Questão 1 - Pilha

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;



namespace Questão1Flex
{
    class Dobro
    {
        private double dobroV;
        public double DobroV
        {
            get { return dobroV; }
            set { dobroV = value; }
        }
    }
    class Celula
    {
        Dobro dobroV;
        Celula proximo;
        public Celula(Dobro dobroV)
        {
            this.dobroV = dobroV;
            this.proximo = null;
        }
        public Celula()
        {
            dobroV = null;
            proximo = null;
        }
        public Dobro DobroV
        {
            get { return dobroV; }
            set { dobroV = value; }
        }
        public Celula Proximo
        {
            get { return proximo; }
            set { proximo = value; }
```

```
        }



    }
    class Pilha
    {
        public Celula topo;
        public Celula fundo;
        public Pilha()
        {
            Celula sentinela = new Celula();
            topo = sentinela;
            fundo = sentinela;
        }
        public void Empilhar(Dobro dobroV)
        {
            Celula novaCelula = new Celula(dobroV);
            novaCelula.Proximo = topo;
            topo = novaCelula;
            novaCelula = null;
        }
        public Dobro Desempilhar()
        {
            if (!(topo == fundo))
            {
                Dobro desempilhado = topo.DobroV;
                topo = topo.Proximo;
                return desempilhado;
            }
            else
            {
                throw new Exception("A pilha está vazia.");
            }
        }
        public Dobro Pico()
        {
            return topo.DobroV;
        }



    }
}
```

Questão 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;



namespace Questão1Flex
{
    class Teste
    {
        static void Main(string[] args)
        {
            Pilha pilha = new Pilha();
            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 '+':
                            double num = pilha.Desempilhar().DobroV;
                            double num2 = pilha.Desempilhar().DobroV;
                            Dobro num3 = new Dobro();
                            num3.DobroV = num + num2;
                            pilha.Empilhar(num3);
                            break;
                        case '-':
                            num = pilha.Desempilhar().DobroV;
                            num2 = pilha.Desempilhar().DobroV;
                            num3 = new Dobro();
                            num3.DobroV = num2 - num;
                            pilha.Empilhar(num3);
                            break;
                        case '*':
                            num = pilha.Desempilhar().DobroV;
                            num2 = pilha.Desempilhar().DobroV;
                            num3 = new Dobro();
                            num3.DobroV = num * num2;
```

```
                    pilha.Empilhar(num3);
                    break;
                case '/':
                    num = pilha.Desempilhar().DobroV;
                    num2 = pilha.Desempilhar().DobroV;
                    num3 = new Dobro();
                    num3.DobroV = num2 / num;
                    pilha.Empilhar(num3);
                    break;
                default:
                    break;
            }
        }
        else
        {
            Dobro numero = new Dobro();
            numero.DobroV = Char.GetNumericValue(not[i]);
            pilha.Empilhar(numero);
        }
    }
    Console.WriteLine(pilha.Pico().DobroV);
    Console.ReadLine();
  }
 }
}
```

Questão 2 - Pilha

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Questão2Flex
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;
```

```
namespace Questiones2
{
    class Linha
    {
        private char stringV;
        public char StringV
        {
            get { return stringV; }
            set { stringV = value; }
        }
    }
    class Celula
    {
        Linha stringV;
        Celula proximo;
        public Celula(Linha stringV)
        {
            this.stringV = stringV;
            this.proximo = null;
        }
        public Celula()
        {
            stringV = null;
            proximo = null;
        }
        public Linha StringV
        {
            get { return stringV; }
            set { stringV = value; }
        }
        public Celula Proximo
        {
            get { return proximo; }
            set { proximo = value; }
        }



    }
    class Pilha
    {
        public Celula topo;
        public Celula fundo;
        public Pilha()
```

```
        {
            Celula sentinela = new Celula();
            topo = sentinela;
            fundo = sentinela;
        }
        public void Empilhar(Linha stringV)
        {
            Celula novaCelula = new Celula(stringV);
            novaCelula.Proximo = topo;
            topo = novaCelula;
            novaCelula = null;
        }
        public Linha Desempilhar()
        {
            if (!(topo == fundo))
            {
                Linha desempilhado = topo.StringV;
                topo = topo.Proximo;
                return desempilhado;
            }
            else
            {
                throw new Exception("A pilha está vazia.");
            }
        }
        public Linha Pico()
        {
            return topo.StringV;
        }



    }
  }
}
```

Questão 2

```
using Questão2Flex.Questiones2;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

namespace Questão2Flex
{
    internal class teste
    {
        public static bool Seq(string seq)
        {
            Pilha pilha = new Pilha();
            for (int i = 0; i < seq.Length; i++)
            {
                Linha simb = new Linha();
                simb.StringV = seq[i];
                if (seq[i] == '(' || seq[i] == '[')
                {

                    pilha.Empilhar(simb);
                }
                else if (seq[i] == ')')
                {
                    if (pilha.Pico().StringV == '(')
                    {
                        pilha.Desempilhar();
                    }
                    else
                    {
                        return false;
                    }
                }
                else if (seq[i] == ']')
                {
                    if (pilha.Pico().StringV == '[')
                    {
                        pilha.Desempilhar();
                    }
                    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 3 (Não separei em parte já que o main são poucas linhas de código)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Questão3
{
    class Dobro
    {
        private int dobroV;
        public int DobroV
        {
            get { return dobroV; }
            set { dobroV = value; }
        }
    }
    class Celula
    {
        Dobro dobroV;
        Celula proximo;
        public Celula(Dobro dobroV)
        {
            this.dobroV = dobroV;
            this.proximo = null;
        }
        public Celula()
        {
            dobroV = null;
            proximo = null;
        }
        public Dobro DobroV
        {
```

```
        get { return dobroV; }
        set { dobroV = value; }
    }
    public Celula Proximo
    {
        get { return proximo; }
        set { proximo = value; }
    }



}
class Pilha
{
    public Celula topo;
    public Celula fundo;
    public Pilha()
    {
        Celula sentinela = new Celula();
        topo = sentinela;
        fundo = sentinela;
    }
    public void Empilhar(Dobro dobroV)
    {
        Celula novaCelula = new Celula(dobroV);
        novaCelula.Proximo = topo;
        topo = novaCelula;
        novaCelula = null;
    }
    public Dobro Desempilhar()
    {
        if (!(topo == fundo))
        {
            Dobro desempilhado = topo.DobroV;
            topo = topo.Proximo;
            return desempilhado;
        }
        else
        {
            throw new Exception("A pilha está vazia.");
        }
    }
    public Dobro Pico()
    {
```

```
            return topo.DobroV;
        }



    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Insira um número: ");
            int num = int.Parse(Console.ReadLine());
            Pilha pilha = new Pilha();
            while(num > 0)
            {
                Dobro numero = new Dobro();
                numero.DobroV = num % 8;
                pilha.Empilhar(numero);
                num /= 8;
            }
            while(pilha.Pico() != null)
            {
                Console.Write(pilha.Desempilhar().DobroV);
            }
            Console.ReadLine();

        }
    }
}
```