

Curso: Programação Orientada a Objetos com C#

<https://www.udemy.com/programacao-orientada-a-objetos-csharp>

Prof. Nelio Alves

Lista de Exercícios – Revisão de Lógica de Programação

1) Um agricultor precisa aplicar veneno em suas plantações. Para isso ele precisa escolher um frasco e uma bomba para aplicação. O frasco de veneno possui um volume de veneno líquido medido em ml (mililitro). Já a bomba de aplicação possui uma vazão* medida em ml/s (mililitro por segundo). O agricultor deseja saber: dado o volume inicial de um frasco de veneno (em ml), a vazão da bomba (em ml/s), e um tempo (em s), qual volume de veneno restará no frasco após o uso da bomba durante este tempo?

* Vazão, neste contexto, significa o quanto a bomba consegue bombear em um certo intervalo de tempo.

Entrada:

A entrada possui três números inteiros representando respectivamente: o volume inicial de um frasco de veneno (em ml), a vazão da bomba (em ml/s), e um tempo (em s).

Saída:

A saída contém um número inteiro representando o volume de veneno (em ml) restante no frasco depois o uso da bomba durante o tempo informado. Se o volume do frasco não for suficiente, mostre a mensagem "nao eh possivel".

Exemplos:

Entrada	Saída
400 2 30	340
500 10 60	nao eh possivel

SOLUÇÃO:

```
using System;

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

            int volumeInicial, vazao, tempo, volumeFinal;

            volumeInicial = int.Parse(Console.ReadLine());
            vazao = int.Parse(Console.ReadLine());
            tempo = int.Parse(Console.ReadLine());

            if (vazao * tempo > volumeInicial) {
                Console.WriteLine("nao eh possivel");
            }
            else {
                volumeFinal = volumeInicial - (vazao * tempo);
                Console.WriteLine(volumeFinal);
            }

            Console.ReadLine();
        }
    }
}
```

2) Deseja-se fazer um programa para calcular a nota final de um estudante, considerando as três notas que ele tirou nos três trimestres de seu curso. Caso a nota final do estudante seja menor que 60.00, deve-se mostrar uma mensagem "DEPENDENCIA" também.

Entrada:

A entrada possui as três notas do estudante, com duas casas decimais.

Saída:

A saída possui a nota final do estudante com duas casas decimais. Caso a nota seja menor que 60.00, a saída também possui a mensagem "DEPENDENCIA".

Exemplos:

Entrada	Saída
22.00 30.10 25.50	77.60

Entrada	Saída
15.00 20.00 20.00	55.00 DEPENDENCIA

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            double N1, N2, N3, soma;

            string[] vet = Console.ReadLine().Split(' ');
            N1 = double.Parse(vet[0], CultureInfo.InvariantCulture);
            N2 = double.Parse(vet[1], CultureInfo.InvariantCulture);
            N3 = double.Parse(vet[2], CultureInfo.InvariantCulture);

            soma = N1 + N2 + N3;

            Console.WriteLine(soma.ToString("F2", CultureInfo.InvariantCulture));

            if (soma < 60.0) {
                Console.WriteLine("DEPENDENCIA");
            }

            Console.ReadLine();
        }
    }
}
```

3) Dadas as dimensões de uma foto e de um porta retrato, você deve verificar se uma foto cabe em um porta retrato. Você pode girar a foto 90° para adequá-la à dimensão do porta retrato.

Entrada:

A primeira linha da entrada contém a largura e altura da foto, e a segunda linha contém a largura e altura do porta retrato.

Saída:

A saída contém a palavra "SIM" se a foto couber no porta retrato, ou a palavra "NAO" caso contrário.

Exemplos:

Entrada	Saída
10 15 10 20	SIM

Entrada	Saída
10 15 20 10	SIM

Entrada	Saída
10 15 20 8	NAO

SOLUÇÃO:

```
using System;

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

            int fotoX, fotoY, prX, prY;
            string[] vet;

            vet = Console.ReadLine().Split(' ');
            fotoX = int.Parse(vet[0]);
            fotoY = int.Parse(vet[1]);

            vet = Console.ReadLine().Split(' ');
            prX = int.Parse(vet[0]);
            prY = int.Parse(vet[1]);

            if ((fotoX <= prX && fotoY <= prY) || (fotoY <= prX && fotoX <= prY)) {
                Console.WriteLine("SIM");
            }
            else {
                Console.WriteLine("NAO");
            }

            Console.ReadLine();
        }
    }
}
```

4) Fazer um programa para ler a quantidade de glicose no sangue de uma pessoa e depois mostrar na tela a classificação desta glicose de acordo com a tabela de referência ao lado.

Classificação	Glicose
Normal	Até 100 mg/dl
Elevado	Maior que 100 até 140 mg/dl
Diabetes	Maior de 140 mg/dl

Entrada

A entrada contém o valor de glicose com uma casa decimal.

Saída

A saída contém a classificação da glicose.

Exemplos:

Entrada	Saída
90.0	Normal

Entrada	Saída
140.0	Elevado

Entrada	Saída
143.2	Diabetes

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            double glicose;

            glicose = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            if (glicose <= 100.0) {
                Console.WriteLine("Normal");
            }
            else if (glicose <= 140.0) {
                Console.WriteLine("Elevado");
            }
            else {
                Console.WriteLine("Diabetes");
            }

            Console.ReadLine();
        }
    }
}
```

5) No arremesso de dardo, o atleta tem três chances para lançar o dardo à maior distância que conseguir. Você deve criar um programa para, dadas as medidas das três tentativas de lançamento, informar qual foi a maior.

Entrada

A entrada contém os valores das medidas em metros das três distâncias às quais o dardo foi arremessado, com duas casas decimais.

Saída

A saída contém a maior das distâncias, com duas casas decimais.

Exemplos:

Entrada	Saída
83.21 87.20 79.53	87.20

Entrada	Saída
87.20 87.20 79.53	87.20

Entrada	Saída
83.21 87.20 83.21	87.20

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            double x, y, z;

            string[] vet = Console.ReadLine().Split(' ');
            x = double.Parse(vet[0], CultureInfo.InvariantCulture);
            y = double.Parse(vet[1], CultureInfo.InvariantCulture);
            z = double.Parse(vet[2], CultureInfo.InvariantCulture);

            if (x > y && x > z) {
                Console.WriteLine(x.ToString("F2", CultureInfo.InvariantCulture));
            }
            else if (y > z) {
                Console.WriteLine(y.ToString("F2", CultureInfo.InvariantCulture));
            }
            else {
                Console.WriteLine(z.ToString("F2", CultureInfo.InvariantCulture));
            }

            Console.ReadLine();
        }
    }
}
```

6) Fazer um programa para ler dois números inteiros e mostrar a soma de todos os números ímpares entre eles, inclusive eles próprios.

Entrada:

A entrada consiste de dois números inteiros, sendo o primeiro menor ou igual ao segundo (ou seja: não precisa se preocupar em verificar qual dos dois números é o menor ou o maior).

Saída:

A saída consiste de um número inteiro representando a soma.

Exemplos:

Entrada	Saída
-10 2	-24

Entrada	Saída
3 14	48

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            int a, b, soma;

            string[] vet = Console.ReadLine().Split(' ');
            a = int.Parse(vet[0]);
            b = int.Parse(vet[1]);

            soma = 0;

            for (int i=a; i<=b; i++) {
                if (i % 2 != 0) { // testa se i é impar
                    soma = soma + i;
                }
            }

            Console.WriteLine(soma);
            Console.ReadLine();
        }
    }
}
```

7) Leia números inteiros até que um 0 (zero) seja lido. Para cada valor lido, o programa deve mostrar o quadrado do valor.

Lembrete: você não precisa ler todos dados para só depois começar a mostrar a saída. Você pode ir lendo um número e mostrando a saída referente a ele, um de cada vez.

Entrada:

Cada linha contém um número inteiro.

Saída:

Cada linha contém um número inteiro correspondente ao quadrado do número lido. Não se deve mostrar o quadrado para o número 0.

Exemplos:

Entrada	Saída
4	16
1	1
10	100
0	

Entrada	Saída
0	

SOLUÇÃO:

```
using System;

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

            int x, quadrado;

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

            while (x != 0) {
                quadrado = x * x;
                Console.WriteLine(quadrado);
                x = int.Parse(Console.ReadLine());
            }

            Console.ReadLine();
        }
    }
}
```

8) Leia três valores inteiros N, A e B. Em seguida, mostrar todos os múltiplos de N contidos no intervalo A e B (inclusive os próprios valores A e B).

Entrada:

A entrada é composta de três números inteiros N, A e B, respectivamente, sendo $A < B$ (não precisa se preocupar em verificar se A é menor que B).

Saída:

A saída contém os múltiplos de N entre A e B, sendo um valor por linha. Se não houver nenhum, não se deve imprimir nada.

Exemplos:

Entrada	Saída
5 10 33	10 15 20 25 30
8 2 7	

SOLUÇÃO:

```
using System;

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

            int N, A, B;

            N = int.Parse(Console.ReadLine());
            A = int.Parse(Console.ReadLine());
            B = int.Parse(Console.ReadLine());

            for (int i=A; i<=B; i++) {
                if (i % N == 0) { // testa se i é múltiplo de N
                    Console.WriteLine(i);
                }
            }

            Console.ReadLine();
        }
    }
}
```


9) Alexandre está se mordendo de vontade de jogar muito vídeo game depois das provas finais do primeiro trimestre. Entretanto, ele quer se prevenir de ficar sem bateria para o controle do vídeo game. Para isso, ele deseja se planejar, recarregando completamente a bateria do controle do vídeo game, e fazendo um acompanhamento de tempo a cada vez que for jogar. Assim, ele deseja informar quantas vezes pretende jogar, e quanto tempo vai jogar em cada vez. Você deve construir um programa que informa o tempo restante de cada vez que ele for jogar.

Entrada:

A entrada consiste do tempo total de duração da bateria completamente recarregada (em horas - ponto flutuante com uma casa decimal), seguido de um número inteiro N indicando quantas vezes Alexandre pretende jogar, seguido dos tempos que ele vai jogar em cada vez (em horas - ponto flutuante com uma casa decimal).

Saída:

A saída é composta do tempo restante de duração da bateria a cada vez que Alexandre for jogar (em horas - ponto flutuante com uma casa decimal). Se o tempo restante da bateria não for suficiente para uma determinada vez que Alexandre for jogar, mostrar a palavra "recarregar".

Exemplos:

Entrada	Saída
20.0	16.0
2	13.0
4.0	
3.0	

Entrada	Saída
10.0	6.0
4	2.0
4.0	recarregar
4.0	recarregar
4.0	
4.0	

Entrada	Saída
10.0	5.0
3	0.0
5.0	recarregar
5.0	
5.0	

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            double tempoTotal, tempoRestante, tempoJogado;
            int N;

            tempoTotal = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            N = int.Parse(Console.ReadLine());

            tempoRestante = tempoTotal;
            for (int i=0; i<N; i++) {
                tempoJogado = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
                tempoRestante = tempoRestante - tempoJogado;
                if (tempoRestante < 0.0) {
                    Console.WriteLine("recarregar");
                }
                else {
                    Console.WriteLine(tempoRestante);
                }
            }

            Console.ReadLine();
        }
    }
}
```

10) O curso de Computação Gráfica do IFTM é trimestral, sendo que as notas do ano são divididas em três partes: 30, 35 e 35, totalizando 100 pontos no ano. O professor de programação está querendo automatizar o processo de fechamento de notas, fazendo um programa para somar as notas dos três trimestres de todos alunos da turma.

Entrada:

A entrada é composta por um número inteiro N representando a quantidade de alunos da turma, seguida pelas três notas de cada aluno da turma. As notas de um aluno são representadas por três números de ponto flutuante com duas casas decimais.

Saída:

A saída mostra a nota final de cada aluno (duas casas decimais).

Exemplo:

Entrada	Saída
3	81.00
20.00 31.00 30.00	86.32
25.00 31.10 30.22	66.56
19.25 29.00 18.31	

SOLUÇÃO:

```
using System;
using System.Globalization;

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

            int N;
            double nota1, nota2, nota3, total;

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

            for (int i=0; i<N; i++) {
                string[] vet = Console.ReadLine().Split(' ');
                nota1 = double.Parse(vet[0], CultureInfo.InvariantCulture);
                nota2 = double.Parse(vet[1], CultureInfo.InvariantCulture);
                nota3 = double.Parse(vet[2], CultureInfo.InvariantCulture);
                total = nota1 + nota2 + nota3;
                Console.WriteLine(total.ToString("F2", CultureInfo.InvariantCulture));
            }

            Console.ReadLine();
        }
    }
}
```

11) Você foi contratado para fazer um programa que controla o volume de um equipamento de som. O volume do equipamento obedece a uma escala de 0 a 100. O equipamento inicialmente está em um certo volume. Em seguida, podem ser dados comandos de alteração de volume, representados por números inteiros (positivo = aumenta o volume; negativo = diminui o volume). A cada comando, o volume do equipamento deve ser ajustado. Se a medida de alteração do volume ultrapassar algum limite (0 ou 100), então o volume não deve passar do limite.

Entrada:

A entrada contém o volume inicial (V) do equipamento ($0 \leq V \leq 100$), seguido de vários comandos de alteração (CA) de volume ($-100 \leq CA \leq 100$). A entrada termina quando for entrado o valor 0 (zero) para o comando de alteração de volume. Os dados são números inteiros.

Saída:

A saída contém os volumes (números inteiros) do equipamento após cada comando de alteração de volume.

Exemplo:

Entrada	Saída
30	40
10	90
50	100
20	30
-70	35
5	0
-50	15
15	35
20	
0	

SOLUÇÃO:

```
using System;

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

            int volume, comando;

            volume = int.Parse(Console.ReadLine());
            comando = int.Parse(Console.ReadLine());

            while (comando != 0) {
                volume = volume + comando;
                if (volume < 0) {
                    volume = 0;
                }
                if (volume > 100) {
                    volume = 100;
                }
                Console.WriteLine(volume);
                comando = int.Parse(Console.ReadLine());
            }

            Console.ReadLine();
        }
    }
}
```