

Samuel Garcia Porto Vieira
Código Aluno: 1522645

Questão 1:

using System;

```
namespace ex_1{
    class Program
    {
        public static void Main(string[] args)
        {
            int num = 0;
            int cont = 1;
            int salvacont = 1;
            int num2 = int.MinValue;
            do
            {
                Console.Write("Digite um número: ");
                num = int.Parse(Console.ReadLine());

                //Tratamento para evitar bugs de contagem
                if (num == -1)
                {
                    break;
                }

                if (num2 < num)
                {
                    cont++;
                }
                else
                {
                    cont = 1;
                }

                if (cont > salvacont)
                {
                    salvacont = cont;
                }
                num2 = num;
            } while (num != -1);

            Console.WriteLine("Maior sequência: " + salvacont);
        }
    }
}
```

Questão 2:

using System;

namespace ex_2

{

class Program

{

//Faça um programa que leia um número inteiro positivo N e imprima as N primeiras linhas do triângulo de Floyd.

public static void Main(string[] args)

{

Console.Write("Insira o número de termos: ");

int termo = int.Parse(Console.ReadLine());

int num = 1;

for (int i = 1; i <= termo; i++)

{

for (int j = 1; j <= i; j++)

{

Console.Write(num + " ");

num++;

}

//Espaço final das linhas

Console.WriteLine();

}

}

}

}

Questão 3:

using System;

namespace ex_3

{

class Program

{

static void Main(string[] args)

{

// Definir tamanho dos vetores

Console.Write("Informe o tamanho dos arrays: ");

int size = int.Parse(Console.ReadLine());

// Vetores iniciais

int[] vetX = new int[size];

int[] vetY = new int[size];

```

// Vetores solicitados
int[] vetSoma = new int[size];
int[] vetProduto = new int[size];
int[] vetDiferenca = new int[size];
int[] vetIntersecao = new int[size];

// Vetor União
int[] vetUniao = new int[size * 2];
int contUniao = 0;

// Inserção de valores nos vetores
for (int i = 0; i < size; i++)
{
    Console.WriteLine("Insira um valor no array X: ");
    vetX[i] = int.Parse(Console.ReadLine());

    Console.WriteLine("Insira um valor no array Y: ");
    vetY[i] = int.Parse(Console.ReadLine());

    Console.WriteLine();
}

// Processamento das operações
for (int i = 0; i < size; i++)
{
    vetSoma[i] = vetX[i] + vetY[i];
    vetProduto[i] = vetX[i] * vetY[i];
    vetDiferenca[i] = vetX[i] - vetY[i];

    // Interseção
    if (vetX[i] == vetY[i])
    {
        vetIntersecao[i] = vetX[i];
    }

    // União
    bool contX = false;
    bool contY = false;

    for (int j = 0; j < contUniao; j++)
    {
        if (vetUniao[j] == vetX[i])
        {
            contX = true;
        }
        if (vetUniao[j] == vetY[i])
        {
            contY = true;
        }
    }
}

```

```

    }
}

if (!contX)
{
    vetUniao[contUniao++] = vetX[i];
}
if (!contY)
{
    vetUniao[contUniao++] = vetY[i];
}
}

// Impressão de valores
for (int i = 0; i < size; i++)
{
    Console.WriteLine($"Soma: {vetSoma[i]}");
    Console.WriteLine($"Produto: {vetProduto[i]}");
    Console.WriteLine($"Diferença: {vetDiferenca[i]}");

    if (vetX[i] == vetY[i])
    {
        Console.WriteLine($"Interseção: {vetIntersecao[i]}");
    }
    Console.WriteLine();
}

// Impressão do array União
Console.WriteLine("União:");
for (int i = 0; i < contUniao; i++)
{
    Console.WriteLine(vetUniao[i]);
}

Console.ReadKey();

}
}
}

```

Questão 4:

using System;

namespace ex_4

```

{
    class Program
    {
        static void Main(string[] args)

```

```

{
    //Inserção inicial dos representantes e variáveis
    Console.WriteLine("Insira o numero de representantes: ");
    int num = int.Parse(Console.ReadLine());
    string[] nomes = new string[num];
    int[] votacao = new int[num];
    int maior = int.MinValue;
    int menor = int.MaxValue;
    int primeiro = -1;
    int ultimo = -1;
    int nulo = 0;

    //Preencher vetor representantes
    for (int i = 0; i < nomes.Length; i++)
    {
        Console.WriteLine($"Nome do representante {i}: ");
        string nome = Console.ReadLine();
        nomes[i] = nome;
    }

    //Processamento votos
    for(int i = 0; i < 60; i++)
    {
        Console.WriteLine($"Voto do aluno {i + 1}: ");
        int voto = int.Parse(Console.ReadLine());
        if (voto >= 0 && voto < num)
        {
            votacao[voto]++;
        }
        else
        {
            nulo++;
        }
    }

    //Conferir votos
    for (int i = 0; i < votacao.Length; i++)
    {
        if (votacao[i] > maior)
        {
            maior = votacao[i];
            primeiro = i;
        }

        if (votacao[i] < menor)
        {
            menor = votacao[i];
        }
    }
}

```

```

        ultimo = i;
    }
}
//Output
Console.WriteLine($"O candidato mais votado foi: {primeiro} com {maior}");
Console.WriteLine($"O candidato menos votado foi: {ultimo} com {menor}");
Console.WriteLine($"Quantidade de votos nulos: {nulo}");

    }
}
}

```

Questão 5:

```

using System;
using System.IO;
using System.Text;

```

```

namespace ex_5
{
    public class Jogador
    {
        private int numero;

        public int Numero
        {
            get { return numero; }
            set { numero = value; }
        }
        private string nome = "";

        public string Nome
        {
            get { return nome; }
            set { nome = value; }
        }
        private string posicao = "";

        public string Posicao
        {
            get { return posicao; }
            set { posicao = value; }
        }

        public Jogador(int numero, string nome, string posicao)
        {
            Numero = numero;
            Nome = nome;
            Posicao = posicao;
        }
    }
}

```

```

    }
}
public class Time
{
    //Atributos
    private string Nome { get; set; }
    private Jogador[] titulares;
    private Jogador[] reserva;
    private int quantTitulares;
    private int quantReserva;

    //Construtor
    public Time(string nome)
    {
        Nome = nome;
        titulares = new Jogador[11];
        reserva = new Jogador[12];
        quantTitulares = 0;
        quantReserva = 0;
    }

    //Metodos
    public bool AdicionarTitular(Jogador jogador)
    {
        if (quantTitulares < 11)
        {
            titulares[quantTitulares] = jogador;
            quantTitulares++;
            return true;
        }
        return false;
    }
    public bool AdicionarReserva(Jogador jogador)
    {
        if (quantReserva < 12)
        {
            reserva[quantReserva] = jogador;
            quantReserva++;
            return true;
        }
        return false;
    }
    public bool SubstituirTitular(Jogador jogador, string substituir)
    {
        for(int i = 0; i < quantTitulares; i++)
        {
            if (titulares[i].Nome == substituir)
            {

```

```

        titulares[i] = jogador;
        return true;
    }
}
return false;
}
public bool SubstituirReserva(Jogador jogador, string substituir)
{
    for (int i = 0; i < quantReserva; i++)
    {
        if (reserva[i].Nome == substituir)
        {
            reserva[i] = jogador;
            return true;
        }
    }
    return false;
}
public bool ConsultarTitular(string jogador)
{
    for(int i = 0; i < quantTitulares; i++)
    {
        if (titulares[i].Nome == jogador)
        {
            return true;
        }
    }
    return false;
}
public bool ConsultarReserva(string jogador)
{
    for (int i = 0; i < quantReserva; i++)
    {
        if (reserva[i].Nome == jogador)
        {
            return true;
        }
    }
    return false;
}
public bool ExcluirTitular(string jogador)
{
    for (int i = 0; i < quantTitulares; i++)
    {
        if (titulares[i].Nome == jogador)
        {
            for (int j = i; j < quantTitulares - 1; j++)
            {

```



```

        titulares[j] = titulares[j + 1];
    }
    titulares[quantTitulares - 1] = null;
    quantTitulares--;
    return true;
}
}
return false;
}
public bool ExcluirReserva(string jogador)
{
    for (int i = 0; i < quantReserva; i++)
    {
        if (reserva[i].Nome == jogador)
        {
            for (int j = i; j < quantReserva - 1; j++)
            {
                reserva[j] = reserva[j + 1];
            }
            reserva[quantReserva - 1] = null;
            quantReserva--;
            return true;
        }
    }
    return false;
}
public void GerarArq(string path)
{
    try
    {
        StreamWriter arquivo = new StreamWriter(path, false, Encoding.UTF8);
        using(arquivo)
        {
            arquivo.WriteLine("Titulares:");

            for (int i = 0; i < quantTitulares; i++)
            {
                Jogador titular = titulares[i];
                arquivo.WriteLine($"Número: {titular.Numero}, Nome: {titular.Nome},
Posição: {titular.Posicao}");
            }

            arquivo.WriteLine("\nReservas:");

            for (int i = 0; i < quantReserva; i++)
            {
                Jogador reserva = this.reserva[i];

```

```

        arquivo.WriteLine($"Número: {reserva.Numero}, Nome: {reserva.Nome},
Posição: {reserva.Posicao}");
    }
}
}
catch (Exception e)
{
    Console.WriteLine(e);
}
}
}
}
using System;
using System.IO;

namespace ex_5
{
    class Program
    {
        static void Main(string[] args)
        {
            //Instanciamento inicial
            Time time = new Time("Time Puc Minas");

            // Adicionar jogadores titulares
            time.AdicionarTitular(new Jogador(1, "Jogador 1", "Atacante"));
            time.AdicionarTitular(new Jogador(2, "Jogador 2", "Meio-campo"));
            time.AdicionarTitular(new Jogador(3, "Jogador 3", "Defensor"));
            time.AdicionarTitular(new Jogador(4, "Jogador 4", "Goleiro"));
            time.AdicionarTitular(new Jogador(5, "Jogador 5", "Atacante"));

            // Adicionar jogadores reservas
            time.AdicionarReserva(new Jogador(6, "Jogador 6", "Meio-campo"));
            time.AdicionarReserva(new Jogador(7, "Jogador 7", "Defensor"));
            time.AdicionarReserva(new Jogador(8, "Jogador 8", "Goleiro"));

            // Substituir um titular
            time.SubstituirTitular(new Jogador(9, "Jogador 9", "Atacante"), "Jogador 1");

            // Substituir um reserva
            time.SubstituirReserva(new Jogador(10, "Jogador 10", "Meio-campo"), "Jogador 6");

            // Consultar se um jogador é titular ou reserva
            bool titular = time.ConsultarTitular("Jogador 9");
            bool reserva = time.ConsultarReserva("Jogador 7");
            Console.WriteLine(titular);
            Console.WriteLine(reserva);
        }
    }
}

```

```

        // Excluir um titular
        bool tirar = time.ExcluirTitular("Jogador 5");
        Console.WriteLine(tirar);

        // Excluir um reserva
        bool tirarReserva = time.ExcluirReserva("Jogador 7");
        Console.WriteLine(tirarReserva);

        // Gerar o arquivo com as informações do time
        string path = "time.txt";
        time.GerarArq(path);
        Console.WriteLine($"Arquivo {path} gerado com sucesso.");
    }
}
}

```

Questão 6:

using System;

namespace ex_6

```

{
    internal class Livro
    {
        private string titulo;
        private string autores;
        private string editora;

        public string Editora
        {
            get { return editora; }
            set { editora = value; }
        }
        public string Autores
        {
            get { return autores; }
            set { autores = value; }
        }
        public string Titulo
        {
            get { return titulo; }
            set { titulo = value; }
        }
        public Livro(string titulo, string autores, string editora)
        {
            Titulo = titulo;
            Autores = autores;
            Editora = editora;
        }
    }
}

```

```

}
internal class Biblioteca
{
    private Livro[] acervo;
    private int numLivros;
    private const int MAXLIV = 50;

    public Biblioteca()
    {
        acervo = new Livro[MAXLIV];
        numLivros = 0;
    }

    public void AdicionarLivro(string titulo, string autores, string editora)
    {
        if (numLivros < MAXLIV)
        {
            acervo[numLivros].Titulo = titulo;
            acervo[numLivros].Autores = autores;
            acervo[numLivros].Editora = editora;
            numLivros++;
        }
    }

    public void AdicionarLivro(Livro livro)
    {
        if (numLivros < MAXLIV)
        {
            acervo[numLivros] = livro;
            numLivros++;
        }
    }

    public Livro Procurar(string titulo)
    {
        for (int i = 0; i < numLivros; i++)
        {
            if (acervo[i].Titulo == titulo)
            {
                return acervo[i];
            }
        }
        return null;
    }

    public void ListaNomes()
    {
        for (int i = 0; i < numLivros; i++)
        {
            Console.WriteLine(acervo[i].Titulo);
        }
    }
}

```

```

    }
    public string AcervoCompleto()
    {
        string lista = "";
        for(int i = 0; i < numLivros; i++)
        {
            lista += $"Título: {acervo[i].Titulo}\n";
        }
        return lista;
    }
    public int NumeroAcervo()
    {
        return numLivros;
    }
}
}

```

```
using System;
```

```
namespace ex_6
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            //Instanciamento do objeto
            Biblioteca biblioteca = new Biblioteca();

            //Adicionar livros
            biblioteca.AdicionarLivro(new Livro("O Senhor dos Anéis", "J.R.R. Tolkien",
"HarperCollins"));
            biblioteca.AdicionarLivro(new Livro("1984", "George Orwell", "Companhia das
Letras"));
            biblioteca.AdicionarLivro(new Livro("O Pequeno Príncipe", "Antoine de
Saint-Exupéry", "Agir"));
            biblioteca.AdicionarLivro(new Livro("Dom Quixote", "Miguel de Cervantes", "Penguin
Books"));

            //Buscar dados do livro
            string busca = "1984";
            Livro livro = biblioteca.Procurar(busca);

            if (livro != null)
            {
                Console.WriteLine($"Livro encontrado: \nTítulo: {livro.Titulo}\nAutor(es):
{livro.Autores}\nEditora: {livro.Editora}\n");
            }
            else

```

```

    {
        Console.WriteLine($"Livro com o título '{busca}' não encontrado.\n");
    }

    //Relação de livros
    int numero = biblioteca.NumeroAcervo();
    Console.WriteLine("Livros no acervo: " + numero);
    Console.WriteLine("Nome dos livros no acervo:");
    string lista = biblioteca.AcervoCompleto();
    Console.WriteLine(lista);
}
}
}

```

Questão 7:

```

using System;

namespace ex_7
{
    public class Estacionamento
    {
        //Propiedades
        private string nome;

        public string Nome
        {
            get { return nome; }
            set { nome = value; }
        }
        private int numVagasLivre;

        private string[] vagas;

        public string[] Vagas
        {
            get { return vagas; }
            set { vagas = value; }
        }

        //Construtor
        public Estacionamento(string nome,int numTotalVagas)
        {
            Nome = nome;
            numVagasLivre = numTotalVagas;
            Vagas = new string[numTotalVagas];
        }
    }
}

```

```

        //Propiedade do NumVagas
public int NumVagasLivre
{
    get { return numVagasLivre; }
}
public int Estacionar(string placa)
{
    for(int i = 0; i < vagas.Length; i++)
    {
        if (vagas[i] == null)
        {
            vagas[i] = placa;
            return i;
        }
    }
    return -1;
}
public int BuscarNumVaga(string placa)
{
    for(int i = 0; i < vagas.Length; i++)
    {
        if (vagas[i] == placa)
        {
            return i;
        }
    }
    return -1;
}
public void Retirar(string placa)
{
    for(int i = 0; i < vagas.Length; i++)
    {
        if (vagas[i] == placa)
        {
            vagas[i] = null;
        }
    }
}
public void ExibirOcupacao()
{
    for(int i = 0; i < vagas.Length; i++)
    {
        if (vagas[i] != null)
        {
            Console.WriteLine("Numero da vaga: " + i);
            Console.WriteLine("Numero da placa: " + vagas[i]);
        }
    }
}

```

```

    }
}

using System;

namespace ex_7
{
    class Program
    {
        static void Main(string[] args)
        {
            //Instanciamento
            Estacionamento estacionamento = new Estacionamento("Estacionamento Puc
Minas", 30);

            //Adicionando carros
            estacionamento.Estacionar("1111");
            estacionamento.Estacionar("2222");
            estacionamento.Estacionar("3333");
            estacionamento.Estacionar("4444");
            estacionamento.Estacionar("5555");

            //Primeira exibição
            Console.WriteLine("Exibição 1:");
            estacionamento.ExibirOcupacao();

            //Buscar placa
            string pesquisa = "3333";
            int numPlaca = estacionamento.BuscarNumVaga(pesquisa);
            Console.WriteLine($"A placa {pesquisa} foi encontrada na vaga: {numPlaca}");

            //Retirar carro
            estacionamento.Retirar("5555");

            //Exibir denovo o estacionamento
            Console.WriteLine("Exibição 2:");
            estacionamento.ExibirOcupacao();

            //Adicionando mais 3 carros
            estacionamento.Estacionar("6001");
            estacionamento.Estacionar("1112");
            estacionamento.Estacionar("1231");

            //Ultima exibição
            Console.WriteLine("Exibição 3:");
            estacionamento.ExibirOcupacao();

```



```
        //Vagas livres
        Console.WriteLine(estacionamento.NumVagasLivre);
    }
}
```