

# **Colégio Técnico da UFMG - COLTEC**

PROF. MÁRCIO FANTINI

Lista 00 - Programação C - Revisando Conceitos

## **Introdução**

- Essa lista contém 100 problemas.
- Eles estão organizados em ordem crescente de complexidade, iniciando-se do mais básicos de todos (o Hello World) até problemas que envolvem structs, funções, entrada-saída de arquivos e algoritmos de ordenação.
- Os problemas estão organizados em 13 seções, também organizadas em ordem de complexidade dos assuntos.
- A ideia é verificar o que vocês já viram no 1o ano e já sabem fazer, o que já viram mas não lembram bem e o que vocês não viram.
- É uma lista opcional, nesse momento. Mas aqueles que já quiserem ir fazendo, fiquem a vontade.
- Disponibilizei videoaulas explicando alguns exercícios e como fazer um menu para acesso aos exercícios.

# 1 Exercícios Básicos: Entrada/Saída e Tipos de Dados e Operações Aritméticas

1. Faça um programa C para imprimir a mensagem "Hello World!".
2. Faça um programa C para ler um inteiro e mostrá-lo.
3. Faça um programa C para ler um inteiro e mostrá-lo com a frase: "Inteiro Digitado:  $X$ ", sendo  $X$  o valor dado.
4. Faça um programa C para ler dois inteiros, somá-los e mostrar o resultado.
5. Faça um programa C para ler dois inteiros, somá-los e mostrar os dois números lidos e o resultado na frase: "A soma de  $X$  com  $Y = Z$ ", sendo  $X, Y$  os números lidos e  $Z = X + Y$ .
6. Faça um programa C para ler dois inteiros e mostrar o resultado e o resto da divisão inteira primeiro e o segundo número lidos.
7. Faça um programa C para ler dois inteiros e mostrar o resultado da divisão real (float). Mostre o resultado e os dois números lidos assim: " $x/y = z$ ", sendo  $x$  e  $y$  os números lidos e  $z$  o resultado da divisão real (float).
8. Faça um programa em C para ler um caractere, mostrá-lo e mostrar seu código ASCII.
9. Faça um programa em C para ler um nome do usuário (primeiro nome apenas, uma palavra, sem espaços) e mostrá-lo.
10. Faça um programa em C para ler o nome completo do usuário (com espaços) e mostrá-lo.
11. Faça um programa em C para ler e mostrar 4 variáveis: um inteiro, um caractere, um número real (float) e uma string (palavra única, sem espaços). Os dados lidos devem ser apresentados como mostrado abaixo (sendo  $X, Y, Z, S$  os dados lidos):

```
inteiro: X
char:    Y
float:   Z
string:  S
```

12. Faça um programa em C para ler dois números ( $x$  e  $y$ ) e mostrar o resultado da potência do primeiro elevado ao segundo, isto é  $z = x^y$ .

## 2 Exercícios Básicos: Tipos de Dados, Strings, Operações Lógicas, Decisões

13. Faça um programa em C para ler uma string, imprimi-la de trás pra frente.
14. Faça um programa C para ler uma frase, mostrá-la e informar quantos caracteres ela tem.
15. Faça um programa C para ler uma frase, mostrá-la e dizer quantos caracteres espaços e quantos caracteres 'a' ela contém.
16. Faça um programa C para ler um inteiro e dizer se ele é par ou ímpar.
17. Faça um programa em C para ler um inteiro e dizer se é positivo, negativo ou zero.
18. Faça um programa em C para ler dois inteiros,  $x$  e  $y$  e dizer se o primeiro é divisível pelo segundo.
19. Faça um programa em C para ler dois inteiros,  $x$  e  $y$  e dizer se o segundo é múltiplo do primeiro.
20. Faça um programa em C para ler um inteiro na base 16 e mostrar a saída como:

```
Número lido:
    Na base 16: x16
    Na base 10: x10
```

(sendo  $x16$  o número lido na base 16 e  $x10$  o número lido na base 10).

21. Faça um programa em C para ler dois números inteiros (base 10) e mostrar o resultado das operações AND, OR e XOR entre eles.
22. Faça um programa em C para ler dois números inteiros na base 16 e mostrar o resultado das operações AND, OR e XOR entre eles.
23. Faça um programa em C para ler dois números inteiros na base 16, de (no máximo) 16 bits cada e mostrar o resultado das operações AND, OR e XOR entre eles, efetuadas **bit-a-bit**.
24. Faça um programa em C para ler três inteiros e mostrar se eles podem formar um triângulo.

### 3 Exercícios Básicos: Decisões e Repetições

25. Faça um programa em C para ler uma string e dizer se ela é um palíndromo.
26. Faça um Programa em C para ler 4 números inteiros e mostrar o maior e o menor dentre eles.
27. Faça um Programa em C para ler 6 números inteiros, mostrá-los e mostrar o valor da soma dos números pares e dos números ímpares.
28. Fazer um Programa C que leia um número entre 0 e 100. O número deve ser positivo e está dentro do intervalo pedido. Se não satisfizer as condições o programa deve abortar. O número corresponde a uma nota final. O programa deve imprimir o conceito do aluno, que vai de A até F, segundo a regra:

- |   |                                     |  |
|---|-------------------------------------|--|
| • $90 \leq nota \leq 100$ , conceito A. | • $70 \leq nota < 80$ , conceito C  | • $nota < 60$ , conceito E (reprovado) |
| • $80 \leq nota < 90$ , conceito B.     | • $60 \leq nota < 70$ , conceito D. |  |

29. Faça um programa em C para calcular e mostrar o valor do fatorial de um número inteiro  $n$  dado.
30. Fazer um programa em C que calcule e escreva o valor de S, dado por

$$S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \cdots + \frac{99}{50}$$

31. Fazer um programa em C que calcule e escreva o valor de S, dado por

$$S = \frac{2^1}{50} + \frac{2^2}{49} + \frac{2^3}{48} + \cdots + \frac{2^{50}}{1}$$

32. Fazer um programa em C que calcule e escreva o valor de S, dado por

$$S = \frac{37 \times 38}{1} + \frac{36 \times 37}{2} + \frac{35 \times 36}{3} + \cdots + \frac{1 \times 2}{37}$$

33. Fazer um programa em C que calcule e escreva a soma dos 50 primeiros termos da série abaixo:

$$\frac{1000}{1} - \frac{997}{2} + \frac{994}{3} - \frac{991}{4} + \cdots$$

34. Fazer um programa em C que calcule e escreva a soma dos 30 primeiros termos da série abaixo:

$$\frac{480}{10} - \frac{475}{11} + \frac{470}{12} - \frac{465}{13} + \cdots$$

## 4 Exercícios Básicos/Intermediários: Repetições e Somatórios

35. Fazer um programa em C para gerar e escrever uma tabela com valores do seno de uma ângulo  $\theta$ , em radianos, utilizando a série de Mac-Laurin truncada, apresentada na equação 1. Considere:

- Os valores dos ângulos  $\theta$  devem variar de 0 a  $2\pi$ , variando de 0.1.
- Compare os valores calculados com o valor "real" do seno, usando a função `sin()`.

$$\text{sen}(\theta) = \theta - \frac{\theta^3}{6} + \frac{\theta^5}{120} - \frac{\theta^7}{5040} \quad (1)$$

36. Fazer um programa em C para calcular e escrever o valor do número  $\pi$ , com precisão de 0.0001, usando a série:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots \quad (2)$$

- Para definir a precisão do  $\pi$  calculado por você, efetue a diferença dele com a variável `M_PI` da biblioteca `<math.h>`.

37. Faça um programa em C para calcular o valor de

$$\frac{\text{seno}(x)}{x}$$

usando a série

$$\frac{\text{seno}(x)}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} \dots$$

Use termos na série para que o erro entre o valor real (obtido pela conta  $\sin(x)/x$ ) e o valor calculado seja menor que 0.01. O valor do ângulo  $x$  deve ser usado na série e na função `sin()` **em radianos**. O programa deve pedir o valor de  $x$ .

38. Faça um programa em C para efetuar o somatório (com  $N$  dado).

$$R = \sum_{n=-N}^N \frac{n+2}{2n-1}$$

39. Faça um programa em C para efetuar o somatório (com  $x$ ,  $N$  dados).

$$R = \sum_{n=1}^N \frac{x^n}{n!}$$

40. Faça o programa em C para efetuar o somatório (com  $y$  e  $N$  dados).

$$R = \sum_{n=1}^N \frac{y^{-n}}{n!}$$

41. Faça o programa em C para efetuar o somatório (com  $N$  dado).

$$R = \sum_{n=1}^N \frac{n+2\pi}{2n\pi-1}$$

## 5 Exercícios Básicos: Figuras Geométricas com Loops aninhados

42. Faça um programa em C para gerar um quadrado de lado  $N$  formado de asteriscos. O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

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

43. Faça um programa em C para gerar um quadrado de lado  $N$  formado de  $N$  números em cada linha e de  $N$  linhas. O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

```
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
```

44. Faça um programa em C para gerar um quadrado de lado  $N$ , isto é,  $N$  linhas, com linhas que vão de 1 a  $N$ . O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

45. Faça um programa em C para gerar um quadrado de lado  $N$ , isto é,  $N$  linhas, com linhas que vão de  $N$  a 1. O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

```
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
```

46. Faça um programa em C para gerar um triângulo retângulo, formado de asteriscos, de catetos de tamanho  $N$  ( $N$  asteriscos), ou seja  $N$  linhas e primeira linha de  $N$  asteriscos, segunda linha de  $N - 1$  asteriscos, etc. O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

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

47. Faça um programa em C para gerar um triângulo retângulo formado de asteriscos, de catetos de tamanho  $N$  ( $N$  asteriscos), ou seja  $N$  linhas e primeira linha com 1 asterisco, segunda linha 2 asteriscos, até a última linha de  $N$  asteriscos. O valor de  $3 \leq N \leq 15$  é dado pelo usuário. A figura abaixo mostra o caso de  $N = 5$ .

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

48. Faça um programa em C para gerar um triângulo de números, de  $N$  linhas, com linhas de números crescentes, a partir de 1 até  $N$  na primeira linha, de 1 até  $N - 1$  na segunda, e assim por diante, até a última linha com apenas o número 1. A figura abaixo mostra o caso de  $N = 5$ . Considere  $2 \leq N \leq 9$ .

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

49. Faça um programa em C para gerar um triângulo de números, de  $N$  linhas, com linhas de números decrescentes, a partir de  $N$  até 1 na primeira linha, de  $N$  até 2 na segunda, e assim por diante, até a última linha com apenas o número igual  $N$ . A figura abaixo mostra o caso de  $N = 5$ . Considere  $2 \leq N \leq 9$ .

```
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

## 6 Exercícios Básicos/Intermediários: Figuras Geométricas

50. Faça um programa em C que lê um inteiro positivo  $n$  e imprime um triângulo constituído por números, iniciando com o número lido com o seguinte formato:

```
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Observação: No caso a tabela acima foi impressa com valor de  $n$  igual a 6.

51. Faça um programa em C que imprima na tela um triângulo isóceles de altura e largura  $n$ . Por exemplo, se  $n = 6$ , a figura deve ficar como abaixo:

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

52. Faça um programa em C para imprimir um triângulo de altura  $2n - 1$  e largura  $n$ . Por exemplo, para  $n = 4$  a figura deve ficar como abaixo:

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

53. Faça um programa em C para imprimir um triângulo de altura  $n$  e largura  $2n - 1$ . Por exemplo, para  $n = 6$ , a figura deve ficar como abaixo:

números dados a seguir estão na base 10. Quantos bits são necessários para representar

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



54. Faça um programa em C para ler um número  $N$  (entre 1 e 20) e imprimir um quadrado na tela, de tamanho  $N$ , sendo  $N$  o número de asteriscos. Por exemplo, se  $N=5$ , o programa deve imprimir o quadrado dado abaixo:

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

55. Faça um programa em C para imprimir na tela a figura (diamante) dada abaixo:

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

56. Repita o programa acima, agora para ler um número  $N$ , ímpar, entre 1 e 19 e gerar o diamante de  $N$  linhas, com a mesma simetria da figura acima.
57. Faça um programa para gerar um triângulo de altura  $N$  ( $N$  linhas) e largura  $N$  (maior coluna igual a  $N$ ), montado com números que vão de 1 a  $N$ . A figura abaixo foi criada para  $N = 5$ . Além de montar o triângulo, o programa deve mostrar (a uma distância adequada, com alinhamento correto) o valor da soma de cada linha, como dado na figura abaixo. Considere  $1 \leq N \leq 9$ .

1	1
1 2	3
1 2 3	6
1 2 3 4	10
1 2 3 4 5	15

## 7 Exercícios Básicos/Intermediários: Vetores, Matrizes

58. Faça um programa em C para ler 5 inteiros e armazená-los num vetor. Mostrar o vetor como vetor linha e como vetor coluna. Mostrar o maior e o menor dos elementos do vetor.
59. Faça um programa em C para gerar e mostrar um vetor de  $N$  números aleatórios entre 1 e 100, sendo  $0 < N < 20$  um inteiro dado pelo usuário. Após gerar o vetor, o programa deve mostrar o maior, o menor e a média.
60. Faça um programa em C para criar uma matriz  $4 \times 4$  de números inteiros aleatórios entre 1 e  $n$ , sendo  $n$  dado pelo usuário. O programa deve mostrar a matriz no formato matricial.
61. Faça um programa em C para preencher uma matrix  $M$   $n \times m$  segundo a regra:

$$M(i, j) = i + j$$

(use os índices do C: primeira linha e primeira coluna =  $(0, 0)$ .  $n$  e  $m$  são dados pelo usuário e tem que ser no máximo 6.).

62. Faça um programa em C para preencher uma matrix  $M$   $n \times m$  com números aleatórios. Os valores  $n$  e  $m$  são dados pelo usuário e tem que ser no máximo 6.
63. Faça um programa em C para calcular e escrever a tabuada dos números de 1 a 10, mostrando os dois termos e o resultado.
64. Faça um programa em C para:
- (a) preencher aleatoriamente um vetor  $5 \times 1$ , com números aleatórios entre 0 e 10.
  - (b) achar a posição do maior elemento desse vetor (se tiver mais que um, informar a posição de todos)
  - (c) mostrar a soma dos elementos desse vetor
65. Repetir o exercício acima para uma matriz  $3 \times 3$
66. Fazer um programa para preencher uma matriz segundo a regra  $X(i, j) = i + j$  e depois mostrar a soma de cada linha e de cada coluna. O programa deve perguntar no início o tamanho da matriz, que pode variar de  $1 \times 1$  até  $10 \times 10$ .
67. Fazer um programa para preencher um vetor de caracteres (de tamanho 1 a 10, definido pelo usuário).
68. Fazer um programa para preencher criar um vetor aleatório de 100 pontos. Os elementos devem ser inteiros na faixa entre 0 e 200.

## 8 Exercícios Intermediários: Vetores, Matrizes, Somatórios

69. Considere a matriz  $Z$ ,  $3 \times 3$  dada abaixo.

$$Z = \begin{bmatrix} 2 & 3 & -1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

A partir dessa matriz, faça um programa que calcule a soma  $\sigma$  dada por:

$$\sigma = \sum_1^3 Z_{i2} + \sum_1^3 Z_{3j}$$

Os índices considerados nos somatórios são os da matemática. Obviamente deve-se fazer a adaptação para o programa em C.

70. Faça um programa para preencher uma matriz  $n \times m$  ( $n$  e  $m$  são inteiros informados pelo usuário. Ambos devem ser menor ou igual a 6) com a regra:

$$M_{ij} = 2 * i + j$$

Após preencher a matriz o programa deve calcular a soma de cada linha e mostrar o valor calculado. O programa deve também mostrar a matriz no formato matricial. Ao preencher usando a regra deve-se considerar a notação da linguagem C (isto é, a primeira posição da matriz é (0,0)).

71. Faça um programa para preencher um vetor de 10 posições usando a regra dado abaixo.

$$v[i] = i^3$$

72. Faça um programa para preencher um vetor de 10 posições usando a regra dada abaixo.

$$v[i] = \begin{cases} i/2 & \text{se } i \text{ é par} \\ (i+1)/2 + i & \text{se } i \text{ é ímpar} \end{cases} \quad (3)$$

73. Faça um programa C para gerar duas matrizes  $5 \times 5$  de inteiros, mostrá-as, calcular e mostrar a soma delas.

74. Faça um programa que preencha uma matriz  $6 \times 6$  numérica, com o seguinte formato:

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 4 & 3 & 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

75. Faça um programa para criar uma matriz  $9 \times 9$  usando a mesma lógica do exercício anterior. Utilizando a matriz gerada, calcule e mostre:

- (a) A soma de todos seus elementos.
- (b) A soma dos elementos por linha.
- (c) A soma dos elementos da diagonal principal e secundária.

## 9 Exercícios Básicos/Intermediários: Ponteiros

76. Faça um programa em C para mostrar o tamanho em bytes das variáveis tipo int, float, double, long int e char.
77. Faça um programa em C que define uma variável  $x$  inteira e uma variável  $y$  float. O programa deve também definir dois ponteiros: um para  $x$  e outro para  $y$ . O programa deve ler valores para  $x$  e  $y$  e mostrar:
- a) O valor de  $x$  e  $y$
  - b) O endereço na RAM de  $x$  e  $y$
  - c) O conteúdo do endereço na RAM de  $x$  e  $y$
  - d) Os endereços armazenados nos ponteiros que apontam para  $x$  e  $y$ .
  - e) O conteúdo dos endereços armazenados nos ponteiros que apontam para  $x$  e  $y$
78. Seja o programa e sua saída, dados na figura 1. A partir dessa figura, monte uma tabela com a pilha de memória, colocando as variáveis definidas no programa. Entenda o que o programa faz. Escreva-o e execute-o.

```
#include <stdio.h>
int main(){
    int x,y,z,*px,*py,*pz;
    //int v[] = {6,5,4,3,2,1,0,-1},i;
    x = 0x2; y = 0xABCD; z = 0x23;
    px = &x;
    py = &y;
    pz = &z;
    printf("*px = %X\n",*px);
    printf("px = %p\n",px);
    printf("*py = %X\n",*py);
    printf("py = %p\n",py);
    printf("*pz = %X\n",*pz);
    printf("pz = %p\n",pz);

    printf("* (px+1) = %X\n",*(px+1));
    printf("(px+1) = %p\n", (px+1));

    printf("* (px+2) = %X\n",*(px+2));
    printf("(px+2) = %p\n", (px+2));

    printf("* (px+3) = %X\n",*(px+3));
    printf("(px+3) = %p\n", (px+3));

    printf("* (px+10) = %X\n",*(px+10));
    printf("(px+10) = %p\n", (px+10));
    return 0;}
```

```
Terminal
*px = 2
px = 0x7fff5d77917c
*py = ABCD
py = 0x7fff5d779180
*pz = 23
pz = 0x7fff5d779184
*(px+1) = ABCD
(px+1) = 0x7fff5d779180
*(px+2) = 23
(px+2) = 0x7fff5d779184
*(px+3) = 5D77917C
(px+3) = 0x7fff5d779188
*(px+10) = 0
(px+10) = 0x7fff5d7791a4

-----
(program exited with code: 0)
Press return to continue
```

Figura 1: Exemplo de programa com ponteiros e sua saída.

## 10 Questões Teóricas Sobre Ponteiros

1. O que é um ponteiro em C? Como declarar ponteiros? dê exemplos.
2. Explique o que fazem os operadores & e \*.
3. O que é mostrado quando damos um printf() no conteúdo de um ponteiro?
4. Como fazer para saber quanto de memória uma vetor ocupa?
5. Defina barramento de dados e barramento de endereços. Dê exemplos.
6. Para os números na base 16 dados, faça a conversão para base 2 e base 10.

- |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| (a) <i>A010</i> | (d) <i>10</i>   | (g) <i>15</i>   | (j) <i>20</i>   |
| (b) <i>AAFF</i> | (e) <i>A</i>    | (h) <i>16</i>   | (k) <i>FFFF</i> |
| (c) <i>A10</i>  | (f) <i>ABCD</i> | (i) <i>1010</i> | (l) <i>AA</i>   |

7. Explique o que faz o comando sizeof(). Dê exemplos.
8. Considere um micro cujo barramento de endereços possui 16 bits. Considere um ponteiro p apontando para a primeira posição de memória da tabela dada a seguir. Considere que todos os tipos de variáveis ocupam apenas 1 byte de memória RAM.

Responda:

120A	AB
120B	01
	FF
	12
	13
	6A
	FF

- (a) qual o valor de p?
- (b) qual o valor de \*p?
- (c) qual o valor de &p
- (d) qual o valor de \*(p+1)?
- (e) qual o valor de (p+5)?
- (f) qual o valor de (p+A)?
- (g) qual o valor de \*(p+2)?

9. Responda: existe diferença entre os dois trechos de programa abaixo? se sim, diga qual. Se não diga porque.

TRECHO 1

```
int a[2], *pa;
pa = &a[0];
```

TRECHO 2

```
int a[2], *pa;
pa = a;
```

10. Explique a diferença entre as três expressões (sendo p um ponteiro para inteiros)

p++            (\*p)++            \*(p++)

## 11 Exercícios Básicos/Intermediários: Ponteiros

79. Na linguagem C padrão, dado um array `arr[]`, sabemos que o valor de `arr[0]` pode também ser dado pela notação com o operador asterisco (\*), ou seja `arr[0]` pode ser dado por `*(arr+0)`. Faça um programa em C para gerar um vetor de tamanho 6, de números inteiros aleatórios entre 1 e 20, e use a notação com o operador \* para mostrar seus elementos.
80. Veja o trecho de programa dado abaixo. Rode-o e veja o que ocorre.

```
int main(){
int y[5],i;
int *z;
for(i=0;i<5;i++)
z[i]=i*i;
for(i=0;i<5;i++)
printf("z[%d]=%d ou *(z+%d) = %d\n",i,z[i],i,*(z+i));
return(0);

}
```

Sobre o programa, responda:

- a) A compilação foi feita com sucesso?
- b) A execução foi feita com sucesso? Se não, por quê?
- c) Onde está o erro, se o programa foi compilado com sucesso?
- d) Discuta o motivo do erro na execução.
- e) Como resolver o erro de execução?

Ainda sobre o programa, entenda a relação entre ponteiro e vetor. Repare que no exercício anterior definimos um vetor e usamos como ponteiro. Nesse exemplo tentamos fazer o contrário: definir um ponteiro e usá-lo como vetor. Responda ainda:

- f) É possível definir um ponteiro e usá-lo como vetor? sim/não? por quê?
- g) Resolva o problema do programa e execute-o com sucesso? Como fazer para resolver o problema? (é possível resolvê-lo acrescentando apenas uma linha

81. Reescreva o programa abaixo usando ponteiros

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main(){

    int V[10], i;
    time_t t; //tipo time_t
    srand((unsigned)time(&t)); //funcao time() recebe o endereco de t
    for (i=0;i<10;i++)
        V[i]=rand()%20;
    for (i=0;i<10;i++)
        printf("V[%d]=%i\n",i+1,V[i]);
    return(0);
}
```

82. Seja o código abaixo:

```
#include <stdio.h>
int main(void)
{ int mat[3] = {5, 9 , 11};
  int *ptr;
  ptr = mat;
  printf(" %d", *ptr + 1);
  printf(" %d", *(ptr + 1));
  return 0;
}
```

Pede-se:

- (a) O que será impresso (responda antes de executar o programa)?
- (b) E se acrescentarmos as três linhas abaixo, o que será impresso?

```
    ptr = mat;
    printf(" %d", *(ptr));
    printf(" %d", *(ptr + 2));
```

## 12 Exercícios Intermediários: Funções

83. Faça uma função em C que receba um  $n \geq 0$  e retorne o valor de seu fatorial,  $y = n!$ . Faça o programa em C para testá-la.
84. Faça uma função em C que receba três números inteiros e retorne o maior valor. Faça o programa em C para testá-la.
85. Faça uma função em C que receba três números inteiros e retorne o menor valor. Faça o programa em C para testá-la.
86. Faça uma programa para calcular as raízes de uma equação do segundo grau. O programa deve receber os coeficientes  $a, b, c$  da equação  $ax^2 + bx + c = 0$ , calcular  $x_1$  e  $x_2$  e mostrar o resultado. Crie uma função para calcular (e retornar) o valor de  $\Delta = b^2 - 4ac$  (ela deve receber os coeficientes e retornar  $\Delta$ ). Na função `main()`, ao calcular  $\sqrt{\Delta}$ , caso  $\Delta < 0$  deve-se apresentar uma mensagem dizendo que não há raízes reais.
87. Refaça agora o problema acima usando função. Faça uma função void para calcular as raízes de uma equação do segundo grau. A função deve receber os coeficientes  $a, b, c$  da equação  $ax^2 + bx + c = 0$  e um vetor  $2 \times 1$  vazio e calcular  $x_1$  e  $x_2$ , salvando-os no vetor recebido. Todas as saídas e entradas de dados devem se feitas na função `main()`. Atenção para o caso  $\Delta < 0$ .
88. Faça uma função para calcular a exponencial de uma número. A função deve receber dois floats ( $x$  e  $y$ ) e retornar o resultado, também float,  $z$ , sendo  $z = x^y$ . Faça o programa `main()` para testar a função. Os dados de entrada e saída devem se processados na função `main()`.
89. Faça uma função para calcular o valode  $\pi$  dado por:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots \quad (4)$$

A função deve receber o valor da precisão desejada e retornar o valor de  $\pi$  calculado. O somatório deve ser efetuado até a precisão desejada ser alcançada.

90. Faça uma função (e o programa `main()` para testá-la) que calcule e retorne o valor do somatório

$$S = \sum_{i=-n}^{i=m} \frac{2+i}{i-3}$$

com  $n$  e  $m$  inteiros positivos dados pelo usuário. Atenção para a exceção no cálculo da fração. Todos as ações de entrada e saída devem ser feitas na função `main()`.

91. Faça uma função (e o programa `main()` para testá-la) que receba um vetor  $v$  de inteiros e retorne o maior valor do vetor. A função `main()` deve mostrar o vetor e seu maior valor. Fica a cargo do programador definir os limites e condições do programa.



## 13 Exercícios Intermediários: Funções, Structs e Arquivos

92. Faça uma função em C para receber os coeficientes de uma equação do segundo grau e retornar o resultado (se for real). Se o resultado não for real, a função deve retornar um "flag" indicando. Use estrutura (pense na melhor forma de retornar os valores pedidos no exercício). Faça o programa `main()` para testar a função. Todas entradas e saídas devem estar na função `main()`.
93. Faça um programa em C com estruturas para armazenar dados de um aluno em uma banco de dados de uma escola. Os dados devem ser os dados pessoais (nome, cpf, data de nascimento, nome do pai e da mãe) e dados escolares (numero de matricula, escola anterior, série matriculada). Deve-se ter uma estrutura separada para endereço. Rode o programa lendo dados para a estrutura e depois mostrando-os.
94. Faça um programa para fazer operações com número complexos, usando uma estrutura. O programa deve ler um número complexo e fazer as operações (soma, multiplicação, e módulo). Rode o programa com exemplos numéricos.
95. Nesse exercício você deverá criar uma struct chamada Ponto, contendo apenas a posição  $x$  e  $y$  (inteiros) do ponto. Declare 2 pontos, leia a posição (coordenadas  $x$  e  $y$ ) de cada um e calcule a distância entre eles. Apresente no final a distância entre os dois pontos. Organize seu programa em funções.
96. Faça um programa em C para criar um arquivo de duas colunas (X-Y) correspondente a função  $y = f(x) = \sin(x)$ . O valor de  $x$  deve variar de  $-x_0$  até  $x_0$  ( $x_0$  dado pelo usuário) como intervalo de  $x_0/N$  com  $N$  dado pelo usuário. O arquivo salvo deve ser texto e deve ser salvo com o nome `seno1.m`. Depois de salvo você deve usar o Octave para plotar a função.
97. Melhore o programa anterior para que o arquivo criado possibilite que os dados sejam lidos no Octave como uma matriz  $M$ , de dimensão  $m \times 2$ , sendo  $m$  o número de linhas. Por exemplo se o seu programa gerar os dados:

-3.14	-0.00
-2.14	-0.84
-1.14	-0.91
-0.14	-0.14
0.86	0.76
1.86	0.96
2.86	0.28

Então seu arquivo deve fazer:

```
M=[-3.14    -0.00
-2.14     -0.84
-1.14     -0.91
-0.14     -0.14
 0.86      0.76
 1.86      0.96
 2.86      0.28];
```

para criar a matriz  $M$ . E depois você deve plotar a matriz  $M$  (com  $M(:,1)$  no eixo  $x$  e  $M(:,2)$  no eixo  $y$ ).

98. Faça uma função em C para preencher um vetor  $v$  com  $N$  números aleatórios.  $2 \leq N \leq 10$ . A função deve ser `void()` e receber o vetor e o valor  $N$ . Faça o programa `main()` para testar a função e imprimir o vetor.
99. Faça uma função em C que receba dois parâmetros: um vetor de tamanho  $N$  e o valor de  $N$ .  $N$  no máximo 20 e no mínimo 2. a função deve retornar o maior valor do vetor. Faça o programa `main()` para testar a função. Use a função do item anterior para gerar o vetor. O valor máximo retornado deve ser mostrado na função `main()`.
100. Faça uma função C que receba um vetor  $V$  de tamanho  $3 \leq N \leq 15$ , criado com inteiros aleatórios na função `main()` e retorne esse vetor ordenado. A função `main()` deve mostrar o vetor original e o vetor ordenado.

Compilado 24 de setembro de 2021 - 06:28:57