

**Orientações:** Nosso cérebro não se encontra na ponta dos nossos dedos, dito isso e lembrando de nossa 1ª e 2ª aulas, sugiro:

Ao fazer um dos nossos exercícios de programação, primeiro faça uma análise prévia:

- O que o programa pede, qual o objetivo?
- identifique as variáveis de entrada e seus tipos de dados, dê nomes coerentes a elas.
- qual o processamento? o que será feito com as variáveis de entrada, cálculos, etc.
- identifique a(s) variáveis a serem apresentadas na saída, qual o tipo? Qual a forma de apresentação?

- Utilize o *template* sugerido, que está a seguir:

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
void main(void)
{
    setlocale(LC_ALL, "Portuguese");

}
```

Após essa breve análise, só então passe para a codificação.

Como última recomendação, sugiro recorrerem aos livros sugeridos, aos slides, a min, ou ainda aos colegas, mas não consultem internet ou Chat GPT. Nessas duas últimas opções certamente encontrarão as respostas, mas esses programas iniciais, servem para que desenvolvam um raciocínio objetivo e preciso no que se refere à programação e sedimentem a sintaxe da linguagem e desenvolvam sua capacidade de depuração ("debugar") os programas

Os exercícios constantes da próxima folha se referem somente a execução em sequência, assim como fizemos em sala de aula. Para dar nome aos seus arquivos façam da seguinte forma:

**Legenda:**

**ex\_nn\_2\_sel\_t#\_rrrrrrr\_nome.c**, onde:

**nn** – número do exercício

**#** - a, para a turma a e assim sucessivamente.

**rrrrrrr** – RA do aluno, somente números.

**nome** – nome do aluno\_primeirosobrenome.

**Tudo em minúsculo, não serão aceitos exercícios realizados de outra forma!**

**Exemplo:** Aluno Tony Stark, da turma C que vai dar nome ao exercício 1

**ex\_01\_2\_sel\_tc\_12345678\_tony\_stark.c**

Após realizar todos os exercícios da folha, os compacte em formato zip e enviem para o meu e-mail ([miguel.archanjo@ceub.edu.br](mailto:miguel.archanjo@ceub.edu.br)), a data de entrega será informada em mensagem específica para a turma, o arquivo compactado deverá ter o seguinte nome:

**t#\_sel\_rrrrrrr\_nome.zip**, seguindo a mesma legenda acima, no caso do aluno Tony Stark acima, seria: **tc\_sel\_12345678\_tony\_stark.zip**

## Exercícios de Seleção

1. Faça um programa que leia o ano de nascimento de uma pessoa e calcule sua idade. Após isso verifique se ela já tem idade para votar (16 anos ou mais). mostre a mensagem informando a situação dela:

- a) A idade é xx anos e já pode votar.
- b) A idade é xx anos e não pode votar.

### ALTERAÇÕES:

- a. Na tela de saída, mostre também a data de nascimento.
- b. Mostre também a idade da pessoa.

2. Faça um programa que leia dois valores quaisquer e mostre o maior deles ou mostre a mensagem "Os valores são iguais."

### TESTE:

Teste 1: Entrada: 5 e 10	Saída: O maior valor é 10
Teste 2: Entrada: 10 e 5	Saída: O maior valor é 10
Teste 3: Entrada: 5 e 5	Saída: Os valores são iguais.

### ALTERAÇÕES:

- a. Se eles forem diferentes, mostre os valores digitados na ordem decrescente.
  - b. Se eles forem iguais, mostre a mensagem e o valor digitado.
3. Elabore o programa que leia um número qualquer e verifique se ele é positivo, nulo ou negativo.

Teste 1: número = 4	Saída: Número Positivo
Teste 2: número = 0	Saída: Número Nulo
Teste 3: número = -4	Saída: Número Negativo

### ALTERAÇÕES:

- a. Além da mensagem, mostre também o número digitado pelo usuário.
- b. Se o número for positivo, mostre a mensagem, o valor da variável e o dobro de seu valor;  
Se negativo, mostre a mensagem, o valor da variável e o seu triplo de seu valor;  
Se nulo, mostre a mensagem, o valor da variável número.

4. Construa o programa que calcule o peso ideal de uma pessoa.

Utilize as seguintes fórmulas:

- Se homem, o peso ideal é calculado assim:  $(72,7 * altura) - 58$ ;
- Se mulher, o peso ideal é calculado assim:  $(62,1 * altura) - 44,7$ .

Análise o problema e verifique quais são os dados que o usuário precisa fornecer (digitar).

Teste 1: Entrada: altura = 1.70 e gênero = 1 Saída: peso ideal = 65.59 Kg

Teste 2: Entrada: altura = 1.70 e gênero = 2 Saída: peso ideal = 60.8699999 Kg

#### ALTERAÇÕES:

a. Mostre o peso com duas casas decimais.

c. Troque a entrada para 'm' ou 'f'.

Teste 3: altura = 1.8      genero = m      Saída: peso ideal = 72.86 Kg

d. Mostre uma mensagem de erro se ele digitar valor de gênero diferente de 'm' ou 'f'.

5. Elabore o programa que verifica se o valor inteiro fornecido pelo usuário é par ou ímpar. Analise o problema e verifique quais são os dados que o usuário precisa fornecer.

6. Analise o resultado de uma transação comercial. Verifique a situação final do comerciante trabalhando com os valores lidos, ou seja, o preço de compra e o preço de venda. Gere a tela de saída com uma das seguintes mensagens:

“Teve lucro.”, “Teve prejuízo.” ou “Os valores são iguais.”.

Utilize os valores abaixo para testar seu programa:

Teste 1: Entrada: compra = 1000, venda = 1200      Saída: Teve lucro.

Teste 2: Entrada: compra = 1200, venda = 1000      Saída: Teve prejuízo.

Teste 3: Entrada: compra = 1000, venda = 1000      Saída: Os valores são iguais.

Alterações: Na saída, mostre também o valor do preço de compra e do preço de venda.

7. Refaça o programa que calcule a média aritmética de um aluno que realizou duas avaliações. Além do valor da média, inclua na tela de saída uma das mensagens: “Aluno aprovado.” ou “Aluno reprovado.”. Considere que o aluno será aprovado com a média maior ou igual a cinco.

Teste 1: Entrada: nota1 = 5, nota2 = 6      Saída: Média = 5.5      Aluno aprovado.

Teste 2: Entrada: nota1 = 5, nota2 = 2      Saída: Média = 3.5      Aluno reprovado.

#### ALTERAÇÕES:

a. Mostre o valor da média aritmética com duas casas decimais.

b. Altere a saída. Mostre a média e a mensagem na mesma linha:      Média do aluno: x.xxx  
Aluno aprovado ou Aluno reprovado.

c. Refaça-o considerando que a primeira prova tem peso três e a segunda, peso cinco. Ou seja, calcule a média ponderada do aluno.

Teste 3: nota1 = 5, nota2=6, peso1=3, peso2=5      Saída: média = 5,625

d. Deixe o programa mais interessante, permita que o usuário digite o valor dos pesos para usar no cálculo da média ponderada.

8. Projete o programa que leia um valor numérico e verifique se ele é maior ou igual a 100. Mostre uma das mensagens: “Valor maior ou igual a cem.” Ou “Valor menor que cem.”

Teste 1: valor = 200                      Saída: Valor maior ou igual a cem.

Teste 2: valor = 20                      Saída: Valor menor que cem.

Teste 3: valor = 100                      Saída: Valor maior ou igual a cem.

#### ALTERAÇÕES:

a. Mostrar também o valor numérico lido, na tela de saída.

9. Projete o programa que calcule as raízes de uma equação do 2º grau, levando em consideração a análise da existência de raízes reais. Se o valor de delta for menor que zero, não existem raízes reais; se delta for igual a zero, existem duas raízes iguais; se delta for maior que zero, existem duas raízes diferentes.

Expressão:  $ax^2 + bx + c = 0$

$x = (-b \pm \sqrt{\Delta})/2a$

$\Delta = (b^2 - 4ac)$

Observação: inclua a biblioteca math.h nas declarações de pré-processamento e utilize a função sqrt, por exemplo:

```
float x=4, raizq;
```

```
raizq = sqrt(x);
```

A função pow recebe a base e o expoente e retorna o resultado, veja o exemplo:

```
int x=2, y=2, pot;
```

```
pot = pow(x,y);
```