

7ª Lista de Exercícios

Assunto: Funções e passagem por referência com vetor e matriz (Tópico 7)

Essa lista de exercícios tem como objetivo principal desenvolver algoritmos a partir dos conteúdos abordados em sala de aula. Todos os exercícios também devem ser implementados em linguagem C. Nos programas que pedem para implementar apenas funções desenvolva também o programa principal (main) para testá-los.

1. Desenvolva uma função que receba dois pontos no plano cartesiano (x_1, y_1) , (x_2, y_2) , calcule e retorne a distância entre esses pontos.
2. Desenvolva uma função que receba um número n , calcule e retorne o somatório de 1 até n .
3. Desenvolva uma função que calcule se um número n é primo. Caso o número seja primo retorne 1 e 0, caso contrário.
4. Desenvolva uma função que ache as raízes de uma equação de segundo grau. Implemente uma função para calcular o valor do delta. Além disso, emita uma mensagem caso a equação não seja de segundo grau ou não tenha raízes reais.
5. Desenvolva uma função que calcule e retorne o fatorial de um número n .
6. Desenvolva uma função que calcule e retorne o n ésimo termo de fibonacci.
7. Desenvolva duas funções que recebam 3 números inteiros e retornem o Máximo Divisor Comum (MDC) e o Mínimo Múltiplo Comum (MMC) desses números. Crie um programa que leia 3 números inteiros e mostre o MDC e o MMC desses números.
8. Fazer uma função para calcular a raiz quadrada de um número positivo, baseado no método de aproximações sucessivas de Newton:
 - Seja $Y > 0$ o número e $N > 0$ a quantidade de aproximações.
 - A primeira aproximação para a raiz de Y é $X_1 = \frac{Y}{2}$.
 - As demais aproximações serão $X_{n+1} = \frac{X_n^2 + Y}{2X_n}$
9. Numa fábrica trabalham homens e mulheres divididos em três classes:
 - A os que fazem até 30 peças por mês.
 - B os que fazem de 31 a 35 peças por mês.
 - C os que fazem mais que 35 peças por mês.

A classe A recebe salário-mínimo. A classe B recebe salário-mínimo e mais 3% do salário-mínimo por peça acima das 30 iniciais. A classe C recebe salário-mínimo e mais 5% do salário-mínimo por peça acima das 30 iniciais. Desenvolva um programa com os seguintes requisitos:

- (a) Uma função que cadastre o nome, número de peças fabricadas por mês e o sexo de n funcionários. A quantidade de funcionários é definida pelo usuário.
- (b) Uma função que armazene em uma estrutura de dados o salário de todos os funcionários.
- (c) Uma função que calcule o valor total da folha de pagamento da fábrica.
- (d) Uma função que calcule a média de salários dos homens.
- (e) Uma função que calcule a média de peças fabricadas pelas mulheres em cada classe.
- (f) Uma função que exibe o nome, sexo e quantidade de peças fabricadas pelo funcionário de maior salário (suponha que não existe empate).

10. Utilizando funções, desenvolva um programa que exiba um menu com as seguintes opções:
- 1. $w = u + v$ (soma de vetores).
 - 2. $\langle u, v \rangle$ (produto vetorial).
 - 3. $u = \alpha * v$ (produto vetor por escalar).

O programa deve atender aos seguintes requisitos:

- (a) Os vetores são sempre gerados aleatoriamente dentro de um intervalo $[\min, \max]$ definido pelo usuário. Os valores gerados devem pertencer ao domínio dos números reais.
- (b) Os vetores não podem ser gerados antecipadamente, ou seja, eles devem ser gerados a partir da opção escolhida pelo usuário.
- (c) Para cada opção selecionado, os vetores gerados devem ser impressos. Da mesma forma, os vetores resultantes também devem ser impressos.
- (d) A consistência da operação deve ser avaliada, ou seja, o usuário deve ser obrigado a digitar as dimensões para vetores que permitam a execução da operação. O programa deve tratar tal situação.

11. Utilizando funções, desenvolva um programa que exiba um menu com as seguintes opções:
- 1. $C = A + B$ (soma de matrizes).
 - 2. A^t (matriz transposta).
 - 3. $u = A * v$ (produto matriz por vetor).
 - 4. $C = A * B$ (produto de matrizes).

O programa deve atender aos seguintes requisitos:

- (a) As matrizes e vetores são sempre gerados aleatoriamente dentro de um intervalo $[\min, \max]$ definido pelo usuário. Os valores gerados devem pertencer ao domínio dos números reais.
- (b) As matrizes e vetores não podem ser gerados antecipadamente, ou seja, eles devem ser gerados a partir da opção escolhida pelo usuário.

- (c) Para cada opção selecionado, as matrizes e vetores gerados devem ser impressos. Da mesma forma, os vetores e matrizes resultantes também devem ser impressos.
- (d) A consistência da operação deve ser avaliada, ou seja, o usuário deve ser obrigado a digitar as dimensões para vetores e matrizes que permitam a execução da operação. O programa deve tratar tal situação.
12. Desenvolva uma função a partir do pseudocódigo abaixo. O vetor deverá ser gerado com valores aleatórios no intervalo inteiro $[\text{Min}, \text{Max}]$ definido pelo usuário. Utilize outra função para iniciar o vetor. A dimensão do vetor também é definida pelo usuário. Logo, no programa principal, inclua uma chamada para a função que inicia o vetor e outra chamada para a função descrita no pseudocódigo. Você também irá precisar de uma função que imprima o vetor gerado e o vetor após a execução da função descrita abaixo. O que exatamente esse código está fazendo? Tente entender o seu funcionamento.

Algorithm 1 *Faco_algo_com_vetor1*

Require: $v[1 \dots \text{Size}]$

```

i ← 1
j ← 0
for i < Size do
  for j < Size − 1 do
    if  $v[j] > v[j + 1]$  then
      aux ←  $v[j]$ 
       $v[j] \leftarrow v[j + 1]$ 
       $v[j + 1] \leftarrow aux$ 
    end if
    j ← j + 1
  end for
  i ← i + 1
end for

```

13. Desenvolva um programa que utilize funções distintas para calcular a média, mediana e moda para um conjunto de dados. O conjunto de dados é formado por valores inteiros dentro de um intervalo definido pelo usuário. Dicas:
1. Utilize a função que inicializa o vetor do exercício anterior.
 2. A mediana é o valor intermediário. Para determinar esse valor, você precisará executar primeiro a função descrita no pseudocódigo do exercício anterior.
 3. A moda é o valor que ocorre com maior frequência entre os dados.
 4. O conjunto de dados deve conter pelo menos 100 elementos.
14. Desenvolva uma função a partir do pseudocódigo abaixo. O vetor deverá ser gerado com valores aleatórios no intervalo inteiro $[\text{Min}, \text{Max}]$ definido pelo usuário. Utilize outra função para iniciar o vetor. A dimensão do vetor também é definida pelo usuário. O valor de *value* é fornecido pelo usuário e deve estar dentro do intervalo $[\text{Min}, \text{Max}]$. Seu

código deve tratar quando o usuário entrar com um valor fora do intervalo. Logo, no programa principal, inclua uma chamada para a função que inicia o vetor, receba o *value* e chame a função descrita no pseudocódigo. Você também irá precisar de uma função que imprima o vetor gerado.

Algorithm 2 *Faco_algo_com_vetor2*

Require: $v[1\dots Size]$, $value$

```
 $i \leftarrow 0$ 
while  $i < Size$  and  $v[i] \neq value$  do
     $i \leftarrow i + 1$ 
end while
if  $v[i] = value$  then
    return  $i$ 
else
    return 0
end if
```

15. Desenvolva uma função a partir do pseudocódigo abaixo. O vetor deverá ser gerado com valores aleatórios no intervalo inteiro $[Min, Max]$ definido pelo usuário. Utilize outra função para iniciar o vetor. A dimensão do vetor também é definida pelo usuário. O valor de *value* é fornecido pelo usuário e deve estar dentro do intervalo $[Min, Max]$. Seu código deve tratar quando o usuário entrar com um valor fora do intervalo. Logo, no programa principal, inclua uma chamada para a função que inicia o vetor, receba o *value* e chame a função descrita no pseudocódigo. Você também irá precisar de uma função que imprima o vetor gerado.

Algorithm 3 *Faco_algo_com_vetor3*

Require: $v[1\dots Size]$, $value$

```
Faco_algo_com_vetor1( $v$ )
 $i \leftarrow 0$ 
 $j \leftarrow Size - 1$ 
while  $i < j$  do
     $m \leftarrow \lfloor (i + j) / 2 \rfloor$ 
    if  $value > v[m]$  then
         $i \leftarrow m + 1$ 
    else
         $j \leftarrow m$ 
    end if
end while
if  $v[i] = value$  then
    return  $i$ 
else
    return 0
end if
```

16. Desenvolva um algoritmo que verifica se uma data composta por 3 números inteiros (dia, mês e ano) é válida ou não. Para isso, implemente as seguintes funções:
 - (a) Verificar se um ano é válido (aceitar anos entre 0 e 9999).
 - (b) Verificar se um mês é válido (aceitar mês entre 1 e 12).
 - (c) Verificar se um ano é bissexto.
 - (d) Retornar quantos dias tem um mês (dias_do_mês) em um determinado ano.
 - (e) Verificar se um dia é válido (aceitar dia entre 1 e dias_do_mês).

17. Modularize o programa do jogo da velha definido na lista 5. Para isto crie as seguintes funções:
 - (a) Função para escrever o tabuleiro: void escreverTabuleiro(char matriz[][TAM]).
 - (b) Função para preencher o tabuleiro: void preencherTabuleiro(char matriz[][TAM], int movLinha, int movColuna).
 - (c) Função para verificar o estado do jogo (Xganhou, Oganhou, empate): void verificaJogo(char matriz[][TAM]).