

UNIVERSIDADE POSITIVO
Disciplina de Algoritmos de Programação
Prof.^a Mariane Cassenote

Lista de Exercícios de Modularidade

1. Crie uma função que preencha um vetor de mil posições com valores aleatórios de 1 até 6. Conte quantas vezes cada número foi sorteado. Em seguida, exiba a porcentagem de vezes em que cada número foi sorteado.

2. Faça um programa para lançar uma moeda. Quando chamamos uma função, ela deve retornar cara ou coroa (pode ser representado como 0 e 1). Em outra função, faça n lançamentos de moedas, em que n é o valor que o usuário quiser, e mostre a porcentagem de vezes em que deu cara e em que deu coroa. Se você jogar a moeda 10, 100, 1000, um milhão de vezes...o que tende a acontecer?

3. Escrever um programa para somar duas matrizes de inteiros. Crie funções separadas para a) ler uma matriz; b) somar duas matrizes; c) imprimir uma matriz.

4. Criar uma função que receba um vetor de inteiros positivos, um vetor de caracteres e o tamanho (único) dos vetores. A função deve imprimir cada um dos caracteres do 2º vetor n vezes, onde n é o conteúdo da posição correspondente no vetor de inteiros.

5. Faça um programa que calcula o custo de um churrasco para n pessoas incluindo adultos e crianças. Cada criança maior do que 3 anos paga a metade. O usuário define os itens do churrasco e seus respectivos preços. O programa calcula a quantidade necessária de cada item conforme a quantidade de pessoas. O programa também calcula quanto cada pessoa terá que pagar para ir no churrasco. Suponha que cada adulto consome em média 250g de carne e 700ml de bebida. Suponha que as crianças consomem em média a metade do que um adulto consome.

6. Você foi contratado para desenvolver um programa que simule uma conta bancária. O programa deve permitir que o usuário realize três operações principais: depositar dinheiro na conta, sacar dinheiro da conta e verificar o saldo atual da conta. Para isso, você deve criar um programa em linguagem C que inclui funções para cada uma dessas operações.

Os requisitos do programa são:

- a. Crie uma função chamada “depositar” que permita ao usuário depositar uma quantia de dinheiro na conta bancária. A função deve receber o valor a ser depositado como argumento e atualizar o saldo da conta.

- b. Desenvolva uma função chamada “sacar” que permita ao usuário sacar uma quantia de dinheiro da conta bancária. A função deve verificar se há saldo suficiente na conta para realizar o saque e atualizar o saldo, se possível. Caso contrário, deve exibir uma mensagem de erro.
- c. Implemente uma função chamada “verificarSaldo” que retorne o saldo atual da conta bancária.
- d. No programa principal, permita que o usuário escolha entre as três operações: depositar, sacar ou verificar saldo. O programa deve continuar rodando até que o usuário escolha sair.
- e. Certifique-se de que o saldo da conta não possa se tornar negativo após um saque.

7. Você foi encarregado de desenvolver um programa que gere senhas seguras de acordo com critérios específicos. O programa deve ser capaz de gerar senhas com um comprimento definido e que atendam a requisitos de complexidade.

Os requisitos do programa são:

- a. Crie uma função chamada “gerarSenha” que recebe dois parâmetros: o comprimento desejado da senha e um valor indicando se a senha deve incluir caracteres especiais (como !, @, #, \$, etc.). A função deve retornar a senha gerada.
- b. A senha gerada deve conter uma combinação de letras maiúsculas e minúsculas, números e, se especificado, caracteres especiais.
- c. Certifique-se de que a senha gerada atenda ao comprimento desejado e aos critérios de complexidade.
- d. No programa principal, permita que o usuário insira o comprimento desejado da senha e escolha se deseja incluir caracteres especiais.
- e. Apresente a senha gerada ao usuário.