UFSJ UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

EXERCÍCIOS LINGUAGEM C

DISCIPLINA: Algoritmo e Estrutura de Dados II

PROFESSOR: Rafael Sachetto Oliveira

Exercício 1

Faça um programa que receba um número inteiro e verifique se o número fornecido é primo ou não. Imprima uma mensagem de número primo ou número não primo.
Obs. Um número é primo se este é divisível apenas por um e por ele mesmo.

Exercício 2

Faça um programa que receba um número N e imprima os divisores pares desse número e some os divisores pares.

Exercício 3

Faça um programa que receba dois números e tenha um "menu" (imprimir na tela o menu para o usuário escolher) de opções da seguinte forma:

1) Calcula x^y
2) Calcula a multiplicação do primeiro pelo segundo;
3) Calcula a divisão do primeiro pelo segundo

Imprima os resultados de acordo com a opção escolhida pelo usuário.

Exercício 4

Elabore um programa que preencha uma matriz 4x4 com números inteiros e mostre quantos elementos são maiores que 30 e, em seguida, monte um vetor (use alocação dinâmica de memória) que contenha as posições dos números maiores que 30, no seguinte formato l*c, ou seja, se existe um elemento maior que 30 na posição (1, 2), deve constar no vetor uma string "1*2".

Exercício 5

Pedir para o usuário entrar com dois tamanhos de vetor, gerar os dois vetores com números inteiros quaisquer e exibi-los. Depois da exibição de cada um dos vetores, exibir a intercalação dos mesmos.

Exercício 6

Crie um programa capaz de ler os dados de uma matriz quadrada de inteiros. Ao final da leitura o programa deverá imprimir o número da linha que contém o menor dentre todos os números lidos.

Exercício 7

Faça o programa que apresenta a seguinte saída, perguntando ao usuário o número máximo (no exemplo, 9). Este número deve ser sempre ímpar.

```
1 2 3 4 5 6 7 8 9
  2 3 4 5 6 7 8
    3 4 5 6 7
      4 5 6
        5
```

Exercício 8

Crie um programa capaz de ler dois nomes de pessoas e imprimi-los em ordem alfabética.

Exercício 9

Faça um programa que crie um vetor de pessoas. Os dados de uma pessoa devem ser armazenados em um variável do tipo struct. O programa deve permitir que o usuário digite o nome de 3 pessoas e a seguir imprimi os dados de todas as pessoas. A struct deve armazenar os dados de idade, peso e altura.

Exercício 10

Faça uma função que retorne a posição de um dado caracter dentro de uma string.

Exercício 11

Faça um programa que lê um string qualquer de no máximo 80 caracteres e imprime:
- Quantos caracteres tem o string;
- Quantos caracteres são de pontuação;
- Quantos caracteres são números;
- Quantos caracteres são minúsculas.

Exercício 12

Faça um programa que lê um string contendo palavras separadas por um espaço em branco cada e as imprime uma abaixo das outras.

Exercício 13

Faça um programa que pregunta o nome, o endereço, o telefone e a idade de uma pessoa e monta um string com a seguinte frase: "Seu nome é ..., você tem ... anos, mora na rua ... e seu telefone é ... ."

Exercício 14

Faça uma função que recebe um string como parâmetro e imprime quantas palavras (separadas por espaços em branco) o mesmo contém.

Exercício 15

Implemente um função que faça a mesma coisa que a função "strcpy".

Exercício 16

Faça um programa que solicita um número inteiro e soletra o mesmo na tela. Ex: 124: um, dois, quatro

Exercício 17

Escreva uma função que recebe um string e um carácter como parâmetro e remove todas as ocorrências do caracter do string.

Exercício 18

Escreva uma função em "C" que receba um string um carácter e o índice de uma posição do string como parâmetro e insira o carácter na posição "empurrando" todos os demais para o lado.

Exercício 19

Faça uma função em "C" que recebe um string como parâmetro e devolve um ponteiro para o primeiro carácter branco encontrado.

Exercício 20

Faça uma rotina que recebe como parâmetro um string contendo um número e um inteiro indicando a base na qual o número está expresso, retornando o seu valor em decimal Ex: conv2dec("345", 8) => 229

Fazer após os vídeos sobre listas e pilhas

Exercício 1
Crie o código-fonte que implemente uma lista simplesmente encadeada e organize-o, criando um arquivo .h e um arquivo .c (lista.h e lista.c). Crie um outro arquivo, testes.c, que conterá a função main. Na função main, efetue testes na lista: criação de uma lista, inserção de vários elementos na lista, remoção. Após cada operação, imprima o conteúdo da lista para ver o resultado.

Exercício 2
Modifique a struct que define o Item da lista do exercício anterior, incluindo número de matrícula, nome, identidade e cpf. Repita os testes do exercício 1, levando em consideração os novos componentes.

Exercício 3

Criar uma função para pesquisar uma chave na lista, ou seja, dada uma chave, ela retornará o item ou NULL se não for encontrado. A função pode receber a chave ou um item que esteja apenas com a chave preenchida (esta última opção é melhor em termos de manutenibilidade). Apresente testes para essa função.

Exercício 4

Implemente uma função para apagar o item pela chave. Por exemplo, em uma lista que contém os elementos de chaves 1 2 3 4 5 6, permitir que seja feita a remoção do item de chave 5. A função pode receber a chave ou um item que esteja apenas com a chave preenchida (esta última opção é melhor em termos de manutenibilidade). Apresente testes para essa função.

Exercício 5

Implemente uma função que retorna uma sub-lista da lista passada como parâmetro. Ou seja, a função irá receber como parâmetros uma lista, um índice inicial e um índice final e retornara uma sub-lista com tamanho (índice final – índice inicial). Sua função deverá fazer os tratamentos de erros apropriados.

Exercício 6

Implemente uma função que receba duas listas duplamente encadeadas e retorne a concatenação intercalada dessas listas. Por exemplo, se a lista 1 contém A B C D E e a lista 2 contém F G H, a lista retornada deve ser A F B G C H D E.

Exercício 7

Implemente uma função que, dada uma pilha, retorne uma lista com os n elementos desempilhados, ou seja, a função deve receber o valor de n e desempilhar n elementos, retornando-os.

Fazer após os vídeos sobre pesquisa e ordenação

Exercício 1

Fazer um programa que leia um conjunto de n valores numéricos (n<=20), ordenar e imprimir estes valores em ordem decrescente.

Exercício 2
Implemente a ordenação por Seleção e elabore uma forma de mostrar, com exemplos, que esse método não é estável.

Exercício 3
Implemente o algoritmo de ordenação por inserção e modifique-o, de forma que ele utilize a busca binária para encontrar a posição de inserção de um elemento no vetor destino.