

Lista de Exercícios 2

Exercícios usando vetores e matrizes, mas não usam funções

1. Faça um programa que lê as notas de n alunos, cada nota é um inteiro entre 0 e 100, e imprima a quantidade de vezes com que apareceu cada nota.
2. Faça um programa que lê um número n , depois lê n números reais e o programa imprime, em relação aos números reais, o maior valor, o menor valor, a média e quantos números foram maiores ou iguais a média.
3. Faça um programa que lê um número n , depois lê n números reais e em seguida imprime os números reais lidos em ordem não-crescente (i.e., os maiores primeiro).
4. Faça um programa que lê uma cadeia de caracteres e transforma a cadeia de caracteres na mesma cadeia, mas com as letras transformadas para minúsculas. Em seguida imprime a cadeia de caracteres resultante. Depois, faça o mesmo com letras maiúsculas.
5. Faça um programa que lê uma cadeia de caracteres e troca os caracteres acentuados pelos correspondentes caracteres sem acento. Em seguida, imprime a cadeia de caracteres resultante.
6. Faça um programa que lê duas cadeias de caracteres e diz qual delas vem primeiro na ordem lexicográfica (ordem do dicionário). Para isso, use a solução obtida dos dois exercícios anteriores.
7. Uma frase/palavra é dita ser palíndromo, se ela é a mesma se lida da esquerda para a direita é a mesma que lida da direita para a esquerda. Faça um programa que lê uma cadeia de caracteres e diz se a cadeia é palíndromo ou não.
8. O imperador romano Júlio César usava um método de criptografia para mandar mensagens criptografadas para seus generais. Este método considerava a ordem dos caracteres no alfabeto e trocava cada letra pela k -ésima letra seguinte a letra. Por exemplo, se $k = 3$, então a ocorrência da letra 'A' era trocada pela letra 'D'. A letra 'B' era trocada pela letra 'E', assim por diante. Esta substituição era feita de maneira cíclica, que neste caso a letra 'Z' era trocada pela letra 'C'. Faça um programa que lê um inteiro k e uma cadeia de caracteres e imprime a correspondente mensagem criptografada pela cifra de César com parâmetro k .
9. Faça um programa que lê um inteiro k e uma cadeia de caracteres criptografada pela cifra de César com parâmetro k (veja exercício anterior) e imprime a mensagem decriptografada.
10. Faça um programa lê o nome, idade e salário de n pessoas (n lido). O programa deve ordenar e imprimir os dados destas pessoas ordenados pelo nome, pelo salário e pela idade (de maneira não decrescente).
11. Faça um programa que lê uma matriz 3×3 e imprime o determinante da matriz.
12. Escreva um programa que imprime um Triângulo de Pascal de ordem n (n lido). O programa deve usar apenas um vetor.
13. Uma matriz quadrada inteira é chamada de *quadrado mágico* se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e as soma dos elementos das diagonais principal e secundária são todos iguais. Exemplo: As matrizes abaixo são quadrados mágicos:

$$\begin{bmatrix} 3 & 4 & 8 \\ 10 & 5 & 0 \\ 2 & 6 & 7 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Faça um programa que verifica se uma dada matriz quadrada lida é ou não um quadrado mágico.

14. Faça um programa que lê um número n e em seguida lê dados de n exames: A_1, A_2, \dots, A_n . Cada exame A_i tem o nome do paciente, $Nome_i$; uma taxa de colesterol X_i (número entre 0 e 100); uma taxa de proteínas e B_i (número entre -10 e 10) e uma taxa de torotinas T_i (um valor entre 5000 e 10000). Todos os valores numéricos são reais.
Pessoas normais tem valores de $X_i \in [30, 50]$, $B_i \in [-3, 7]$, $T_i \in [6500, 8000]$. Os pacientes que tem taxa de colesterol abaixo (acima) do normal, têm hipoxolesterol (hipercolesterol). Os que tem taxa de proteínas abaixo (acima) do normal, têm hipoproteína (hiperproteína). As que tem taxa de torotinas abaixo (acima) do normal, têm hipotorotinas (hipertorotinas). Seu programa deve imprimir todos os pacientes que apresentam alguma doença. Para cada uma, deve se
 - (a) imprimir qual o tipo de doença que ela apresenta.
 - (b) se ela apresentar duas doenças, deve se escrever que o estado do paciente é *grave*.
 - (c) se ela apresentar três doenças, deve se escrever que o estado do paciente é *muito grave*.

15. Uma rede elétrica é formada por N pontos, $\{0, \dots, N-1\}$ (para facilitar o mapeamento). Alguns pontos são ligados por cabos de energia, sempre por pares. As ligações são apresentadas através de pares (i, j) , dizendo que existe um cabo que liga o ponto i ao ponto j . Sabe-se que o ponto 0 é chamado de gerador (de energia). Faça um programa que lê um valor N , $N \leq 20$; lê N pares (i, j) , indicando a existência de uma ligação elétrica entre o ponto i e o ponto j e lê uma sequência de inteiros terminando com -1 . Cada inteiro desta sequência anterior ao -1 é um ponto (valor em $\{0, \dots, N-1\}$). Para cada ponto p da sequência, seu programa deve dizer a sequência de pontos que ligam o gerador (ponto 0) até o ponto p (começando com o gerador até chegar em p). Caso não haja conexão do ponto 0 ao ponto p através de ligações, seu programa deve dizer que o ponto p não está conectado à rede elétrica do gerador.

Exercícios sobre funções

16. Faça três funções fatorial, com os comandos **for**, **while** e **do-while**.
17. Faça uma função que tenha como parâmetro uma temperatura em graus Fahrenheit e retorne a temperatura em graus Celsius. Obs.: $(C = 5/9 \cdot (F - 32))$.
18. Faça uma função para verificar se um número é primo e que verifica apenas uma vez os múltiplos de 2 e de 3, testando a divisão apenas para os demais números.
19. Faça uma rotina que tem como parâmetros, um vetor de números reais e o número de elementos que estão no vetor. A rotina deve ordenar o vetor em ordem crescente.
20. Faça uma rotina que tem como parâmetros, um vetor de números reais e o número de elementos que estão no vetor. A rotina deve ordenar o vetor em ordem decrescente.
21. Faça uma função que tem como parâmetros, um inteiro n , um vetor de números reais V já ordenado, e um número real x . Sua rotina deve usar a estratégia de busca binária (como o exemplo da lista telefônica visto em aula) para localizar a posição de x no vetor. Sua função deve devolver a posição/índice onde o elemento x está no vetor. Caso o elemento x não esteja no vetor, sua função deve devolver o valor -1 .
22. (*K-ésimo*) O problema do k -ésimo consiste em, dado um vetor com n elementos, encontrar o k -ésimo menor elemento do vetor.
23. Faça um programa que tem as seguintes definições:

```
#define MAXALUNOS 10
#define MAXNOME 100

/* tipo para uma data dada pelo dia mes e ano */
typedef struct {
    int dia,mes,ano;
} tipodata;

/* tipo para um aluno, que contém info. sobre ra, nome, cr e data de nasc. */
typedef struct {
    int ra;
    char nome[MAXNOME];
    float cr;
    tipodata datanasc;
} tipoaluno;

/* tipo para um cadastro, que armazena até MAXALUNOS alunos. A quantidade
dos alunos é dada pelo campo quant e os alunos são armazenados no campo
aluno[0], aluno[1], ... */
typedef struct {
    int quant;
    tipoaluno aluno[MAXALUNOS];
} tipocadastro;
```

Seu programa deve ter as seguintes rotinas:

- Uma rotina que recebe um cadastro e atualize ele ordenando os alunos existentes pelo maior CR primeiro.
- Uma rotina que recebe um cadastro e atualize ele ordenando os alunos existentes pelos alunos mais novos primeiro.

- Uma função que tem como parâmetro o cadastro e retorna a posição do aluno com o maior CR.
 - Uma função que tem como parâmetro o cadastro e um inteiro que é um RA. A função deve remover do cadastro o aluno que tem aquele RA. Se o aluno com o RA for encontrado, ele deve ser removido e a função devolve 1. Caso contrário (o aluno com aquele RA não existe no cadastro) a função devolve 0.
24. Faça funções que trabalham com vetor de números reais. Cada função deve ter dois parâmetros, um é o vetor e outro é o número de elementos no vetor. Faça funções (uma para cada operação) para devolver o desvio padrão, variância, média, maior elemento, menor elemento, moda, soma de todos os elementos, etc.
 25. Faça um programa que implemente um cadastro de pessoas, como no item anterior, mas com as seguintes informações: Nome, RG, CPF, endereço, CPF da mãe, CPF do pai, data de nascimento, etc. Faça rotinas para trabalhar com este cadastro, como inserção de pessoas, remoção pelo nome, remoção/atualização pelo RG/CPF/nome, impressão de toda a ascendência de uma pessoa (i.e., imprimir seu nome, nome dos pais, nome dos avós, ...), etc.
 26. Faça rotinas e funções para manipular matrizes com dimensões $M \times M$ (onde M é um valor definido por `#define`) para ler uma matriz, imprimir uma matriz, calcular determinante, obter matriz transposta, obter matriz inversa, etc.
 27. Os números de fibonacci n_1, n_2, \dots são definidos da seguinte forma:

$$\begin{aligned} n_0 &= 0, \\ n_1 &= 1, \\ n_i &= n_{i-1} + n_{i-2}, \quad i \geq 2. \end{aligned}$$

Faça uma função que devolve o valor do n -ésimo número de fibonacci.

28. Faça funções que devolvem verdadeiro ou falso (1 ou 0), cada uma tem apenas um parâmetro inteiro, e elas verificam se um número é: primo, par, quadrado perfeito, etc (faça uma função para cada uma dessas verificações).
29. Implemente conjuntos através de vetores e faça os seguintes procedimentos:
 - (a) Procedimento chamado *conjunto*, com dois parâmetros, um vetor e um inteiro positivo $n \geq 0$, indicando a quantidade de elementos no vetor. O procedimento deve remover os elementos duplicados do vetor e atualizar o valor de n .
 - (b) Função chamada *Pertence* com três parâmetros, com a seguinte “cara”:
`int pertence(int *v, int n, int x)`
 os elementos do conjunto estão no vetor v que contém n elementos. A função devolve 1 se o elemento x pertence ao conjunto e 0 caso contrário.
 - (c) Procedimento chamado *Intersecao* com seis parâmetros, com a seguinte “cara”:
`void intersecao(int *v1, int n1, int *v2, int n2, int *v3, int *n3)`
 O vetor v_1 (v_2) contém n_1 (n_2) elementos. O vetor v_3 receberá a interseção dos elementos de v_1 e v_2 e n_3 deve retornar com a quantidade de elementos em v_3 (vamos supor que cada vetor só contém elementos distintos).
 - (d) Procedimento chamado *Uniao* com seis parâmetros, com o seguinte cabeçalho:
`void uniao(int v1, int n1, int v2, int n2, int v3, int *n3)`
 O vetor v_1 (v_2) contém n_1 (n_2) elementos. O vetor v_3 receberá a união dos elementos de v_1 e v_2 e n_3 deve retornar com a quantidade de elementos em v_3 (vamos supor que cada do vetor só contém elementos distintos).
30. Descreva um procedimento, com parâmetro inteiro positivo n e dois outros parâmetros b e k que devem retornar valores inteiros positivos. Os valores a serem retornados em b e k são tais que b seja o menor valor inteiro tal que $b^k = n$.