

TESTE 02 - LINGUAGEM DE PROGRAMAÇÃO 2 (5,0PTS)

NOME: _____ **TURMA:** DS _____ **DATA** _____

EXERCÍCIO 01 (1,5 PTS)

No estudo do Movimento Uniformemente Variado (MUV), a equação de Torricelli permite calcular a velocidade final de um corpo sem a necessidade do tempo.

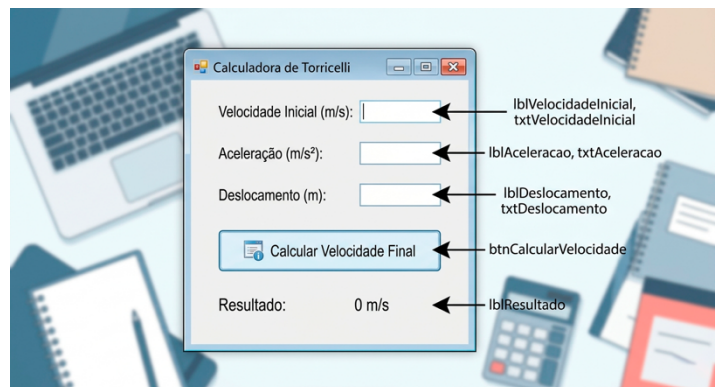
A fórmula é dada por:

$$v = \sqrt{v_0^2 + 2 \cdot a \cdot \Delta s}$$

Onde:

- v → velocidade final
- v_0 → velocidade inicial
- a → aceleração
- Δs → deslocamento

Com base na aplicação Windows Forms em C# abaixo, implemente o código que vai reagir ao clique do botão **btnCalcularVelocidade**. Esse código deverá calcular a velocidade final de um corpo móvel utilizando a fórmula de Torricelli. Caso o valor dentro da raiz quadrada seja negativo, ao invés de imprimir o resultado na label (**lblResultado**), escreva uma mensagem informando: “Erro”.



EXERCÍCIO 02 (3,5 PTS)

Os “Nexos Históricos” representam acontecimentos que alteraram significativamente o curso da humanidade. Cada evento possui um ano associado e ambos são armazenados em vetores paralelos, ou seja, a posição de um ano corresponde à posição do respectivo evento histórico.

Considere uma aplicação de Console C# e os vetores abaixo como exemplos:

```
int[] anos = {
    476, 1215, 1492, 1776, 1758,
    1789, 1914, 1945, 1969,
    1989, 1991, 2001, 2008,
    2020, 2022
};

string[] eventos = {
    "Queda do Império Romano do Ocidente",
    "Magna Carta",
    "Chegada de Colombo à América",
    "Independência dos Estados Unidos",
    "Publicação do Systema Naturae",
    "Revolução Francesa",
    "Início da Primeira Guerra Mundial",
    "Fim da Segunda Guerra Mundial",
    "Chegada do homem à Lua",
    "Queda do Muro de Berlim",
    "Fim da União Soviética",
    "Atentados de 11 de Setembro",
    "Crise Financeira Global",
    "Início da Pandemia de COVID-19",
    "Lançamento público do ChatGPT"
};
```

Para essa questão, considere que os dois vetores já estão preenchidos. Você não precisa lê-los ou criá-los. Mas atenção: **esses vetores servem apenas como exemplos**; sua solução deve ser generalista e saber lidar com qualquer par de vetores que siga o formato apresentado.

- a) (1,0 pt) Escreva um trecho de código em C# que **receba um século** (número inteiro) informado pelo usuário no Console e **imprima todos os eventos históricos que ocorreram naquele século**.

Exemplo:

- Século 20 → anos entre 1901 e 2000
- Século 18 → anos entre 1701 e 1800

Caso nenhum evento pertença ao século informado, exiba a mensagem: **“Nenhum evento encontrado para este século.”**

<u>Exemplo de entrada e saída esperada:</u> Entrada: 18 Saída: 1789 - Revolução Francesa 1758 - Publicação do Systema Naturae 1776 - Independência dos Estados Unidos	<u>Exemplo de entrada e saída esperada:</u> Entrada: 24 Saída: Nenhum evento encontrado para este século.
---	---

- b) (1,0 pt) Escreva um trecho de código que encontre o evento histórico mais antigo armazenado nos vetores. Seu algoritmo deverá percorrer todo o vetor para encontrar o menor ano. Ao final, imprima o ano e o nome do evento correspondente.

Exemplo de saída esperada:
Saída:
476 - Queda do Império Romano do Ocidente

c) (1,0 pt) Os historiadores dividiram os eventos em quatro grandes épocas históricas:

Época Histórica	Intervalo de anos
Idade Antiga	até 476
Idade Média	477 até 1453
Idade Moderna	1454 até 1789
Idade Contemporânea	1790 em diante

Escreva um programa que irá imprimir na tela a quantidade de eventos por época histórica seguindo formato abaixo.

Exemplo de saída esperada:

Idade Antiga: 1 evento(s)

Idade Média: 1 evento(s)

Idade Moderna: 4 evento(s)

Idade Contemporânea: 9 evento(s)

Nota: é pedido que essa contagem seja realizada em um terceiro vetor obrigatoriamente.

d) (0,5 pt) Escreva um programa que faça uso do vetor anterior para imprimir na tela qual época apresenta a maior quantidade de eventos. Se houver empate, imprima a primeira época encontrada com a maior quantidade.

Exemplo de saída esperada:

Idade Contemporânea

