

Questão 1

namespace Pilha

{

class Program

{

static void Main(string[] args)

{

Stack<double> pilha = new Stack<double>();

String not = "3572-*4/+";

for(int i = 0; i < not.Length; i++)

{

if(not[i] == '+' || not[i] == '-' || not[i] == '*' || not[i] == '/')

{

switch (not[i])

{

case '+':

pilha.Push(pilha.Pop() + pilha.Pop());

break;

case '-':

double num = pilha.Pop();

double num2 = pilha.Pop();

pilha.Push(num2 - num);

break;

case '*':

pilha.Push(pilha.Pop() * pilha.Pop());

break;

case '/':

double num3 = pilha.Pop();

double num4 = pilha.Pop();

pilha.Push(num4 / num3);

break;

default:

break;

}

}

else

{

Console.WriteLine(not[i]);

pilha.Push(Double.Parse(not[i].ToString()));

}

}

Console.WriteLine(pilha.Peek());

```

        Console.ReadLine();
    }
}

```

Questão 1 - Classe Pilha

```

namespace Pilha
{
    class Pilha
    {
        private double[] vetor;
        private int topo;
        public Pilha(int tamanho)
        {
            vetor = new double[tamanho];
            topo = 0;
        }
        public double Peek()
        {
            return vetor[topo - 1];
        }
        public void Push(double num)
        {
            if(topo >= vetor.Length)
            {
                Console.WriteLine("Tamanho máximo.");
            }
            else
            {
                vetor[topo] = num;
                topo++;
            }
        }
        public double Pop()
        {
            double num = vetor[topo - 1];
            vetor[topo - 1] = 0;
            topo--;
            return num;
        }
        public void Clear()
        {
            for(int i = 0; i < vetor.Length; i++)

```

```

        {
            vetor[i] = 0;
        }
    }

}

internal class Program
{
    static void Main(string[] args)
    {
        Pilha pilha = new Pilha(5);
        String not = "3572-*4/+";
        for (int i = 0; i < not.Length; i++)
        {
            if (not[i] == '+' || not[i] == '-' || not[i] == '*' || not[i] == '/')
            {
                switch (not[i])
                {
                    case '+':
                        pilha.Push(pilha.Pop() + pilha.Pop());
                        break;
                    case '-':
                        double num = pilha.Pop();
                        double num2 = pilha.Pop();
                        pilha.Push(num2 - num);
                        break;
                    case '*':
                        pilha.Push(pilha.Pop() * pilha.Pop());
                        break;
                    case '/':
                        double num3 = pilha.Pop();
                        double num4 = pilha.Pop();
                        pilha.Push(num4 / num3);
                        break;
                    default:
                        break;
                }
            }
            else
            {
                Console.WriteLine(not[i]);
                pilha.Push(Double.Parse(not[i].ToString()));
            }
        }
    }
}

```

```

        Console.WriteLine(pilha.Peek());

        Console.ReadLine();
    }
}

```

Questão 2

```

namespace sequencia
{
    internal class Program
    {
        public static bool Seq(string seq)
        {
            Stack<char> pilha = new Stack<char>();
            for(int i = 0 ; i < seq.Length; i++)
            {
                if (seq[i] == '(' || seq[i] == '[')
                {
                    pilha.Push(seq[i]);
                }
                else if (seq[i] == ')')
                {
                    if(pilha.Peek() == '(')
                    {
                        pilha.Pop();
                    }
                    else
                    {
                        return false;
                    }
                }
                else if (seq[i] == ']')
                {
                    if (pilha.Peek() == '[')
                    {
                        pilha.Pop();
                    }
                    else
                    {
                        return false;
                    }
                }
            }
        }
    }
}

```

```

        return true;
    }
    static void Main(string[] args)
    {
        Console.WriteLine("Insira uma sequencia: ");
        string seq = Console.ReadLine();
        Console.WriteLine(Seq(seq));
        Console.ReadLine();
    }
}

```

Questão 2 - Classe Pilha

```

namespace Questão2Custom
{
    class Pilha
    {
        private char[] vetor;
        private int topo;
        public Pilha(int tamanho)
        {
            vetor = new char[tamanho];
            topo = 0;
        }
        public char Peek()
        {
            return vetor[topo - 1];
        }
        public void Push(char num)
        {
            if (topo >= vetor.Length)
            {
                Console.WriteLine("Tamanho máximo.");
            }
            else
            {
                vetor[topo] = num;
                topo++;
            }
        }
        public char Pop()
        {
            char num = vetor[topo - 1];
            vetor[topo - 1] = ' ';
        }
    }
}

```

```

        topo--;
        return num;
    }
    public void Clear()
    {
        for (int i = 0; i < vetor.Length; i++)
        {
            vetor[i] = ' ';
        }
    }
}

internal class Program
{
    public static bool Seq(string seq, int tamanho)
    {
        Pilha pilha = new Pilha(tamanho);
        for (int i = 0; i < seq.Length; i++)
        {
            if (seq[i] == '(' || seq[i] == '[')
            {
                pilha.Push(seq[i]);
            }
            else if (seq[i] == ')')
            {
                if (pilha.Peek() == '(')
                {
                    pilha.Pop();
                }
                else
                {
                    return false;
                }
            }
            else if (seq[i] == ']')
            {
                if (pilha.Peek() == '[')
                {
                    pilha.Pop();
                }
                else
                {
                    return false;
                }
            }
        }
    }
}

```

```
    }  
  }  
  return true;  
}  
static void Main(string[] args)  
{  
    Console.WriteLine("Insira uma sequencia e um tamanho: ");  
    string seq = Console.ReadLine();  
    int tamanho = int.Parse(Console.ReadLine());  
    Console.WriteLine(Seq(seq, tamanho));  
    Console.ReadLine();  
}  
}  
}
```