

MC 102 - Algoritmos e Programação de Computadores

Primeiro Semestre de 2014

Lista de Exercícios 3

1. Escreva um programa que imprima todas as potências: $2^0, 2^1, \dots, 2^{10}, 3^0, \dots, 3^{10}, \dots, 10^{10}$.
2. Escreva um programa que imprima os valores de $n!$ para $n = 1, \dots, 20$.
3. Escreva um programa que lê um número inteiro n e imprime o primeiro número da série de fibonacci que é maior ou igual a n .
4. Escreva um programa que lê um inteiro n e imprime o maior número primo que é menor ou igual a n .
5. Escreva um programa que lê um valor inteiro n (positivo ou negativo) e imprime todos os valores a e b (inclusive negativos) tais que $a * b = n$.
6. Escreva um programa que lê um número inteiro n e depois lê n inteiros e guarda-os em um vetor. Além disso, o programa lê dois números inteiros i e j que são índices do vetor (ou seja $0 \leq i, j \leq n$). O programa deve trocar os elementos das posições i e j entre si e imprimir o vetor na tela.
7. Faça um programa que lê um vetor de 30 inteiros e guarda o vetor na ordem inversa que foi lido em um outro vetor de saída.
8. Escreva um programa que lê um inteiro n e depois lê n inteiros e guarda-os em um vetor. O programa deve imprimir a soma dos números pares do vetor.
9. Faça um programa que lê um inteiro n e depois lê n números reais. O programa deve imprimir o vetor e o desvio padrão dos números do vetor usando a seguinte fórmula:

$$\sqrt{\frac{1}{n-1} \left(\sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 \right)}$$

10. Modifique cada um dos seguintes algoritmos vistos em sala, para que estes ordenem um vetor em ordem **decrecente** de valor:
 - selectionSort
 - insertionSort
 - BubbleSort
11. Escreva o algoritmo de busca sequencial. Descreva a busca pelos elementos 31, 41, e 128 no vetor (31, 41, 59, 26, 41, 58, 15, 19). Quantos passos (acessos ao vetor) foram realizados em cada uma dessas buscas?

12. Use cada um dos algoritmos abaixo e mostre passo-a-passo, como visto em aula, os passos para ordenar o vetor (31, 41, 59, 26, 41, 58, 15, 19):
 - selectionSort
 - insertionSort
 - BubbleSort
13. Escreva o algoritmo de busca binária. Após a ordenação do vetor (31, 41, 59, 26, 41, 58, 15, 19), mostre os passos da busca binária pelos elementos 15, 26, 41, e 58. Quantos passos (acessos ao vetor) foram realizados em cada uma dessas buscas?
14. Crie um programa que lê um inteiro n e depois lê n inteiros e guarda-os em um vetor. Seu programa deve determinar o número total de inversões no vetor **vet**. Uma inversão existe quando um elemento em uma posição $i < j$ é tal que $vet[i] > vet[j]$. Por exemplo, no vetor (10, 4, 6, 1, 2) existem 4 inversões para o número 10, 2 inversões para o número 4, 2 inversões para o número 6, nenhuma inversão para 1, e nenhuma para o 2. Portanto o total de inversões é 8. Modifique o algoritmo *bubbleSort* para computar o número de inversões em um vetor.
15. Faça um programa que leia uma matriz no máximo 30×30 e imprima a sua transposta.
16. Faça uma programa que lê um inteiro n e uma matriz quadrada de no máximo 30×30 , além de dois inteiros i, j , que são índices de linha e coluna respectivamente da matriz. O programa deve imprimir a matriz lida e a soma total dos elementos da linha i com os elementos da coluna j da matriz.
17. Faça uma programa que lê um inteiro n e uma matriz quadrada de no máximo 30×30 , além de dois inteiros i, j . O programa deve trocar os conteúdos das linhas i e j desta matriz entre si. Esta é uma operação de matrizes conhecida como permutação de linhas.
18. Faça uma programa que lê um inteiro n e uma matriz quadrada de no máximo 30×30 , além de dois inteiros i, j . que são índices de linha e coluna respectivamente da matriz. O programa deve então calcular e imprimir o produto interno do vetor linha i com o vetor coluna j da matriz.
19. Faça um programa que leia uma matriz quadrada no máximo 30×30 e dois inteiros i, j . O programa deve então salvar em um vetor de saída a soma do vetor correspondente a linha i da matriz com o vetor correspondente a coluna j .da matriz. O programa deve então imprimir este vetor soma.
20. Escreva um programa que lê m inteiro n e uma matriz quadrada de no máximo 30×30 e verifica se esta é simétrica ou não.
21. Uma matriz quadrada de inteiros é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna, a soma dos elementos da diagonal principal e da diagonal secundária são todos iguais. A matriz abaixo é um exemplo de quadrado mágico:

3	4	8
10	5	0
2	6	7

Faça um programa que lê uma matriz quadrada e determina se ela é um quadrado mágico.

22. Escreva um programa que leia duas palavras do teclado e determine se a segunda é um *anagrama* da primeira. Uma palavra é um anagrama de outra se todas as letras de uma ocorrem na outra, *em mesmo número, independente da posição*. Exemplos: ROMA, MORA, ORAM, AMOR, RAMO são anagramas entre si.
23. Faça um programa que lê um texto T e uma palavra p do teclado. Em seguida o programa deverá imprimir todas as posições onde ocorrem a palavra p em T .
Se por exemplo $T = \text{"duas bananas e 4 abacates. Nao haverá mais bananas."}$, e $p = \text{"bananas"}$, então o programa deveria imprimir 5 e 43.
24. Escreva um programa que lê uma string de até 50 caracteres, e imprime "Palindromo" caso a string seja um palindromo e "Nao Palindromo" caso contrário. OBS: Um palindromo é uma palavra ou frase, que é igual quando lida da esquerda para a direita ou da direita para a esquerda (assuma que só são usados caracteres minúsculos e sem acentos. Espaços em brancos devem ser descartados). Exemplo de palindromo: saudavel leva duas.
25. Historicamente César foi o primeiro a codificar mensagens. Ele reorganizava o texto de suas mensagens de maneira que o texto parecia não ter sentido. Cada mensagem sempre possuía uma contagem de letras cujo total equivalia a um quadrado perfeito, dependendo de quanto César tivesse que escrever. Assim, uma mensagem com 16 caracteres usava um quadrado de quatro por quatro; se fossem 25 caracteres, seria cinco por cinco; 100 caracteres requeriam um quadrado de dez por dez, etc. Seus oficiais sabiam que deviam transcrever o texto preenchendo as casas do quadrado sempre que uma mensagem aleatória chegasse. Ao fazerem isso, podiam ler a mensagem na vertical e seu sentido se tornaria claro.

Escreva um programa que lê o tamanho de uma string e a string. Depois o programa escreve a mensagem decifrada.

Exemplo:

36

MEEUMOCSHMSC1T*AGUOA***L2****T*****A

Esta mensagem pode ser transcrita em um quadrado perfeito 6x6.

M	E	E	U	M	O
C	S	H	M	S	C
1	T	*	A	G	U
0	A	*	*	*	L
2	*	*	*	*	T
*	*	*	*	*	A

Lendo cada coluna da matriz (desconsiderando o char '*'), a saída deverá conter:

MC102 ESTA EH UMA MSG OCULTA.

26. Sudoku é jogado numa malha de 9x9 quadrados, dividida em sub-malhas de 3x3 quadrados, chamada "quadrantes". O objetivo do jogo é preencher os quadrados com números entre 1 e 9 de acordo com as seguintes regras:

- Cada número pode aparecer apenas uma vez em cada linha.
- Cada número pode aparecer apenas uma vez em cada coluna.
- Cada número pode aparecer apenas uma vez em cada quadrante.

Exemplo:

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

Escreva um programa que lê um jogo de Sodoku (matriz 9x9, toda preenchida com números de 1 a 9) e verifica se é um jogo válido ou não. Um jogo válido respeita as três regras acima.

27. Refaça os algoritmos de busca sequencial e busca binária vistos em aula assumindo que o vetor possui chaves que podem aparecer repetidas. Neste caso, você deve armazenar em um outro vetor todas as posições onde a chave foi encontrada e imprimir este vetor na tela.