



Professor: Fabrício Valadares      Disciplina: Algoritmos e programação estruturada Curso:  
TADS/SI/Engenharias      Período: Material: Guia de estudo Valor: 300

Nome: Leandro Henrique Lopes Sampaio  
Curso: Análise e Desenvolvimento de Sistemas  
Disciplina: Algoritmos e Programação Estruturada  
Professor: Fabrício Valadares  
Turma ADS Noite  
RA: 36885254094

1. Criar um algoritmo que imprima a sequência dos 100 primeiros números pares.

```
main.c
1  #include <stdio.h>
2  //ainda não sei o que comentar
3  int main() {
4      int p = 2;
5
6      for (p = 2; p <= 100; p +=2){
7          printf("%i\n",p);
8      }
9
10     return 0;
11 }
```

2.Crie um algoritmo que imprima a tabuada de multiplicação de um número dado pelo usuário (e.g., se o usuário informar o número 10, deve-se exibir a multiplicação do 1 até o 10, no mesmo formato da tabuada).

```
main.c
1  #include <stdio.h>
2  //criando uma tabuada
3  int main() {
4      int tabuada, t1, t2;
5
6      printf("Digite um número de 1 a 10:");
7      scanf("%i",&t1);
8
9      for (t2 = 1; t2 <=10; t2 ++){
10         tabuada = t1*t2;
11         printf("%i * %i = %i \n",t1, t2,tabuada);
12     }
13 }
14
15 return 0;
16 }
```

3.Diferencie os três comandos de repetição apresentados: while, do-while, for.

while = enquanto

do-while = faça/enquanto

for = para idexan

while e do-while são estruturas condicionais

for é uma estrutura determinística

while e for fazem um teste condicional no cabeçalho (eles testam primeiro executam depois)

já o do-while executa primeiro testa depois

4.Imprima a seguinte frase na vertical: “Estruturas de repetição facilitam muito a minha vida, somente assim eu conseguiria imprimir esta frase”.

```
main.c
1  #include <stdio.h>
2
3  int main(void)
4  {
5      char FraseVertical [] = "Estruturas de repeticao facilitam muito a
        minha vida, somente assim eu conseguiria imprimir esta frase. FALEI
        QUE IA FAZER DE NOVO";
6
7      int i;
8
9      for(i = 0; FraseVertical[i] != '\0'; i++)
10     {
11         printf("%c\n", FraseVertical[i]);
12     }
13
14     return 0;
15 }
16 //FONTE:http://linguagemc.com.br/string-em-c-vetor-de-caracteres/
```

5.Imprima a seguinte figura:

```
*
**
***
****
*****
```

Altura deve ser determinada pelo usuário. (não ultrapasse 20 linhas)

6.Altere o algoritmo anterior, caso ainda não tenha feito, para que ele impessa o usuário de informar valores negativos, ou, valores acima de 20. Quando isso acontecer o usuário deve ser informado do erro, e perguntado se deseja ou não continuar.

7. Seu algoritmo deve ler o sexo e a idade de uma população de 20 pessoas, e, em seguida, informar a média de idade da população masculina e a população feminina.

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define PESSOAS 20
5
6 void limparEntradaDeDados()
7 {
8     int ch;
9     while ((ch = fgetc(stdin)) != EOF && ch != '\n')
10     {
11     }
12 }
13
14 void mediaIdadePorSexo()
15 {
16     char sexo;
17     int idade;
18     int idadeAcumulado;
19     double mediaMasculino;
20     double mediaFeminino;
21     int contMasculino = 0;
22     int contFeminino = 0;
23
24     idadeAcumulado = 0;
25     mediaMasculino = 0;
26     mediaFeminino = 0;
27     printf("\n\n");
28
29     for (int i = 0; i < PESSOAS; i++)
30     {
31         printf("\nPor favor, informe o %2iª idade: ", (i + 1));
32         scanf("%i", &idade);
33         limparEntradaDeDados();
34
35         printf("\nPor favor, informe o sexo\n(M)asculino (F)eminino:");
36         scanf("%c", &sexo);
37         limparEntradaDeDados();
38
39         if (sexo == 'f' || sexo == 'F')
40         {
41             printf("Sexo Feminino\n");
42             contFeminino++; // contFeminino += 1 || contFeminino = contFeminino + 1
43             mediaFeminino += idade;
44         } else if (sexo == 'm' || sexo == 'M')
45         {
46             printf("Sexo Masculino\n");
47             contMasculino++;
48             mediaMasculino += idade;
49         } else {
50             printf("O sexo informado é inválido\n\n");
51         }
52     } // Fim do for
53
54     printf("\n\n");
55     printf("Resultado:\n");
56
57     printf("Foram contadas %2i mulheres e a média de idade é: %2.2lf", contFeminino,
58         (mediaFeminino / contFeminino));
59
60     printf("\n\nFim...");
61 }
62
63 int main(void) {
64     mediaIdadePorSexo();
65     return EXIT_SUCCESS;
66 }
```

8. Crie um algoritmo que calcule o IMC de uma pessoa, para isso, você deve coletar a altura e o peso do mesmo. Pesquise na internet como o cálculo é realizado, lembre-se de desprezar a diferença entre sexos. Para cada intervalo do IMC existe uma classificação, ela deve ser exibida por seu algoritmo, junto com uma mensagem informativa ao usuário. Boa referência: <https://dms.ufpel.edu.br/casca/modulos/imc-calc#comp/imc-main>

```

main.c
1 #include <stdio.h>
2 //LEIA TODOS OS COMENTARIOS PLEASE =)
3 //CALCULAO DO IMC (PESO / (ALTURA * ALTURA) = IMC)
4 //TABELA VALORES DO IMC
5 //MENOR QUE 18.5 MAGREZA Grau 0
6 //ENTRE 18.5 E 24.9 NORMAL Grau 0
7 //ENTRE 25.0 E 29.9 SOBREPESO Grau I
8 //ENTRE 30.0 E 39.9 OBESIDADE Grau II
9 //MAIOR QUE 40.0 OBESIDADE GRAVE Grau III
10
11 int main(void)
12 {
13     float IMC;
14     float ALTURA;
15     float PESO;
16
17     printf("Digite sua ALTURA:\n");
18     scanf("%f", &ALTURA);
19
20     printf("Digite sua PESO:\n");
21     scanf("%f", &PESO);
22
23     IMC = PESO / (ALTURA * ALTURA);
24     printf("%.2f ", IMC);
25
26
27     //MENOR QUE 18.5 MAGREZA Grau 0
28     if (IMC <= 18.5)
29     {
30         printf("MAGREZA Grau 0: seu IMC e menor ou igual a 18.5");
31     }
32
33     //ENTRE 18.5 E 24.9 NORMAL Grau 0
34     else if (IMC > 18.5 && IMC < 25)
35     {
36         printf("NORMAL Grau 0: seu IMC está entre 18.5 e 24.9");
37     }
38
39     //ENTRE 25.0 E 29.9 SOBREPESO Grau I
40     else if (IMC >= 25.0 && IMC < 30)
41     {
42         printf("SOBREPESO Grau I: seu IMC está entre 25.0 e 29.9");
43     }
44
45     //ENTRE 30.0 E 39.9 OBESIDADE Grau II
46     else if (IMC >= 30.0 && IMC < 40)
47     {
48         printf("OBESIDADE Grau II: seu IMC está entre 30.0 e 39.9");
49     }
50
51     //MAIOR QUE 40.0 OBESIDADE GRAVE Grau III
52     else if (IMC >= 40.0)
53     {
54         printf("OBESIDADE GRAVE Grau III: seu IMC e maior ou igual a 40.0");
55     }
56
57     return 0;
58 }
59
60 //FONTE
61 //https://www.progranasaudefacil.com.br/calculadora-de-imc
62
63 //FABRICIO USE ALTURA 1.70 e PESO 72.25 para descobri o "buraco" no codigo dos dos demais alunos e quem descobriu o "buraco" foi o Carlos mas o Tulio e
    eu ficamos quase 1 hr discutindo isso ate Tulio achar a solução.

```

9. Os números primos existentes entre os 100 primeiros números naturais não nulos.

```
main.c
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void)
5  {
6      int p=1;
7      int dividir=1;
8      int sobra=0;
9      int total=0;
10
11
12     //numeros primos são numeros divisiveis apenas por dois numeros divisiveis por 1 ou por ele
    mesmo.
13     printf("números naturais não nulos de 0 a 100:\n");
14
15     for (p = 1; p <=100; p++)
16     {
17         while (dividir <=100)
18         {
19             sobra = (p% dividir);
20
21             if (sobra == 0)
22             {
23                 total++;
24             }
25         }
26         dividir++;
27
28         if(total == 2 )
29         {
30             printf("%i", p);
31         }
32         dividir = 1;
33         sobra = 0;
34         total = 0;
35     }
36
37     return 0;
38 }
```

10. Escreva um programa em C++ que lhe permita calcular o fatorial de um número dado pelo usuário.

```
main.c
1  #include <stdio.h>
2
3  int main(void){
4      int fatorial;
5      int n;
6      printf("Insira valor para calcular seu fatorial:");
7      scanf("%d", &n);
8
9      for(fatorial = 1; n > 1; n = n - 1)
10         fatorial = fatorial*n;
11     printf("Seu Fatorial Calculado e: %d", fatorial);
12
13     return 0;
14 }
```

11. Escreva um algoritmo que imprima os n primeiros números da sequência de Fibonacci. Ref.: [https://pt.wikipedia.org/wiki/Sequ%C3%Aancia\\_de\\_Fibonacci](https://pt.wikipedia.org/wiki/Sequ%C3%Aancia_de_Fibonacci), esta referência já tem uma implementação do algoritmo em C, não sugiro copiar, pois o objetivo é aprender a fazer!

12. Codifique um programa que faça a leitura de três valores reais e verifique se estes valores representam os lados de um triângulo. O programa deve verificar se existe um triângulo com estes lados e imprimir uma mensagem ao usuário informando o resultado. Para verificar a condição de existência de um triângulo, com os lados A, B, e C, qualquer lado deve ser menor que a soma dos outros dois, ou seja:

$$A < B + C \text{ e } B < A + C \text{ e } C < A + B$$

Abaixo estão duas ilustrações de execuções do programa:

Execução 1

Entrada

CONDIÇÃO DE EXISTÊNCIA DE UM TRIÂNGULO  
DIGITE O VALOR DO LADO A: 1.25 DIGITE O VALOR  
DO LADO B: 2.35  
DIGITE O VALOR DO LADO C: 1.25

Saída

OS VALORES 1.25, 2.35 e 1.25  
REPRESENTAM OS LADOS DE UM TRIÂNGULO

Execução 2

#### Entrada

```
CONDIÇÃO DE EXISTÊNCIA DE UM TRIÂNGULO  
DIGITE O VALOR DO LADO A: 0.89  
DIGITE O VALOR DO LADO B: 1.98  
DIGITE O VALOR DO LADO C: 0.4
```

#### Saída

```
TRIÂNGULO INEXISTENTE ! OS VALORES 0.89, 1.98 E 0.4  
NÃO REPRESENTAM OS LADOS DE UM TRIÂNGULO
```

## Comentários Gerais

- **Submissão:**

- Um arquivo em formato PDF com seu nome completo, RA, turma e e-mail de contato.
  - \* Todos os códigos devem estar inseridos dentro do arquivo PDF, organizados na sequência da lista.
- Código fonte, extensão .c, não anexar os executáveis! Com a seguinte identificação: q1.c, q2.c, ..., qx.c, onde o x, indica o número da questão resolvida.
- Ambos os arquivos devem ser inseridos em um arquivo compactado, **NO FORMATO ZIP!!!**, com o seguinte identificador: seu\_nome\_completo\_le02.zip – Em breve será definido o local de entrega.
- Comece a resolver esta lista logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.

- **Entrega:**

- Até a próxima aula – entrega parcial.
  - Por e-mail: [fabricao.valadares@pitagoras.com.br](mailto:fabricao.valadares@pitagoras.com.br) – Lembrem-se do ED e do desafio nota máxima.
- Todos os algoritmos devem ser apresentados C, exceto quando houver instrução contrária. Lembre-se que antes de escrever sua solução, é **necessário pensar em quais são as entradas, saídas e qual o processamento necessário para solucionar o problema**. Este é um passo importante, pois o planejamento prévio evitará falhas durante a construção do algoritmo.
  - O raciocínio lógico possui um papel importantíssimo em programação de computadores. Além de conhecer todas as instruções e como utilizá-las, é necessário saber quando utilizá-las.