

**Orientações gerais:**

- Os exercícios devem ser feitos individualmente, ou seja, exercícios iguais serão zerados;
- As dúvidas serão esclarecidas nos laboratórios;
- Os exercícios deverão ser entregues via Moodle;
- **Os executáveis não devem ser enviados, apenas os arquivos de extensão “.c”. Todos devem ser zipados em um arquivo chamado “lista1.zip”.**

**Primeiros programas e Estruturas básicas**

**Orientações:**

a) O objetivo dos primeiros exercícios é a familiarização com o ambiente de programação e um primeiro contato com a linguagem C e algumas de suas características, como o fato de todas as linhas terminarem com ponto-e-vírgula.

b) Usaremos, nos exercícios abaixo, o conceito de variáveis. Variáveis em C se comportam de forma distinta de variáveis da matemática, mas inicialmente podemos supor que são similares. Para usar uma variável precisamos declará-la, para que o computador saiba quanto espaço de memória deve ser alocado, de acordo com o tipo da variável. Para declarar basta dizer seu tipo e o nome escolhido. Alguns exemplos:

- `float x;`
  - Neste caso declaramos uma variável de nome “x”, do tipo float;
- `double y;`
  - Similar ao caso anterior, declaramos uma variável de nome “y”, do tipo double;
- `float a, b, c;`
  - Desta vez, estamos usando uma única linha para declarar três variáveis do tipo float, chamadas “a”, “b” e “c”.

c) Algumas operações matemáticas úteis são:

- Soma:  $a + b$
- Subtração:  $a - b$
- Multiplicação:  $a * b$
- Divisão:  $a / b$
- Raiz quadrada: `sqrt(a)`
- Seno: `sin(a)`

Para usar algumas funções matemáticas, pode ser necessário o uso da seguinte linha no cabeçalho de seu programa: `#include <math.h>`

d) Para usar funções de entrada e saída de dados é necessário incluir o cabeçalho:

`#include<stdio.h>`

e) Para atribuir um valor a uma variável, usamos o operador de atribuição “=”. Neste caso, a variável onde o valor será atribuído deve estar à esquerda do símbolo “=”.

Exemplos:

- `x = 10;`
  - Guarda o valor 10 na variável x;
- `y = x;`
  - Neste caso o valor armazenado na variável x, qualquer que seja, será copiado para a variável y;
- `z = x + y*y;`

- Neste caso, o valor de  $z$  será o valor de  $x$  somado ao quadrado de  $y$ . Caso os três comandos acima sejam executados nesta ordem, então  $z$  será 110.

f) A função `printf()` é usada para a impressão de dados na tela. Por exemplo, `printf("Oi")` imprimirá a mensagem "Oi" na tela.

- Em determinados casos, para imprimir números, precisamos fornecer informações adicionais à função `printf`. Para cada símbolo `%` (porcentagem) o comando `printf` lerá o conteúdo de uma variável e o escreverá na tela.
- Ao imprimir um dado tipo "float" por exemplo usamos `printf("%.2f", x)`, para imprimir o valor de  $x$  com duas casas decimais.
- Para dados do tipo "double" usamos `printf("%lf", x)`.

g) A função `scanf()` é usada para a leitura de dados da tela. Por exemplo, `scanf("%d", &n)` irá armazenar um valor inteiro no endereço de memória da variável  $n$ . Lembre-se de adicionar o caracter `&` precedendo a variável de armazenamento, referindo-se ao seu endereço na memória.

## Exercícios

1) Escreva um código para calcular o seno de um valor em graus. O valor deve ser lido do usuário usando a função `scanf` e seu seno deve ser impresso usando a função `printf`. Compile o código, verificando se algum erro ocorreu. Execute o código com diversos valores de ângulos e verifique se o resultado apresentado está correto.

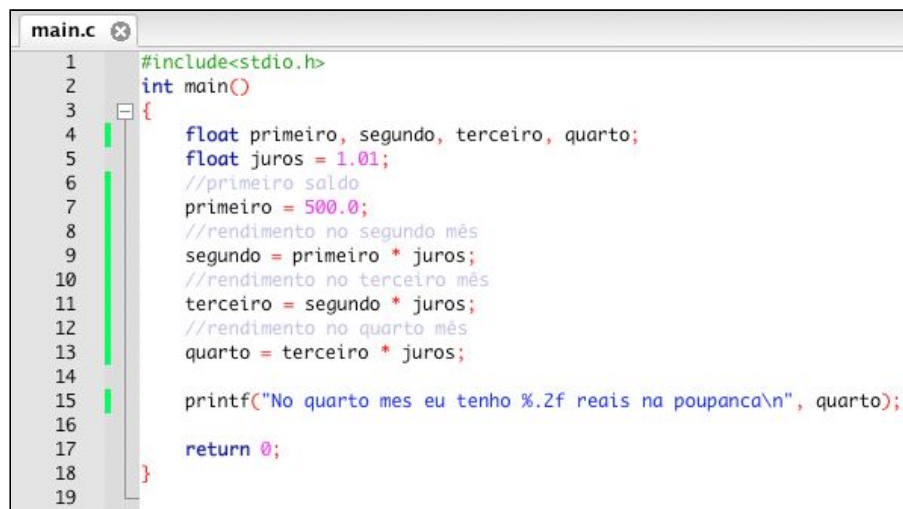
Note que:

$$- \text{radiano} = \frac{\text{graus} \times \pi}{180}$$

- a função  $\sin(x)$  recebe como parâmetro o  $x$  em radiano.

[salve o seu código com o nome: **exercicio1.c**]

2) Uma conta poupança foi aberta com um depósito de R\$500,00. Esta conta é remunerada em 1% de juros ao mês. O código a seguir apresenta uma forma de implementação para três meses de acúmulo de juros. Reescreva esse código usando apenas duas variáveis



```

main.c
1  #include<stdio.h>
2  int main()
3  {
4      float primeiro, segundo, terceiro, quarto;
5      float juros = 1.01;
6      //primeiro saldo
7      primeiro = 500.0;
8      //rendimento no segundo mês
9      segundo = primeiro * juros;
10     //rendimento no terceiro mês
11     terceiro = segundo * juros;
12     //rendimento no quarto mês
13     quarto = terceiro * juros;
14
15     printf("No quarto mes eu tenho %.2f reais na poupanca\n", quarto);
16
17     return 0;
18 }
19

```

[salve o seu código com o nome: **exercicio2.c**]

---

3) Elaborar um programa em Linguagem C para resolver o seguinte problema:

Considere que os valores (inteiros e positivos) para as variáveis **a**, **b** e **c** correspondem aos lados de um triângulo retângulo com catetos **a** e **b**, e hipotenusa **c**. Determinar a área do triângulo pela fórmula:

$$área = \sqrt{s * (s - a) * (s - b) * (s - c)}, \text{ onde } s = \frac{a + b + c}{2}$$

[salve o seu código com o nome: **exercicio3.c**]

---

4) Sabendo que:

- Um binário de  $n$  bits é capaz de comportar até  $2^n$  diferentes combinações de 1 e 0. Por exemplo, um binário de 2 bits comporta  $2^2 = 4$  valores.
- A conversão entre binário e decimal é dada por

$$d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \dots + d_02^0$$

Ou seja, para um binário de  $n = 4$  dígitos, teríamos por exemplo:

$$\begin{aligned} &0 \ 1 \ 1 \ 0 = \\ &0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = \\ &0 + 4 + 2 + 0 = 6 \end{aligned}$$

Como referência, a tabela a seguir apresenta todas as possíveis combinações para um binário de 2 bits, bem como a conversão de cada valor para decimal.

Binário	Conversão	Decimal
00	$0 \times 2^1 + 0 \times 2^0$	0
01	$0 \times 2^1 + 1 \times 2^0$	1
10	$1 \times 2^1 + 0 \times 2^0$	2
11	$1 \times 2^1 + 1 \times 2^0$	3

Com essas informações, escreva um código que:

- a) usa a função `pow` para calcular e imprimir quantos valores um binário de 4 bits comporta. (**Nota:** `pow(x, y)` calcula  $x^y$ )
- b) Recebe do usuário 4 valores referentes aos bits de um binário e imprime o decimal equivalente. (**Nota:** para facilitar, os valores podem ser lidos com “%d” como se cada bit fosse um inteiro).

[salve o seu código com o nome: **exercicio4.c**]

---

5) Sabendo que na linguagem C um inteiro com sinal ocupa 32 bits, sendo o primeiro deles um indicador do sinal, escreva um código para:

- a) calcular os limites inferior e superior que um inteiro comporta.
- b) Usar a função `print` para imprimir os limites inferior e superior calculados.
- c) Some 1 do maior valor, subtraia 1 do menor valor e os imprima novamente.

[salve o seu código com o nome: **exercicio5.c**]

---

6) Leia um valor inteiro em segundos e imprima o equivalente em horas, minutos e segundos

[salve o seu código com o nome: **exercicio6.c**]

7) Três amigos jogaram na loteria. Caso eles ganhem, o prêmio deve ser repartido proporcionalmente ao valor que cada um deu para a realização da aposta. Faça um programa que leia quanto cada apostador investiu, o valor do prêmio, e imprima quanto cada um ganharia do prêmio com base no valor investido.

[salve o seu código com o nome: **exercicio7.c**]

### Estruturas de controle

8) A fórmula de Bhaskara é considerada uma das mais importantes da matemática. Ela é usada para resolver as equações de segundo grau, e é dada por:

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

Essa equação dá como resultado duas raízes  $x_1$  e  $x_2$ , visto que a equação deve ser resolvida para  $-b + \sqrt{\Delta}$  e também para  $-b - \sqrt{\Delta}$ . Escreva um programa que receba os valores  $a$ ,  $b$  e  $c$  da equação e calcule as raízes  $x_1$  e  $x_2$  de acordo com a fórmula de Bhaskara.

**Nota:** a  $\sqrt{\Delta}$  só pode ser calculada para  $\Delta > 0$ , essa verificação deve ser incluída no código.

[salve o seu código com o nome: **exercicio8.c**]

---

9) Iremos fazer um programa para imprimir na tela o rendimento semestral global (RSG) de um aluno na UFMG. O programa a ser desenvolvido irá receber as notas de quatro disciplinas (nota1, nota2, nota3 e nota4), notas estas de 0 a 100, bem como a quantidade de créditos de cada disciplina (cred1, cred2, cred3 e cred4). Os créditos poderão ser valores de 20 a 60. Vamos assumir que o aluno não trancou nenhuma disciplina. O seu programa deve imprimir mensagens de erro se os valores forem inválidos (notas fora da faixa 0-100 e créditos fora da faixa 20-60, ambos sendo intervalos fechados).

No cálculo do RSG na UFMG, convertem-se os conceitos obtidos em cada atividade/disciplina em valores, observando-se a seguinte correspondência:

Conceito - Nota	Conceito	Valor
100 – 90	A	5
89 – 80	B	4
79 – 61	C	3
60 – 41	D	2
40 – 31	E	1
30 – 0	F	0

O valor do conceito de cada atividade em que o aluno se matriculou no semestre, excluindo as porventura trancadas, é multiplicado por seu respectivo número de créditos; os produtos assim obtidos são somados e o resultado é dividido pelo número total de créditos em que o aluno se matriculou no semestre.

[salve o seu código com o nome: **exercicio9.c**]

**10)** O cálculo do IMC é feito a partir da divisão do peso pela altura ao quadrado. Por exemplo, uma pessoa que pesa 80kg e que tem altura de 1,80 m, terá um IMC de  $80 / (1.80)^2 = 24,69$ .

Faça um programa que leia o peso e a altura da pessoa, e imprima na tela o valor do IMC, bem como indique a situação da pessoa, baseado no seu IMC.

Resultado	Situação
Abaixo de 17	Muito abaixo do <i>peso</i>
Entre 17 e 18,49	Abaixo do <i>peso</i>
Entre 18,5 e 24,99	<i>Peso</i> normal
Entre 25 e 29,99	Acima do <i>peso</i>
Entre 30 e 34,99	<i>Obesidade</i> I
Entre 35 e 39,99	<i>Obesidade</i> II (severa)
Acima de 40	<i>Obesidade</i> III (mórbida)

Compile, execute e teste o seu código implementado. [código: **exercicio10.c**].

**11)** Considere a seguinte definição de ano bissexto (ano em que o mês de fevereiro tem 29 dias):

- Um ano não divisível por 100 e divisível por 4 é bissexto;
- Um ano divisível por 400 é bissexto;
- Os demais anos **não** são bissextos.
- Lógica a ser implementada:
  - (não divisível por 100 E divisível por 4) OU (divisível por 400);

Elabore o seu código principal do programa, que deve fazer o seguinte:

- Receba um valor inteiro que é o ano a ser testado;
- Teste se o ano é bissexto;
- Imprima na tela para o usuário a mensagem “O ano <valor do ano> informado é bissexto” ou então “O ano <valor do ano> informado não é bissexto”.

[salve o seu código com o nome: **exercicio11.c**]

### Estruturas de repetição e Funções

**12)** Escreva um programa em C para cálculo do máximo divisor comum (MDC) entre três números.

Dica:  $\text{mdc}(a,b,c) = \text{mdc}(a, \text{mdc}(b,c))$

Exemplo: caso a entrada seja 156, 182, 429, sua saída deve ser 13.

[salve o seu código com o nome: **exercicio12.c**]

---

**13)** Escreva um programa que leia dois inteiros N e M, e imprima todos os números ímpares maiores que N e menores que M.

Exemplo: caso N = -4 e M = 7, a saída deve ser:  
-3 -1 1 3 5

[salve o seu código com o nome: **exercicio13.c**]

---

**14)** Escreva um programa que leia um número positivo N e imprima N linhas do triângulo ordenado. Um triângulo ordenado pode ser visto abaixo, para o caso N = 6.

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

Note que foram impressas exatamente N = 6 linhas e a i-ésima linha contém exatamente i números.

[salve o seu código com o nome: **exercicio14.c**]

---

**15)** Estendendo a questão 4, escreva um programa que recebe do usuário o número (n) de bits de um binário e:

a) usa a função `pow` para calcular e imprimir quantos valores um binário de (n) bits comporta. (**Nota:** `pow(x, y)` calcula  $x^y$ )

b) Recebe do usuário (n) valores referentes aos bits do binário e imprime o decimal equivalente. (**Nota:** para facilitar, os valores podem ser lidos com “%d” como se cada bit fosse um inteiro).

[salve o seu código com o nome: **exercicio15.c**]

---

**16)** Uma empreiteira paga seus pedreiros por metros quadrados de serviços produzidos **diariamente**, adicionando também o valor do almoço e passagem para ir e voltar do trabalho. Essa construtora procurou você para desenvolver um sistema que calcula quanto ela deve pagar para cada um de seus pedreiros. Apresente um programa que resolva o problema citado.

- O programa deverá receber:
  - O valor do vale-alimentação;
  - O valor do transporte unitário (a cada dia o pedreiro usa 2 vales);

- Quantos metros quadrados trabalhou;
- O valor do metro quadrado é calculado da seguinte forma:

Quantos metros quadrados trabalhou?	Valor do metro quadrado
Menos de 10m <sup>2</sup>	R\$ 10,00
Mais ou igual a 10m <sup>2</sup> até 20m <sup>2</sup>	R\$ 11,50
Mais ou igual a 20m <sup>2</sup>	R\$ 13,00

- Crie a lógica para calcular o valor a ser pago ao pedreiro;
- Depois na função principal, receba os valores especificados, execute os cálculos e armazene o valor do pagamento;
- Ao final, o programa deve imprimir a seguinte mensagem: "O pedreiro deve receber R\$ X", onde x é o valor a ser pago.

[salve o seu código com o nome: **exercicio16.c**]

---

**17)** Escreva as seguintes funções utilizando, se necessário, estruturas condicionais e de repetição em C.

a) fat(n): retorna o valor do fatorial de n.

Exemplo: fat(5) deve retornar 120.

b) mdc(a,b): retorna o máximo divisor comum entre a e b.

c) mdc4(a,b,c,d): retorna o máximo divisor comum entre a, b, c e d.

d) multiplos(n, x): procedimento que escreve na tela todos os múltiplos de n entre 0 e x, inclusive, separados por vírgula.

Exemplo: multiplos (3, 21) deve imprimir:

0, 3, 6, 9, 12, 15, 18, 21

Note que não deve aparecer uma vírgula após o último número e que 0 sempre será impresso.

e) primo(x): retorna 1 se x é primo e 0 em caso contrário.

f) decrescente(x): procedimento que escreve uma sequência de inteiros menores que x e maiores que 0.

Exemplo: decrescente(6) deve imprimir:

5 4 3 2 1

g) pa(a, r, n): retorna a soma dos termos de uma Progressão Aritmética de n termos, com termo inicial a razão r.

h) dig(n): retorna a soma dos dígitos de um inteiro positivo n.

Exemplo: A soma dos dígitos de 132, por exemplo, é 6. Já a soma dos dígitos de 1095 é 15.

i) mmc(a, b): retorna o menor múltiplo comum de a e b.

j) n\_esimo\_primo(n): retorna o n-ésimo primo.

Exemplos:

```
n_esimo_primo(1) deve retornar 2;  
n_esimo_primo(2) deve retornar 3;  
n_esimo_primo(17) deve retornar 59;
```

k) hora (h, m, s): recebe três inteiros **h**, **m** e **s**, correspondentes a hora, minuto e segundo e imprime o horário um segundo depois.

Exemplo:      hora (16, 06, 59) deve imprimir 16:07:00  
                 hora (23, 59, 59) deve imprimir 00:00:00

[salve o seu código com o nome: **exercicio17.c**]