Exercícios Estrutura de Dados

Questão 1 –

/\* Faça um procedimento que receba um número n inteiro e positivo. O procedimento deve

imprimir todos os números do intervalo entre 0 e n que são divisíveis por 2 e por 3

(simultaneamente). Restrição: não utilize comandos de repetição. \*/

public class Program

{

    public static void Main(string[] args)

    {

        int numero = 50;

        RetornaDivisivel(numero, 0);

    }

    public static void RetornaDivisivel(int num, int i)

    {

        if(i <= num)

        {

            if(i % 2 == 0 && i % 3 == 0 && i != 0)

            {

                Console.Write(i + " ");

            }

            RetornaDivisivel(num, i+1);

        }

    }

}

Questão 2 –

/\* Faça uma função que receba um vetor de números inteiros por parâmetro e multiplique

por -1 todos os elementos negativos desse vetor. A função deve retornar o número de

elementos negativos encontrados. Restrição: não utilize comandos de repetição. \*/

public class Program

{

    public static void RetornaNegativo(int[] vetor)

    {

        RetornaNegativo(vetor, 0, 0);

    }

    public static int RetornaNegativo(int[] vetor, int contador, int i)

    {

        if(i < vetor.Length)

        {

            if(vetor[i] < 0)

            {

                vetor[i] \*= -1;

                contador ++;

            }

            return RetornaNegativo(vetor, contador, i+1);

        }

        return contador;

    }

    public static void Main(string[] args)

    {

        int[] vet = new int[] {5, -6, -2, 9, 8, 1, -7, 0, -3, 4};

        int negativos = RetornaNegativo(vet, 0, 0);

        Console.WriteLine("Quantidade de numeros negativos: {0}", negativos);

    }

}

Questão 3 –

/\* Faça um procedimento que receba o montante reservado para a aposentadoria de uma

pessoa e a quantia que ela pretende retirar por mês. O montante reservado é corrigido a

cada mês com a taxa de 0,55%, aplicada logo após a retirada. O procedimento deve

imprimir o montante (após a retirada e a correção), mês a mês. O procedimento também

deve imprimir o número de meses em que a pessoa conseguirá fazer a sua retirada.

Restrição: não utilize comandos de repetição. \*/

public class Program

{

    public static int ImprimeMontante(double montante, double retirada)

    {

        return ImprimeMontante(montante, retirada, 0);

    }

    public static int ImprimeMontante(double montante, double retirada, int contador)

    {

        if(montante >= retirada)

        {

            contador++;

            montante = (montante - retirada) + (montante \* 0.0055);

            Console.WriteLine("Valor após o {0}° mês: {1:n2}", contador, montante);

            return ImprimeMontante(montante, retirada, contador);

        }

        return contador;

    }

    public static void Main(string[] args)

    {

        double montante = 20000;

        double retirada = 4500;

        int meses = ImprimeMontante(montante, retirada);

Console.WriteLine("\nNumero de meses que foi possivel fazer a retirada: {0}", meses);

    }

}

Questão 4 –

/\* Faça uma função que ordene um vetor de inteiros passado por parâmetro utilizando o

método bolha e retorne o número de trocas realizadas. \*/

public class Program

{

    public static void Main(string[] args)

    {

        int[] vet = new int[] {7, 1, 5, 0, 9, 6, 3, 2, 8, 4};

        int trocas = RetornaTrocas(vet);

        Imprime(vet, trocas);

    }

    public static int RetornaTrocas(int[] vetor)

    {

        int aux;

        int contador = 0;

        bool houveTroca;

        do

        {

            houveTroca = false;

            for(int i = 0; i < vetor.Length-1; i++)

            {

                if(vetor[i] > vetor[i+1])

                {

                    aux = vetor[i];

                    vetor[i] = vetor[i+1];

                    vetor[i+1] = aux;

                    contador++;

                    houveTroca = true;

                }

            }

        }while(houveTroca);

        return contador;

    }

    public static void Imprime(int[] vet, int trocas)

    {

        Console.Write("Vetor ordenado: ");

        for(int i = 0; i < vet.Length; i++)

        {

            Console.Write(vet[i] + " ");

        }

        Console.WriteLine("\nNumero de trocas: {0}", trocas);

    }

}