Questão01:

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

namespace Questao01
{
    internal class Questao01
    {
        static void Main(string[] args)
        {
            Stack<double> PilhaPolonesa = new Stack<double>();
            double num1, num2, resultado;
            char controle;
            do
            {

                do
                {
                    Console.WriteLine("Digite a expressão polonesa a ser calculada(Digite
C para calcular a expressão desejada:");
                    controle = char.Parse(Console.ReadLine());
                } while (controle != '0' && controle != '1' && controle != '2' && controle != '3'
&& controle != '4' && controle != '5' && controle != '6' && controle != '7' && controle !=
'8' && controle != '9' && controle != '*' && controle != '-' && controle != '+' && controle
!= '/' && controle != 'C' && controle != 'c');
                switch (controle)
                {
                    case '+':
                        num1 = PilhaPolonesa.Pop();
                        num2 = PilhaPolonesa.Pop();
                        resultado = num2 + num1;
                        PilhaPolonesa.Push(resultado);
                        break;
                    case '-':
                        num1 = PilhaPolonesa.Pop();
                        num2 = PilhaPolonesa.Pop();
                        resultado = num2 - num1;
                        PilhaPolonesa.Push(resultado);
                        break;
                    case '*':
                        num1 = PilhaPolonesa.Pop();
```

```
                num2 = PilhaPolonesa.Pop();
                resultado = num2 * num1;
                PilhaPolonesa.Push(resultado);
                break;
            case '/':
                num1 = PilhaPolonesa.Pop();
                num2 = PilhaPolonesa.Pop();
                resultado = num2 / num1;
                PilhaPolonesa.Push(resultado);
                break;
            default:
                if(controle != 'C' && controle != 'c')
                {
                    PilhaPolonesa.Push((double)Char.GetNumericValue(controle));
                }
                break;
        }


        } while (controle != 'C' && controle != 'c');
        Console.WriteLine("A expressão polonesa = "+ PilhaPolonesa.Pop());
        Console.ReadKey();
    }
  }
}
```

Questão01_02:

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

namespace Questão01._2
{
  internal class Questao01_02
  {
    static void Main(string[] args)
    {
      Pilha_Questao01_02 PilhaPolonesa;
      int tamanho;
      Console.WriteLine("Digite o tamanho da expressão: ");
      tamanho = int.Parse(Console.ReadLine());
```

```
        PilhaPolonesa = new Pilha_Questao01_02(tamanho);
        double num1, num2, resultado;
        char controle;
        for(int i=0; i < tamanho; i++)
        {

            do
            {
                Console.WriteLine("Digite a expressão polonesa a ser calculada(Digite
C para calcular a expressão desejada:");
                controle = char.Parse(Console.ReadLine());
            } while (controle != '0' && controle != '1' && controle != '2' && controle != '3'
&& controle != '4' && controle != '5' && controle != '6' && controle != '7' && controle !=
'8' && controle != '9' && controle != '*' && controle != '-' && controle != '+' && controle
!= '/');
            switch (controle)
            {
                case '+':
                    num1 = PilhaPolonesa.desempilhar();
                    num2 = PilhaPolonesa.desempilhar();
                    resultado = num2 + num1;
                    PilhaPolonesa.empilhar(resultado);
                    break;
                case '-':
                    num1 = PilhaPolonesa.desempilhar();
                    num2 = PilhaPolonesa.desempilhar();
                    resultado = num2 - num1;
                    PilhaPolonesa.empilhar(resultado);
                    break;
                case '*':
                    num1 = PilhaPolonesa.desempilhar();
                    num2 = PilhaPolonesa.desempilhar();
                    resultado = num2 * num1;
                    PilhaPolonesa.empilhar(resultado);
                    break;
                case '/':
                    num1 = PilhaPolonesa.desempilhar();
                    num2 = PilhaPolonesa.desempilhar();
                    resultado = num2 / num1;
                    PilhaPolonesa.empilhar(resultado);
                    break;
                default:
                      PilhaPolonesa.empilhar((double)Char.GetNumericValue(controle));
                    break;
```

```
            }
        }
        Console.WriteLine("A expressão polonesa = " +
PilhaPolonesa.desempilhar());
        Console.ReadKey();

    }
  }
}
```

Pilha_Questão01_02:

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

namespace Questão01._2
{
  internal class Pilha_Questao01_02
  {
    private double[] pilha;
    private int topo;

    public Pilha_Questao01_02(int tamanho)
    {
      pilha = new double[tamanho];
      topo = 0;
    }
    public Boolean pilhaCheia()
    {

      Boolean resp;

      if (topo == pilha.Length)
        resp = true;
      else
        resp = false;

      return resp;
    }

    public Boolean pilhaVazia()
    {
```

```
    Boolean resp;

    if (topo == 0)
        resp = true;
    else
        resp = false;

    return resp;
}
public double desempilhar()
{

    double desempilhado;

    if (!pilhaVazia())
    {
        topo--;
        desempilhado = pilha[topo];
        return desempilhado;
    }
    else
        throw new Exception("Não foi possível desempilhar: a pilha está vazia!");
}

public void empilhar(double novo)
{

    if (!pilhaCheia())
    {
        pilha[topo] = novo;
        topo++;
    }
    else
        throw new Exception("Não foi possível empilhar: a pilha está cheia!");
}

public double consultarTopo()
{

    if (!pilhaVazia())
    {
        return (pilha[topo - 1]);
    }
```

```
            else
                throw new Exception("Não foi possível consultar o elemento do topo da
pilha: a pilha está vazia!");
        }

    }
}
```

Questão02:

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

namespace Questao02
{
    internal class Questao02
    {
        static void Main(string[] args)
        {
            Stack<string> PilhaBemFormado = new Stack<string>();
            string controle;
            do
            {
                Console.WriteLine("Digite a sequência que desejar(Aperte a letra 'S' para
sair):");
                controle = Console.ReadLine();

                if (controle == "(" || controle == "[")
                {
                    PilhaBemFormado.Push(controle);
                }
                else if (controle == ")" && PilhaBemFormado.Count != 0)
                {
                    PilhaBemFormado.Pop();
                }
                else if (controle == "]" && PilhaBemFormado.Count != 0)
                {
                    PilhaBemFormado.Pop();
                }
                else if (controle == "]" && PilhaBemFormado.Count == 0)
                {
```

```
                PilhaBemFormado.Push(controle);
            }
            else if(controle == ")" && PilhaBemFormado.Count == 0)
            {
                PilhaBemFormado.Push(controle);
            }
            controle.ToLower();
        } while (controle != "s");
        if (PilhaBemFormado.Count == 0)
        {
            Console.WriteLine("A sequência está bem-formada!");
        }
        else
            Console.WriteLine("A sequência não está bem-formada!");
        Console.ReadKey();
    }
  }
}
```

Questão02_02:

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

namespace Questao02_02
{
  internal class Questao02_02
  {
    static void Main(string[] args)
    {
        Pilha_Questao02_02 PilhaBemFormado;
        int tamanho;
        string controle;
        Console.WriteLine("Digite o tamanho da sequência: ");
        tamanho = int.Parse(Console.ReadLine());
        PilhaBemFormado = new Pilha_Questao02_02(tamanho);
        for(int i = 0;i < tamanho;i++)
        {
            Console.WriteLine("Digite a sequência:");
            controle = Console.ReadLine();

            if (controle == "(" || controle == "[")
```

```
            {
                PilhaBemFormado.empilhar(controle);
            }
            else if (controle == ")" && !PilhaBemFormado.pilhaVazia())
            {
                PilhaBemFormado.desempilhar();
            }
            else if (controle == "]" && !PilhaBemFormado.pilhaVazia())
            {
                PilhaBemFormado.desempilhar();
            }
            else if (controle == "]" && PilhaBemFormado.pilhaVazia())
            {
                PilhaBemFormado.empilhar(controle);
            }
            else if (controle == ")" && PilhaBemFormado.pilhaVazia())
            {
                PilhaBemFormado.empilhar(controle);
            }
        }
        if (PilhaBemFormado.pilhaVazia())
        {
            Console.WriteLine("A sequência está bem-formada!");
        }
        else
            Console.WriteLine("A sequência não está bem-formada!");
        Console.ReadKey();

    }
  }
}
```

Pilha_Questão02_02:

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

namespace Questao02_02
{
  internal class Pilha_Questao02_02
```

```
{
    private string[] pilha;
    private int topo;

    public Pilha_Questao02_02(int tamanho)
    {
        pilha = new string[tamanho];
        topo = 0;
    }
    public Boolean pilhaCheia()
    {

        Boolean resp;

        if (topo == pilha.Length)
            resp = true;
        else
            resp = false;

        return resp;
    }

    public Boolean pilhaVazia()
    {

        Boolean resp;

        if (topo == 0)
            resp = true;
        else
            resp = false;

        return resp;
    }
    public string desempilhar()
    {

        string desempilhado;

        if (!pilhaVazia())
        {
            topo--;
            desempilhado = pilha[topo];
            return desempilhado;
```

```
        }
        else
            throw new Exception("Não foi possível desempilhar: a pilha está vazia!");
    }

    public void empilhar(string novo)
    {

        if (!pilhaCheia())
        {
            pilha[topo] = novo;
            topo++;
        }
        else
            throw new Exception("Não foi possível empilhar: a pilha está cheia!");
    }

    public string consultarTopo()
    {

        if (!pilhaVazia())
        {
            return (pilha[topo - 1]);
        }
        else
            throw new Exception("Não foi possível consultar o elemento do topo da
pilha: a pilha está vazia!");
    }

  }
}
```