

INSTRUÇÕES:

- Site para envio: <https://github.com>
- Instruções para envio: Google Classroom
- Nome da lista (pasta no github): lista01
 - Enviar ex1.cpp, ..., exN.cpp, cada um com a solução do N exercícios correspondentes.
- Siga **estritamente** o formato da saída solicitada no enunciado.
- Permitido somente **conteúdo aprendido em aula ou em listas anteriores** a não ser que enunciado explicitamente contrário
- **Plágio será severamente punido**
- Dúvidas através de todos os meios de comunicação disponibilizados
- Não se preocupe com a precisão das respostas **numéricas não inteiras**. **Erros numéricos inferiores a 1% serão desconsiderados**, a não ser que o enunciado indique outra precisão desejada.
- Apenas bibliotecas padrões de C/C++ serão permitidas.

LISTA PRÁTICA

1. (0.5 pontos) Dizem por aí que a primeira coisa que precisamos fazer quando vamos aprender a programar é fazer um programa que escreva somente “Hello World!” (sem aspas) na tela (que significa “Olá Mundo!”), senão teremos azar durante toda a vida de programador =D. Por via das dúvidas, melhor obedecer...

Escreva um programa que escreva a expressão citada na tela, lembrando que para isso precisamos usar o **std::cout** ou o **printf**

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|---------|--------------|
| | Hello World! |

2. (0.5 pontos) Faça um programa que leia, nessa ordem, o número de balas, o preço (único) de cada bala, o número de chocolates e o preço (único) de cada chocolate e escreva na tela o valor total da compra.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|---------|-------|
| 20 | 4.0 |
| 0.05 | |
| 1 | |
| 3.00 | |

3. (0.5 pontos) Faça um programa que receba um número inteiro **n** do usuário e escreva os números de 0 a 2n inclusive, separados por vírgula e sem vírgula depois do último número.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|---------|---------------------|
| 3 | 0, 1, 2, 3, 4, 5, 6 |

4. (0.5 pontos) Faça um programa que leia um número inteiro **n** digitado pelo usuário imprima todos os números de 1 a **n**, inclusive, um por linha, que sejam múltiplos simultaneamente de 3, 5, 7, 11 e 17.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|---------|---|
| 100000 | 19635 39270 58905 78540 98175 |

5. (1 ponto) Faça um programa que receba vários (no máximo 1000000) números inteiros positivos do usuário e imprime o número que restará na lista ao repetir o seguinte processo até restar no máximo um elemento (é garantido que isso ocorrerá na correção):

- Escolha um elemento da lista ao acaso
- Se houver outro igual a ele, remova os dois

Se não sobrar nenhum elemento imprima 0.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|---------------------------------|-------|
| 5 1 2 3 5 1 3 | 2 |

6. (1 ponto) Faça uma função com o seguinte cabeçalho:

```
int soma(double a, double b);
```

Ela deve receber dois valores e retornar a parte inteira da soma.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|--------------|-------|
| soma(1, 2.5) | 3 |

7. (1 ponto) Faça uma função com o seguinte cabeçalho:

```
int seq(int n, int k, int *ini);
```

Que calcula o n -ésimo ($n \in \mathbb{N}$) termo da seguinte sequência recursiva, com **n** partindo de zero:

$$a_n = \begin{cases} \sum_{i=n-k}^{n-1} a_i^2 & n \geq k \\ ini[n] & n < k \end{cases}$$

Onde **ini** é um vetor dos k primeiros termos da sequência.

Bibliotecas permitidas: stdio.h, iostream

EXEMPLO

| Entrada | Saída |
|-----------------------|-------|
| seq(3, 3, {1, -2, 1}) | 6 |

8. (1 ponto) Faça uma função com o seguinte cabeçalho:

```
std::string conteudo(std::string nome);
```

Que retorne o conteúdo do arquivo chamado **nome**. Não esqueça de fechar o arquivo! Assuma que o arquivo terá no máximo um milhão de caracteres.

Bibliotecas permitidas: stdio.h, iostream, string.h, fstream, string

EXEMPLO

| Entrada | Saída |
|--|---------|
| ARQUIVO arquivo.txt: Teste CHAMADA: conteudo("arquivo.txt") | "Teste" |

9. (1 ponto) Para calcular a distância entre os pontos P_1 e P_2 dadas suas coordenadas, podemos usar a seguinte expressão:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Leia nessa ordem x_1, y_1, x_2, y_2 , todos **double**, e imprima na tela o resultado com 4 casas decimais.

Bibliotecas permitidas: stdio.h, iostream, iomanip, math.h

EXEMPLO

| Entrada | Saída |
|---------|--------|
| 0.0 | 1.4142 |
| 0.0 | |
| 1.0 | |
| 1.0 | |

10. (1 ponto) Faça um programa que receba o valor de uma compra e o valor em dinheiro pago, nessa ordem. O programa deve escrever o valor do troco na primeira linha seguido da forma de dar o troco que gaste o menor número possível total de notas e moedas, descrevendo através de 12 linhas onde cada linha indica o número de unidades daquela cédula ou moeda.

A forma que gasta sempre o menor número total de notas e moedas, no caso do Brasil, é sempre usar a maior nota ou moeda disponível para o valor. Vejamos o exemplo dado:

- R\$ 99,49 → maior nota menor ou igual a esse valor é R\$ 50,00, então até agora vamos usar 1 nota de R\$ 50,00 restando mais R\$ 49,49
- R\$ 49,49 → maior nota menor ou igual a esse valor é R\$ 20,00, então até agora vamos usar 1 nota de R\$ 50,00 e 1 nota de R\$ 20,00 restando mais R\$ 29,49
- R\$ 29,49 → maior nota menor ou igual a esse valor é R\$ 20,00, então até agora vamos usar 1 nota de R\$ 50,00 e 2 notas de R\$ 20,00 restando mais R\$ 9,49
- R\$ 9,49 → maior nota menor ou igual a esse valor é R\$ 5,00, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00 e 1 nota de R\$ 5,00 restando mais R\$ 4,49
- R\$ 4,49 → maior nota menor ou igual a esse valor é R\$ 2,00, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00 e 1 nota de R\$ 2,00 restando mais R\$ 2,49
- R\$ 2,49 → maior nota menor ou igual a esse valor é R\$ 2,00, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00 e 2 notas de R\$ 2,00 restando mais R\$ 0,49
- R\$ 0,49 → maior moeda menor ou igual a esse valor é R\$ 0,25, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00 e 1 moeda de R\$ 0,25 restando mais R\$ 0,24
- R\$ 0,24 → maior moeda menor ou igual a esse valor é R\$ 0,10, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25 e 1 moeda de R\$ 0,10 restando mais R\$ 0,14
- R\$ 0,14 → maior moeda menor ou igual a esse valor é R\$ 0,10, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25 e 2 moedas de R\$ 0,10 restando mais R\$ 0,04
- R\$ 0,04 → maior moeda menor ou igual a esse valor é R\$ 0,01, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25, 2 moedas de R\$ 0,10 e 1 moeda de R\$ 0,01 restando mais R\$ 0,03
- R\$ 0,03 → maior moeda menor ou igual a esse valor é R\$ 0,01, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25, 2 moedas de R\$ 0,10 e 2 moedas de R\$ 0,01 restando mais R\$ 0,02
- R\$ 0,02 → maior moeda menor ou igual a esse valor é R\$ 0,01, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25, 2 moedas de R\$ 0,10 e 3 moedas de R\$ 0,01 restando mais R\$ 0,01
- R\$ 0,01 → maior moeda menor ou igual a esse valor é R\$ 0,01, então até agora vamos usar 1 nota de R\$ 50,00, 2 notas de R\$ 20,00, 1 nota de R\$ 5,00, 2 notas de R\$ 2,00, 1 moeda de R\$ 0,25, 2 moedas de R\$ 0,10 e 4 moedas de R\$ 0,01 finalizando o cálculo do troco

DICA: Nesse exercício, lembre-se de que cálculos com números inteiros são mais precisos, de forma que usar **double** para muitos cálculos consecutivos pode levar a erros de precisão.

Bibliotecas permitidas: `stdio.h`, `iostream`, `iomanip`

EXEMPLO

| Entrada | Saída |
|---------|------------------|
| 400.51 | Troco: R\$ 99.49 |
| 500.00 | 0 de R\$ 200.00 |
| | 0 de R\$ 100.00 |
| | 1 de R\$ 50.00 |
| | 2 de R\$ 20.00 |
| | 0 de R\$ 10.00 |
| | 1 de R\$ 5.00 |
| | 2 de R\$ 2.00 |
| | 0 de R\$ 1.00 |
| | 0 de R\$ 0.50 |
| | 1 de R\$ 0.25 |
| | 2 de R\$ 0.10 |
| | 0 de R\$ 0.05 |
| | 4 de R\$ 0.01 |

11. (1 ponto) Escreva uma função

```
std::string raizes(double a, double b, double c);
```

que leia os parâmetros a, b e c, nessa ordem e resolva a seguinte equação no conjunto dos reais:

$$ax^2 + bx + c = 0$$

Lembrando que a fórmula para as raízes de uma equação de segundo grau é dada por:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- No caso de haver duas raízes reais distintas, por exemplo 2.0 e 1.0, retorne a string em ordem da menor para a maior: "1.0 e 2.0"
- No caso de haver uma raiz real, por exemplo 1.0, retorne a string apenas com a raiz: "1.0"
- No caso de não haver raiz real, retorne a string apenas com a palavra "Nenhuma"
- No caso de haver infinitas raízes reais retorne a string: "Infinitas"

Em nenhuma das situações há exigência de número de casas decimais.

Para o cálculo de raiz quadrada, lembre-se de incluir a biblioteca "math.h".

Lembre-se de considerar todos os possíveis valores dos parâmetros em sua solução.

Bibliotecas permitidas: stdio.h, iostream, sstream, string, math.h

EXEMPLO

| Entrada | Saída |
|-----------------|--------|
| raizes(1, 2, 1) | "-1.0" |

12. (1 ponto) Escreva uma função

```
int pow(int b, int e, int m);
```

que calcule o resto da divisão por **m** de **b** elevado a **e**. Nesse exercício é proibido o uso de qualquer biblioteca além da `stdio.h`, `iostream`.

EXEMPLO

| Entrada | Saída |
|-----------------|-------|
| pow(2, 10, 100) | 24 |