



Lista de exercícios

1. Funções

- 1.1. Escreva uma função que receba dois números como parâmetros e retorne o MMC (mínimo múltiplo comum) destes dois números.
- 1.2. Escreva uma função que passe por referência um vetor de caracteres. Esta função deve inverter o conteúdo do vetor. Ex: Se o usuário passar um vetor contendo a string "programação", ao fim da função o vetor deve conter "oacamargorp".
- 1.3. Escreva uma função do tipo void que receba dois parâmetros. O primeiro será passado por valor e o segundo por referência. Esta função deve perguntar para o usuário o valor de cada uma das infrações de trânsito que ele obteve no ano. O número de infrações deverá ser informado no primeiro parâmetro. A função então deverá atribuir à variável do segundo parâmetro quanto o usuário receberá de desconto caso pague as infrações até o dia do vencimento (o desconto é de 20% do total). Mostre no main o valor do desconto, bem como o valor a ser pago com desconto, e o valor a ser pago sem desconto.
- 1.4. Escreva um programa que tenha duas variáveis globais. Então crie uma função que peça para o usuário digitar nomes, idade, e salários de funcionários de uma empresa. Esta função deve atribuir à primeira variável global o nome do funcionário mais antigo, e para a segunda o nome do funcionário com maior salário. Mostre na função main estes dois nomes, após a chamada da função.
- 1.5. Escreva uma função que receba por valor um caractere, e por referência um vetor tipo char (string). Esta função deve concatenar esta string com o caractere, sem usar a função strcat. Dica: use o caractere '\0' para demarcar o fim de string.
- 1.6. Escreva uma função que recebe três números como parâmetro. Esta função deve retornar a razão entre o somatório dos dois elementos que são numericamente mais próximos e o outro elemento: Ex: Os números passados foram: 11,3,10. Cálculo: $(11+10)/3=7$.
- 1.7. Elabore 2 funções que tenham por finalidade inserir e remover números de um vetor. A primeira função deverá inserir um número informado como parâmetro na primeira posição vazia de um vetor. Considere que uma posição vazia do vetor contém o valor 0. A segunda

função deverá remover o último elemento de um vetor e retornar este elemento. Não poderão ser usadas variáveis globais.

- 1.8. Crie uma função que receba por referência um vetor do tipo char. A função deverá formatar a string dada como entrada de maneira que ao término da execução da função a primeira letra de todas as palavras sejam maiúsculas e as outras letras minúsculas. O vetor recebido por referência deverá ser acessado através de um ponteiro.
- 1.9. Elabore uma função que tenha por finalidade inserir números em um vetor. A função deverá receber por parâmetro o vetor e o número a ser inserido (e outros, a critério do aluno). Então esta função deverá inserir o número informado na primeira posição do vetor, deslocando todos os outros números uma posição adiante. Caso o vetor esteja lotado, a função deverá retornar o número que foi deslocado para fora do vetor, caso contrário deverá retornar 0. Não poderão ser usadas variáveis globais. Esta estrutura é chamada de fila.
- 1.10. Crie uma função que receba por referência um vetor de origem e um de destino. A função deverá pegar o último elemento do vetor de origem, removê-lo e então colocar este elemento na primeira posição vazia do vetor de destino. Considere que as posições vazias do vetor estão preenchidas com 0.
- 1.11. Crie um programa que gerencie mensagens de texto. Cada mensagem deve possuir um usuário de origem, um de destino, e o corpo da mensagem. Crie um menu com 4 opções. A primeira opção deverá possibilitar a criação de uma mensagem (com estes três campos). A segunda opção deverá verificar todas mensagens que um usuário dado como entrada enviou. A terceira opção deverá mostrar a caixa de entrada (todas mensagens recebidas) de um usuário dado como entrada, e a quarta opção deverá sair do programa.
- 1.12. Crie uma função que receba um valor inteiro como referência e retorne o resto da divisão deste número por 10. Altere também o valor da variável passada por referência, fazendo com que esta receba o resultado inteiro da divisão por 10. Faça um programa que imprima invertido os nomes dos algarismos de um número inteiro (Use a sua função).
- 1.13. Crie uma função que receba por referência um vetor e um número e por valor a posição do vetor. A função deverá realizar troca do elemento do vetor na posição recebida por parâmetro pelo número informado. Não utilize variáveis globais.

2. Arquivos

- 2.1. Escreva um programa que contenha uma função que grave em um arquivo .doc um milhão de caracteres de a até z aleatórios. Escreva outra função que busque dentro desse arquivo e mostre quantas ocorrências de um caractere especificado pelo usuário existem. O usuário deve especificar o nome do arquivo que será criado e consultado. Mostre uma mensagem de erro caso o arquivo não possa ser aberto.

- 2.2. Escreva uma função que receba por referência uma string contendo o nome de um arquivo e uma string de busca. Esta função deve retornar -1 caso haja erro na abertura do arquivo, caso contrário deve retornar o número de ocorrências da string de busca que foram encontradas dentro do arquivo. Mostre no main uma mensagem informando o usuário de quantas ocorrências foram encontradas, ou uma mensagem informando o erro.
- 2.3. Elabore um programa que gere 350 sequências de letras aleatórias de tamanho entre 10 e 25 cada. Armazene estas sequências uma em cada linha de um arquivo de nome escolhido pelo usuário e extensão .txt. Então crie um outro programa que pergunte para o usuário o nome de um arquivo e mostre na tela todas sequências de caracteres contidas no arquivo que possuem tamanho maior que 20.
- 2.4. Elabore um programa que ajude a realizar o controle de versão de arquivos. O programa deverá pedir para o usuário informar o nome de dois arquivos do tipo texto. Então mostre na tela o número da linha e o conteúdo desta linha no segundo arquivo que apresenta diferença com o conteúdo da mesma linha no primeiro arquivo. O programa também deverá criar um outro arquivo chamado "changelog.txt" que contém estas informações que foram mostradas na tela.
- 2.5. Elabore um programa que realize transações bancárias entre clientes. Mostre na tela um menu contendo 5 opções. Caso a primeira opção seja escolhida, deverá ser pedido para o usuário informar o nome, conta bancária e saldo inicial de um cliente, e então este cliente deve ser cadastrado. Caso a segunda opção seja escolhida, deverá ser pedido o número de duas contas bancárias e um valor. Caso as duas contas existam, deverá ser realizado uma transferência do valor informado da primeira conta para a segunda informada previamente. Caso a terceira opção seja escolhida, deverá ser feito um saque (valor removido) da conta informada. Caso a quarta opção seja escolhida, deverá ser feito um depósito. E caso a quinta opção seja escolhida, o programa deverá ser encerrado. O programa deverá carregar e salvar automaticamente os dados bancários dos clientes em um arquivo externo. Cada transação feita também deverá ser salva em um arquivo .txt contendo dados como conta de origem, destino (caso haja) e valor.

3. Structs

- 3.1. Elabore um programa que contenha uma struct global que possibilite o cadastro de animais de uma pet shop. Os animais devem ter os atributos nome, tipo do animal, dia, mês e ano de nascimento, e nome do dono.
- 3.2. Crie então uma função que possibilite o cadastro de um novo animal, pedindo para o usuário informar todos os atributos.
- 3.3. Crie também uma função que pergunte para o usuário uma string. Após, mostre na tela todos os animais que tenham em seu nome ou no nome do seu dono a string informada pelo usuário.

- 3.4. Crie uma função que dado o nome de um animal, mostre na tela a idade deste animal.
- 3.5. Escreva um programa com as seguintes características:
 - 3.5.1. O programa deve conter uma struct global para cadastrar terrenos retangulares loteados. Os terrenos devem ter os atributos comprimento, largura, valor e nome do proprietário do terreno.
 - 3.5.2. Na função main, pergunte quantos terrenos o usuário deseja cadastrar. Crie então uma função que recebe esse número como parâmetro, e então faça o cadastro dos terrenos dentro da função.
 - 3.5.3. Crie uma função que retorne para o main o índice do terreno que possui o maior custo-benefício, ou seja, possui o valor mais barato por área. Mostre no main os dados deste terreno.
- 3.6. Elabore um programa que mostre na tela um menu para controle do consumo de energia elétrica. Caso o usuário opte pela primeira opção, o programa deverá pedir o nome do usuário. Caso o cadastro não exista, peça seu endereço e crie seu cadastro. Caso o usuário já tenha sido criado, peça para ele inserir o valor de consumo de um mês e insira este valor em seu cadastro. Caso escolha a segunda opção do menu, o usuário deverá informar um nome, e o programa deverá mostrar na tela o total e a média de consumo cadastrado desta pessoa, bem como seu nome e endereço. Caso opte pela terceira opção do menu, encerre o programa. As informações cadastradas deverão ser automaticamente gravadas e carregadas de um arquivo externo. Utilize uma struct para armazenar o cadastro das pessoas.
- 3.7. Crie um programa para ajudar a gerenciar a biblioteca de jogos de múltiplos usuários. Crie um menu com cinco opções. Caso o usuário escolha a primeira opção, um novo jogo deverá ser cadastrado, devendo ser informado seu nome gênero e ano de criação. Caso a segunda opção seja escolhida, um novo usuário deverá ser cadastrado, devendo ser informado nome, id, e senha. Caso a terceira opção seja escolhida, o id de um usuário e o nome de um jogo deverão ser digitados. Caso ambos existam no cadastro, este jogo deverá ser inserida na biblioteca do usuário. Um usuário poderá possuir até 255 jogos no máximo. Caso a quarta opção seja escolhida, todos os jogos de um usuário dado como entrada deverão ser listados. Caso a quinta opção seja escolhida, o programa deverá ser encerrado. Os dados cadastrados deverão ser automaticamente carregados e salvos em um arquivo externo.
- 3.8. Crie um programa que auxilie na tarefa de procurar pares para cruza de cães de raça. O programa deverá possuir um menu. Caso a primeira opção seja escolhida, o usuário deverá cadastrar um novo cão fornecendo informações como nome, raça, sexo, e idade. Caso a segunda opção seja escolhida, o usuário deverá informar o nome de um cão, e caso este esteja cadastrado, o programa deverá mostrar os dados de todos os outros cães do sexo oposto e da mesma raça do cão informado. Caso a terceira opção seja escolhida, o programa deverá ser encerrado. As informações cadastradas deverão ser automaticamente

gravadas e carregadas de um arquivo externo. Utilize uma struct para armazenar o cadastro dos cães.

- 3.9. Seu Jatobá possui um pequeno comércio tem o costume de anotar no caderninho quando algum cliente compra fiado. Crie um programa que ajude o Seu Jatobá a gerenciar seu boteco. Mostre um menu na tela com três opções. Caso a primeira seja escolhida, peça para o usuário inserir o nome do cliente e valor da compra. Caso escolha a segunda opção, peça um nome, e mostre o total que este cliente deve para o Seu Jatobá. Caso a terceira opção seja escolhida, finalize o programa. Utilize uma struct para salvar os dados dos clientes. Salve e carregue automaticamente o conteúdo da struct em um arquivo externo.
- 3.10. Escreva um programa que tenha um menu com 5 opções. Caso seja escolhida a primeira opção o usuário deverá informar os dados de cadastro de um candidato à uma prova, que serão seu código de inscrição, cpf/rg, nome completo e situação de inscrição. Faça uma pergunta para o usuário perguntando se ele quer cadastrar o candidato como isento da taxa ou não (a situação deverá ser "aguardando" ou "isento", dependendo da resposta). Caso a segunda opção do menu seja escolhida, o usuário deverá informar o número de cadastro de um candidato. Caso este candidato exista e seu status seja "aguardando", modifique a situação de inscrição dele para "confirmado", informando que ele já pagou a taxa referente à inscrição. Caso a terceira opção seja escolhida, pergunte se o usuário deseja mostrar a lista dos "confirmados" ou dos "isentos", e a seguir, mostre a lista na tela de todos candidatos do critério escolhido. Caso a quarta opção seja escolhida, peça o nome de um arquivo para o usuário e grave a lista de todos candidatos em um arquivo de texto. Caso a quinta opção seja digitada, o programa deverá ser encerrado. As informações dos candidatos deverão ser gravadas em uma struct. As informações da struct deverão ser carregadas e gravadas automaticamente ao início e término do programa.

4. Recursividade

- 4.1. Escreva um programa que através da chamada de uma função recursiva retorne a multiplicação dos dois números passados como parâmetro para a função. Esta função deve fazer a multiplicação através de somas sucessivas. Ex: $5 \times 3 = 5 + 5 + 5$.
- 4.2. Escreva uma função recursiva que retorne o resto da divisão entre dois números passados como parâmetro para a função. A função não pode utilizar o operador resto para fazer este cálculo.
- 4.3. Escreva uma função calcule recursivamente e retorne um elemento de uma PA. O dados como primeiro elemento, razão e o número do elemento desejado deverão ser informados pelo usuário.
- 4.4. Escreva uma função recursiva que inverta uma string passada por parâmetro.