

8ª Lista de Exercícios

Assunto - Passagem de Parâmetros por Referência (Tópico 8)

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. Não utilize variáveis globais.

1. **(Fácil)** Desenvolva um programa para testar as funções a seguir:
 - Uma função que receba dois números a e b , em seguida, faça a troca destes dois números. Dica: a e b devem ser passados por referência.
 - Uma função que receba dois números a e b , em seguida, decremente o primeiro e incremente o segundo. Dica: a e b devem ser passados por referência.
 - Uma função que receba o raio, perímetro e área de um círculo. Em seguida, calcule e retorne nas variáveis perímetro e área o respectivo perímetro e área do círculo. Dica: perímetro e área devem ser passados por referência.
 - Uma função que receba o lado, perímetro e área de um quadrado. Em seguida, calcule e retorne nas variáveis perímetro e área o respectivo perímetro e área do quadrado. Dica: perímetro e área devem ser passados por referência.
 - Uma função que receba os valores a , b e c passados por valor, receba também dois valores $x1$ e $x2$ passados por referência. Em seguida, calcule e retorne as duas raízes da equação do segundo grau nas variáveis $x1$ e $x2$.
2. **(Fácil)** Desenvolva um programa que receba coordenadas cartesianas digitadas pelo usuário e imprima seus valores convertidos para coordenadas polares. O usuário poderá digitar quantas coordenadas ele desejar. Utilize uma função que converte de coordenada cartesiana (x, y) para coordenada polar (r, s) .
3. **(Fácil)** Desenvolva um programa que primeiro utilize uma função para gerar um vetor com tamanho definido pelo usuário e valores aleatórios dentro de um intervalo $[min, max]$ determinado pelo usuário. Em seguida, uma outra função deve receber o vetor gerado, o tamanho do vetor e dois números *maior* e *menor*. Retorne nas variáveis *maior* e *menor* o menor e maior elemento do vetor. Dica1: *menor* e *maior* devem ser passados por referência. Dica2: use apenas uma estrutura de repetição para achar o menor e maior elemento.
4. **(Fácil)** Desenvolva um programa que utilize uma função para inverter os elementos de um vetor (com valores gerados usando a função do exercício anterior) sem utilizar um vetor auxiliar. Apenas o vetor fornecido à função deverá ser utilizado.
5. **(Fácil)** Desenvolva um programa para testar uma função com protótipo `void somabit(int b1, int b2, int *vaium, int *soma);`. A função recebe três bits (inteiros 0 ou 1) $b1$, $b2$ e $*vaium$. A função retorna um bit soma que armazena o resultado da soma dos três primeiros e o bit "vai-um" em $*vaium$.

6. **(Médio)** Desenvolva um programa que leia a quantidade total de segundos e converta para Horas, Minutos e Segundos. Imprima o resultado da conversão no formato HH:MM:SS. Para isso, utilize a função com protótipo *void converteHora(int total_segundos, int *hora, int *min, int *seg)*.
7. **(Médio)** Desenvolva um programa principal (*main*) que deverá ler o dia, mês e ano. Utilizando funções, o programa deverá verificar se os valores fornecidos formam uma data válida. Se for válida, o programa principal imprime a data como ilustrado a seguir:
Entrada: 01/11/2011
Saída: 01 de novembro de 2011
8. **(Médio)** Desenvolva um programa que leia a dimensão $M \times N$ de uma matriz e os valores *min* e *max*. Em seguida, o programa principal utiliza uma função para gerar a matriz $M \times N$ com valores aleatórios dentro de um intervalo $[min, max]$. Uma outra função deve receber a matriz gerada e retornar o maior elemento da matriz, o menor elemento da matriz e o valor médio das entradas da matriz.
9. **(Fácil)** Desenvolva um programa que tenha uma função que leia uma matriz de inteiros positivos de dimensões desconhecidas. O seu programa lerá os números e salvá-los, exceto quando ele for menor que 0. Se o número lido for -1, considere-o como um delimitador que diz o final da linha. Se for -2, a matriz completa foi lida. Além de ler a matriz, a função deve retornar a quantidade de colunas e a quantidade de linhas. As variáveis da matriz, do número de colunas e do número de linhas devem ser passadas por referência. Declare sua matriz utilizando um valor de tamanho máximo.
Exemplo de entrada: 3 5 6 -1 4 5 6 -2
Colunas: 3, Linhas: 2
10. **(Médio)** Faça um programa que leia uma quantidade N de palavras, sendo que N é fornecido pelo usuário. Elas devem ser salvas em uma matriz de modo que cada linha represente uma palavra. Faça uma função que receba como parâmetro um número inteiro, a matriz lida e um vetor de inteiros, todos passados por referência. O que a função deverá fazer é salvar no vetor os números das linhas em que a primeira letra da palavra é igual à última. O número inteiro passado como referência será um contador que dirá o tamanho desse vetor, ou seja, a quantidade de palavras em que a primeira e a última letra são iguais.
11. **(Fácil)** Faça um programa que calcule a pontuação de dois times após uma partida de futebol. Na função principal leia dois números inteiros que correspondem ao número de pontos dos dois times antes do jogo. Depois leia dois inteiros que representam o número de gols que cada time fez na partida. Você deverá fazer uma função separada que receba os pontos dos dois times por referência e os atualize de acordo com o resultado. Em caso de vitória são somados 3 pontos, em caso de empate é somado um ponto, e em caso de derrota, nenhum ponto é somado.
12. **(Médio)** Faça um programa que simule o sistema de um mercado. Determine um conjunto de produtos que terá no seu mercado e estabeleça dois vetores, um de preço e

um de quantidade no estoque, que contenham as informações dos produtos. Crie uma variável que represente o dinheiro no caixa com um valor inicial que desejar. Leia um inteiro na função principal que corresponde à ação que se deseja fazer (1 - adicionar ao estoque, em caso de compra de um fornecedor, e 2 - retirar do estoque, em caso de venda a um cliente). Faça uma função que implemente cada um desses casos. Ambas as funções devem receber os vetores de preço e quantidade, assim como o dinheiro no caixa, através de passagem por referência. Elas também devem imprimir um menu com os produtos, cada produto representado por um número. Depois que for lido o inteiro do produto escolhido, será lido a quantidade comprada ou vendida. Caso seja a função da opção 1, esse valor será adicionado à quantidade no estoque e o dinheiro em caixa será decrementado de acordo com o preço do produto e a quantidade comprada. Caso seja a opção 2, a quantidade no estoque será diminuída, mas o dinheiro no caixa será incrementado.

Exemplo de menu:

Escolha o produto desejado:

- (1) - Arroz
- (2) - Feijão
- (3) - Biscoito
- (4) - Molho de tomate