

LINGUAGEM DE PROGRAMAÇÃO 2

NOME: _____ TURMA: _____

EXERCÍCIO 01 – ANÁLISE DE CÓDIGO

Observe o código C# abaixo e responda o que é pedido:

```
using System;
namespace ConsoleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            int x;
            int y;
            int z;

            Console.WriteLine("Entre com um valor numéricos:");
            x = Convert.ToInt32(Console.ReadLine());

            y = x * 2;
            z = y + 10;
            x = z % 4;

            Console.WriteLine("O valor de x vale: ");
            Console.WriteLine(x);

            Console.WriteLine("O valor de y vale: ");
            Console.WriteLine(y);

            Console.WriteLine("O valor de z vale: ");
            Console.WriteLine(z);

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



- a) Sabendo que cada inteiro ocupa 32 bits de memória RAM, quantos bits foram armazenados para manter as variáveis deste programa? Explique como chegou à resposta.

- b) Suponha que o usuário tenha digitado o valor 2 e teclado ENTER durante a execução deste programa. Escreva **tudo** o que foi impresso na tela do Console para esta situação:

Para validar a sua resposta, escreva o programa no seu ambiente de desenvolvimento e confira o resultado.



EXERCÍCIO 02 – IMPLEMENTAÇÃO DE CÓDIGO

Dentro do contexto da Física Mecânica, temos definições como velocidade, aceleração, deslocamento, tempo decorrido, entre outras. A expressão denominada **equação horária da velocidade** é dada por

$$v = v_0 + a \cdot t$$

e descreve como que a velocidade de um objeto em movimento uniformemente variado (ou seja, que apresenta uma aceleração não nula) se modifica em função do tempo. Nesta, v representa a velocidade final, v_0 a velocidade inicial, a a aceleração e t o tempo decorrido.

Sabendo disso, implemente um programa em C# que dada a velocidade inicial de um objeto em movimento, sua aceleração e o tempo decorrido, imprima na tela qual a velocidade final. Observe um exemplo de entrada e saída:

Entrada:

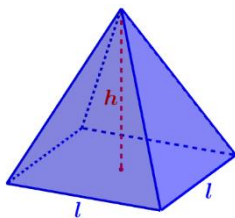
```
--- Calculadora de velocidade (m/s) ---  
Entre com a velocidade inicial (m/s):  
10  
Entre com a aceleração (m/s²):  
2  
Entre com o tempo decorrido(s):  
5
```

Saída:

```
A velocidade final (m/s) vale:  
20
```

EXERCÍCIO 03 – IMPLEMENTAÇÃO DE CÓDIGO

A Geometria é considerada uma das áreas mais importantes da Matemática. Conceitos como área e volume são vastamente explorados de acordo com diferentes formas geométricas. O volume da pirâmide depende diretamente da área da sua base e da sua altura. Considerando uma **pirâmide de base quadrada** escreva um programa que leia o lado e a altura da pirâmide e imprima na tela o seu volume. O seu volume é calculado através da seguinte relação:



$$volume = (\text{área da base}) * (\text{altura})/3$$

Entrada:

```
--- Calculadora de Volume (Pirâmide de Base Quadrada) ---  
Entre com o lado da base (m):  
3  
Entre com a altura (m):  
5
```

Saída:

```
O volume da pirâmide vale 15 m³.
```

EXERCÍCIO 04 – ANÁLISE DE ERROS

O Código a seguir apresenta 4 erros de compilação (aquele tipo de erro que não deixa o programa ser executado sem antes ter o código corrigido). É pedido que:

```
using System;
namespace ConsoleApp38
{
    class Program
    {
        static void Main(string[] args)
        {
            double bas;
            double alt;

            Console.WriteLine("Entre com a base do triângulo:")
            bas = Console.ReadLine();

            Console.WriteLine("Entre com a altura do triângulo:");
            alt = Convert.ToInt(Console.ReadLine());

            Console.WriteLine('O valor da área do triângulo vale:');
            Console.WriteLine(bas * alt / 2);

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

- Sublinhe as 4 linhas de código que apresentam os erros de compilação.
- Indique/descreva o motivo de cada um desses erros.

Para validar a sua resposta, escreva o programa no seu ambiente de desenvolvimento e observe os erros de compilação.

- Corrija o código para que ele possa ser compilado e executado adequadamente.