Class definition for Node
class Node {
public:
    int data;   // Data of the node
    Node* next;  // Pointer to the next node

    // Constructor to initialize a node with a given value
    Node(int value) {
        data = value;
        next = NULL;
    }
};

// Class definition for Stack
class Stack {
private:
    Node* top;  // Pointer to the top node of the stack

public:
    // Constructor to initialize the stack
    Stack() {
        top = NULL;
    }

    // Method to push an element onto the stack
    void push(int value) {
        Node* newNode = new Node(value);  // Create a new node with the given value
        newNode->next = top;              // Link the new node to the current top node
        top = newNode;                    // Update the top pointer to the new node
    }

    // Method to pop an element from the stack
    void pop() {
        if (top != NULL) {
            Node* temp = top;    // Temporarily store the current top node
            top = top->next;     // Update the top pointer to the next node
            delete temp;         // Delete the old top node
        } else {
            cout << "Stack Underflow!" << endl;
        }
    }

    // Method to get the top element of the stack
    int getTop() {
        if (top != NULL) {
            return top->data;  // Return the data of the top node
        } else {
            cout << "Stack is empty!" << endl;
            return -1;          // Indicating an empty stack
        }
    }
};

int main() {
    // Create an instance of the Stack class
    Stack myStack;

    // Push elements onto the stack
    myStack.push(10);
    myStack.push(20);
    myStack.push(30);

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

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

    // Display the top element after the pop operation
    cout << "After pop, top element: " << myStack.getTop() << endl;

    return 0;
}
```

5- Write a C++ program that takes four integers from the user, pushes them onto a stack, and then counts and prints the number of negative elements in the stack.

اكتب برنامج C++ يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack ثم يحسب ويطبع عدد العناصر السالبة في stack .

Input

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

Output

```
Number of negative elements: 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);
    }

    // Count and print the number of negative elements
    int negativeCount = 0;
    while (!myStack.empty()) {
        if (myStack.top() < 0)
            negativeCount++;
        myStack.pop();
    }

    cout << "Number of negative elements: " << negativeCount << endl;

    return 0;
}
```

6- Write a program that takes four integers from the user, pushes them onto a stack, and then prints all the elements between -2 and 8 (inclusive) in the stack.

اكتب برنامجًا يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack ثم يطبع جميع العناصر بين -2 و8 (شاملة) في stack .

Input

```
Enter four integers, pressing Enter after each:
Enter element 1: -1
Enter element 2: 5
Enter element 3: 9
Enter element 4: 11
```

Output

```
Elements between -2 and 8 in the stack: 5 -1
```

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 elements between -2 and 8 (inclusive)
    cout << "Elements between -2 and 8 in the stack: ";
    while (!myStack.empty()) {
        int currentElement = myStack.top();
        if (currentElement >= -2 && currentElement <= 8)
            cout << currentElement << " ";
        myStack.pop();
    }

    return 0;
}
```

7- Write a C++ program that takes four integers from the user, pushes them onto a stack, and then checks if the number 5 is found in the stack. Output whether the number 5 is found or not.

اكتب برنامج C++ يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack ثم يتحقق مما إذا كان الرقم 5 موجودًا في stack . إخراج ما إذا تم العثور على الرقم 5 أم لا.

Input

```
Enter four integers, pressing Enter after each:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 5
Enter element 4: 6
```

Output

```
The number 5 is found in the stack.
```

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);
    }

    // Check if the number 5 is found in the stack
    bool found = false;
    while (!myStack.empty()) {
        if (myStack.top() == 5) {
            found = true;
            break;
        }
        myStack.pop();
    }

    // Output whether the number 5 is found or not
    if (found) {
        cout << "The number 5 is found in the stack." << endl;
    } else {
        cout << "The number 5 is not found in the stack." << endl;
    }

    return 0;
}
```

8- Write a C++ program that takes four integers from the user, pushes them onto a stack, and then checks and outputs the number of occurrences of the number 5 in the stack.

اكتب برنامج C++ يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack ثم يتحقق ويخرج عدد مرات ظهور الرقم 5 في stack .

Input

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

Output

```
Number of occurrences of 5 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);
    }

    // Check the occurrence of the number 5 in the stack
    int occurrence = 0;
    while (!myStack.empty()) {
        if (myStack.top() == 5) {
            occurrence++;
        }
        myStack.pop();
    }

    // Output the number of occurrences of 5
    cout << "Number of occurrences of 5 in the stack: " << occurrence << endl;

    return 0;
}
```

9- Write a C++ program that takes four integers from the user, pushes them onto a stack, and then counts and outputs the number of odd integers in the stack.

اكتب برنامج C++ يأخذ أربعة أعداد صحيحة من المستخدم، ويدفعها إلى stack ثم يحسب ويخرج عدد الأعداد الصحيحة الفردية 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

```
Number of odd integers in the stack: 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);
    }

    // Count the number of odd integers in the stack
    int oddCount = 0;
    while (!myStack.empty()) {
        if (myStack.top() % 2 != 0) {
            oddCount++;
        }
        myStack.pop();
    }

    // Output the number of odd integers
    cout << "Number of odd integers in the stack: " << oddCount << 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;
}
```