

Άσκηση 1: Προσθήκη και Αφαίρεση Στοιχείων σε Στοίβα

Γράψε ένα πρόγραμμα που:

Δημιουργεί μια στοίβα και προσθέτει 5 αριθμούς.
Αφαιρεί έναν αριθμό από τη στοίβα.
Εμφανίζει τα στοιχεία που απομένουν στη στοίβα.

Οδηγίες:

Χρησιμοποίησε τη μέθοδο Push() για να προσθέσεις στοιχεία στη στοίβα.
Χρησιμοποίησε τη μέθοδο Pop() για να αφαιρέσεις το στοιχείο από την κορυφή.

Λύση:

csharp

```
using System;  
using System.Collections.Generic;
```

```
class Program  
{  
    static void Main()  
    {  
        Stack<int> stack = new Stack<int>();  
  
        // Προσθήκη 5 αριθμών στη στοίβα  
        stack.Push(10);  
        stack.Push(20);  
        stack.Push(30);  
        stack.Push(40);  
        stack.Push(50);  
  
        // Αφαίρεση του στοιχείου από την κορυφή  
        int removedElement = stack.Pop();  
        Console.WriteLine("Το στοιχείο που αφαιρέθηκε είναι: " + removedElement);  
  
        // Εκτύπωση των στοιχείων που απομένουν στη στοίβα  
        Console.WriteLine("Τα στοιχεία που απομένουν στη στοίβα:");  
        foreach (int number in stack)  
        {  
            Console.WriteLine(number);  
        }  
    }  
}
```

Άσκηση 2: Έλεγχος Ισορροπημένων Παρενθέσεων

Γράψε ένα πρόγραμμα που ελέγχει αν οι παρενθέσεις μιας μαθηματικής έκφρασης είναι ισορροπημένες. Ισορροπημένες παρενθέσεις σημαίνει ότι κάθε ανοιχτή παρένθεση έχει μια αντίστοιχη κλειστή παρένθεση.

Οδηγίες:

Χρησιμοποίησε μια στοίβα για να αποθηκεύεις τις ανοιχτές παρενθέσεις.

Όταν βρεις κλειστή παρένθεση, έλεγξε αν υπάρχει ανοιχτή παρένθεση στην κορυφή της στοίβας.

Λύση:

```
using System;
using System.Collections.Generic;

class Program
{
    static bool AreParenthesesBalanced(string expression)
    {
        Stack<char> stack = new Stack<char>();

        foreach (char ch in expression)
        {
            if (ch == '(')
            {
                stack.Push(ch); // Προσθήκη ανοιχτής παρένθεσης στη στοίβα
            }
            else if (ch == ')')
            {
                if (stack.Count == 0)
                {
                    return false; // Αν δεν υπάρχει ανοιχτή παρένθεση για να ταιριάζει, είναι μη
ισορροπημένη
                }
                stack.Pop(); // Αφαίρεση ανοιχτής παρένθεσης που ταιριάζει
            }
        }

        return stack.Count == 0; // Αν η στοίβα είναι άδεια, οι παρενθέσεις είναι ισορροπημένες
    }

    static void Main()
    {
        Console.Write("Εισάγετε μια έκφραση με παρενθέσεις: ");
        string expression = Console.ReadLine();

        bool isBalanced = AreParenthesesBalanced(expression);
```

```

    if (isBalanced)
    {
        Console.WriteLine("Οι παρενθέσεις είναι ισορροπημένες.");
    }
    else
    {
        Console.WriteLine("Οι παρενθέσεις ΔΕΝ είναι ισορροπημένες.");
    }
}
}

```

Άσκηση 3: Αντιστροφή μιας Σειράς Χαρακτήρων

Γράψε ένα πρόγραμμα που αντιστρέφει μια σειρά χαρακτήρων χρησιμοποιώντας μια στοίβα. Το πρόγραμμα θα διαβάσει ένα string και θα εμφανίζει το string με τους χαρακτήρες ανάποδα.

Οδηγίες:

Χρησιμοποίησε τη μέθοδο Push() για να προσθέσεις κάθε χαρακτήρα της συμβολοσειράς στη στοίβα.

Χρησιμοποίησε τη μέθοδο Pop() για να αφαιρέσεις τους χαρακτήρες με την αντίστροφη σειρά.

Λύση:

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.Write("Εισάγετε μια σειρά χαρακτήρων: ");
        string input = Console.ReadLine();

        Stack<char> stack = new Stack<char>();

        // Προσθήκη χαρακτήρων στη στοίβα
        foreach (char ch in input)
        {
            stack.Push(ch);
        }

        // Αφαίρεση χαρακτήρων και δημιουργία αντιστραμμένης συμβολοσειράς
        string reversed = "";
    }
}

```

```

while (stack.Count > 0)
{
    reversed += stack.Pop();
}

Console.WriteLine("Η αντιστραμμένη συμβολοσειρά είναι: " + reversed);
}
}

```

Άσκηση 4: Έλεγχος για Παλλινδρομικές Σειρές

Γράψε ένα πρόγραμμα που ελέγχει αν μια συμβολοσειρά είναι παλλινδρομική, δηλαδή αν διαβάζεται το ίδιο και από την αρχή και από το τέλος. Χρησιμοποίησε μια στοίβα για να αποθηκεύσεις τους χαρακτήρες και να συγκρίνεις.

Οδηγίες:

Πρόσθεσε κάθε χαρακτήρα της συμβολοσειράς στη στοίβα.

Βγάλε τους χαρακτήρες από τη στοίβα και σύγκρινέ τους με τους χαρακτήρες της αρχικής συμβολοσειράς.

Λύση:

```

using System;
using System.Collections.Generic;

class Program
{
    static bool IsPalindrome(string input)
    {
        Stack<char> stack = new Stack<char>();

        // Προσθήκη χαρακτήρων στη στοίβα
        foreach (char ch in input)
        {
            stack.Push(ch);
        }

        // Έλεγχος αν η συμβολοσειρά είναι ίδια και από τις δύο πλευρές
        foreach (char ch in input)
        {
            if (ch != stack.Pop())
            {
                return false; // Αν κάποιος χαρακτήρας δεν ταιριάζει, δεν είναι παλλίνδρομο
            }
        }
    }
}

```

```

    }
}

return true; // Αν όλοι οι χαρακτήρες ταιριάζουν, είναι παλλίνδρομο
}

static void Main()
{
    Console.Write("Εισάγετε μια σειρά χαρακτήρων: ");
    string input = Console.ReadLine();

    bool isPalindrome = IsPalindrome(input);

    if (isPalindrome)
    {
        Console.WriteLine("Η συμβολοσειρά είναι παλλινδρομική.");
    }
    else
    {
        Console.WriteLine("Η συμβολοσειρά ΔΕΝ είναι παλλινδρομική.");
    }
}
}

```

Άσκηση 5: Αναίρεση Ενεργειών (Undo)

Γράψε ένα πρόγραμμα που επιτρέπει στον χρήστη να εκτελεί ενέργειες και να κάνει αναίρεση (undo) της τελευταίας ενέργειας που έκανε, χρησιμοποιώντας στοίβα. Κάθε ενέργεια θα αποθηκεύεται στη στοίβα, και όταν ο χρήστης επιλέξει "Undo", θα αφαιρείται η τελευταία ενέργεια.

Οδηγίες:

Χρησιμοποίησε τη μέθοδο Push() για να προσθέσεις τις ενέργειες στη στοίβα.
Χρησιμοποίησε τη μέθοδο Pop() για να κάνεις αναίρεση της τελευταίας ενέργειας.

Λύση:

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {

```

```

Stack<string> actions = new Stack<string>();

while (true)
{
    Console.WriteLine("1. Εκτέλεση Ενέργειας");
    Console.WriteLine("2. Αναίρεση (Undo)");
    Console.WriteLine("3. Προβολή Ενεργειών");
    Console.WriteLine("4. Έξοδος");
    Console.Write("Επιλογή: ");
    int choice = int.Parse(Console.ReadLine());

    if (choice == 1)
    {
        Console.Write("Εισάγετε μια ενέργεια: ");
        string action = Console.ReadLine();
        actions.Push(action);
    }
    else if (choice == 2)
    {
        if (actions.Count > 0)
        {
            string undoneAction = actions.Pop();
            Console.WriteLine("Η ενέργεια '" + undoneAction + "' αναίρεθηκε.");
        }
        else
        {
            Console.WriteLine("Δεν υπάρχουν ενέργειες για αναίρεση.");
        }
    }
    else if (choice == 3)
    {
        Console.WriteLine("Οι ενέργειες είναι:");
        foreach (string action in actions)
        {
            Console.WriteLine(action);
        }
    }
    else if (choice == 4)
    {
        break;
    }
}
}

```