

**Universidade Federal do Ceará**  
**Campus de Quixadá**  
**Sistemas de Informação**  
**Fundamentos de Programação Computacional**  
**Professor Ricardo Reis**  
**Lista de Exercícios: Vetores e Strings**

Para os problemas que seguem determine a solução em forma de algoritmo e em seguida implemente-o em linguagem C

## UTILIZANDO VETORES

- Dado um vetor de números inteiros, determinar,
  - A soma dos elementos.
  - A média dos elementos.
  - A soma dos elementos pares subtraída da soma dos elementos ímpares.
  - Os valores máximo e mínimo entre seus elementos.
  - Os dois elementos de maior valor presentes.
- A *Busca Linear* é um processo de busca de um elemento  $x$  em um vetor  $L$  que testa sequencialmente cada elemento de  $L$  e encerra quando  $x$  é encontrado (busca com sucesso) ou quando o final de  $L$  é extrapolado (busca sem sucesso). Dados como entrada um vetor de números  $L$  e um número  $x$ , determinar utilizando busca linear se  $x$  está ou não presente em  $L$ .
- Reorganizar as chaves de um vetor de entrada de forma que elas fiquem na ordem inversa da original. Por exemplo, se  $x = [4, 9, 11, 3]$  então  $x$  deverá se tornar  $x = [3, 11, 9, 4]$ .
- Rotacionar à direita* um vetor significa colocar seus elementos uma posição adiante com exceção do último elemento que é transferido para a primeira posição. *Rotacionar à esquerda* um vetor significa colocar seus elementos uma posição para trás com exceção do primeiro elemento que é transferido para a última posição. Construir separadamente as rotações à direita e à esquerda para um vetor de inteiros dado como entrada.
- Dados dois vetores  $A$  e  $B$  de entrada, imprimir elementos que os dois possuem em comum. Por exemplo, se  $A = \{1, 7, 2, 9\}$  e  $B = \{7, 6, 9, 53, 13\}$  então deverá ser impresso na saída  $\{7, 9\}$ .
- A *ordenação crescente por seleção* é um algoritmo utilizado para classificar os elementos de um vetor. Ela consiste em visitar sequencialmente as  $n - 1$  primeiras posições de um vetor de comprimento  $n$  e em cada visita buscar pelo menor elemento no sub-vetor formado entre a posição visitada e a última posição do vetor e então trocá-lo com o elemento da posição visitada. Utilizando este algoritmo classificar um vetor de inteiros dado como entrada.
- A *mediana* de um conjunto finito de números é um elemento deste conjunto cuja quantidade de elementos menores ou iguais a ele é no máximo uma unidade a menos que os elementos maiores que ele. Dado um conjunto de entrada  $Q$  em forma de vetor, determinar sua mediana.
- Disponer em um vetor os  $n$  primeiros inteiros primos, dado  $n$  como entrada.
- Um número é dito *pandigital* se seus dígitos são todos distintos entre si. Construir função que determine se um número passado como argumento é ou não pandigital.
- Determinar o maior número natural primo pandigital (ver problema-9) que existe.
- Um polinômio  $P(x)$  é definido como a somatória,

$$P(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + \dots + a_n \cdot x^n$$

$$= \sum_{k=0}^n a_k \cdot x^k$$

onde  $\{a_0, a_1, a_2, \dots, a_n\}$  são denominados *coeficientes do polinômio*,  $n$  o grau do polinômio e  $x$  a *variável independente*. Avaliar um polinômio  $P$  no número  $x_0$  significa calcular  $P(x_0)$ . Dados o grau  $n$  de um polinômio  $P$ , seus respectivos coeficientes em forma de um vetor  $C$  e um número  $x_0$ , avaliar  $P$  em  $x_0$ .

- O *desvio padrão amostral*  $\sigma$  de um conjunto de números  $X = \{x_1, x_2, x_3, x_4, \dots, x_n\}$  é dado pela expressão,

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

onde  $x_i$  é uma referência ao  $i$ -ésimo elemento de  $X$  e  $\bar{x}$  a média dos elementos de  $X$ . Determine o desvio padrão amostral dos elementos de um vetor de números reais dado como entrada.

- No Brasil um número de CPF (Cadastro de Pessoa Física) possui nove dígitos ( $d_1, d_2, \dots, d_9$ ) e mais dois para verificação de validade do documento ( $d_{10}$  e  $d_{11}$ ). A verificação de  $d_{10}$  segue as etapas,

- Multiplica-se os nove primeiros dígitos pelos pesos 10, 9, 8,  $\dots$ , 2 como na ilustração,

$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$\times$
10	9	8	7	6	5	4	3	2	
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	

- Em seguida determina-se a somatória  $S_1$  dos elementos obtidos na multiplicação anterior,

$$S_1 = \sum_{i=1}^9 a_i$$

- Aplica-se  $S_1$  na seguinte expressão de validação,

$$d_{10} = 11 - S_1 \bmod 11$$

Se o valor de  $d_{10}$  calculado é igual ao fornecido na entrada então a primeira parte da verificação se encerra com sucesso. A verificação de  $d_{11}$  ocorre de forma similar, mas leva em conta  $d_{10}$  conforme etapas,

- Multiplica-se os dez primeiros dígitos pelos pesos 11, 10, 9, 8,  $\dots$ , 2 como na ilustração,

$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$	$\times$
11	10	9	8	7	6	5	4	3	2	
$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	$b_{10}$	

- Em seguida determina-se a somatória  $S_2$  dos elementos obtidos na multiplicação anterior,

$$S_2 = \sum_{i=1}^{10} b_i$$

- Aplica-se  $S_2$  na seguinte expressão de validação,

$$d_{11} = 11 - S_2 \bmod 11$$

Verificado com sucesso  $d_{11}$  o documento fica validado. Dado os dígitos de entrada de um documento CPF determine sua validade.

- Busca Binária* é um processo de busca em vetores ordenados que tira proveito desta ordenação para ser substancialmente mais rápida que a busca linear (problema-2). A busca binária por um elemento  $x$  em um vetor  $M$  ordenado segue as etapas,

- Definir os contadores  $i$  e  $j$  e iniciá-los respectivamente com as posições inicial e final de  $M$ .
- Se  $i$  for menor que  $j$  a busca se encerra sem sucesso.
- Determinar a posição  $k$  média entre as posições  $i$  e  $j$ .
- Se o elemento  $M_k$  for igual a  $x$  a busca termina com sucesso.
- Se  $M_k$  for maior que  $x$  então  $j$  recebe  $k - 1$  e volta-se a etapa-ii.
- Se  $M_k$  for menor que  $x$  então  $i$  recebe  $k + 1$  e volta-se a etapa-ii.

Dados como entrada um vetor  $V$  ordenado de inteiros e um inteiro  $x$ , determinar por busca binária se  $x$  pertence ou não a  $V$ .

15. Dado um vetor de números naturais, reorganizar seus elementos de forma que dois números pares não fiquem vizinhos. Informar quando não for possível.
16. O embaralhamento de vetor, ou *shuffle*, consiste em redispôr seus elementos em ordem aleatória. Dado um vetor de inteiros de entrada, embaralhar seus elementos.
17. Dado um vetor de  $m$  inteiros distintos, selecionar aleatoriamente entre eles  $n$  inteiros ( $n \leq m$ ) também distintos.
18. Uma representação *dígito-vetorial* de um número natural  $n$  é um vetor contendo os dígitos de  $n$  justificados à direita e complementados com zeros à esquerda quando necessário. Por exemplo, a representação dígito-vetorial de 15867 pode ser o vetor,

0	0	1	5	8	6	7
---	---	---	---	---	---	---

Dado um número natural como entrada construir sua representação dígito-vetorial.

19. Para multiplicar um número  $M$  em representação dígito-vetorial (ver problema-18) por um escalar  $x$  utiliza-se a ideia descrita a seguir,
  - a. Constrói-se um vetor de saída  $S$  com espaço suficiente para por a resposta.
  - b. Define-se um acumulador  $ac$  e atribui-se zero a ele.
  - c. Faz-se simultaneamente dois contadores  $i$  e  $j$  variarem respectivamente entre a última e a primeira posição dos vetores  $M$  e  $S$ . Em cada etapa desse processo deve-se,
    - i. Calcular  $r = M_i \cdot x + ac$ .
    - ii. Colocar o último dígito de  $r$  em  $S_j$ .
    - iii. Colocar  $r$  sem o último dígito em  $ac$ .
  - d. Se  $i$  atingir a posição inicial de  $M$  antes de  $ac$  zerar então os dígitos de  $ac$  deverão ser distribuídos pelas células de  $S$  antes do último valor de  $j$  encontrado.

Dados um número em representação dígito-vetorial  $M$  e um escalar  $x$  determinar a multiplicação entre eles.

20. O fatorial de números naturais acima de 50 não pode em geral ser representado por um tipo inteiro primitivo. Uma saída é a utilização da representação dígito-vetorial (problema-18). A ideia para determinar o fatorial de  $n$  é a seguinte,
  - (a) Determinar o comprimento  $m$  de  $n!$ . Isso pode ser computado pela expressão,

$$m = 1 + \lfloor \log_{10} n! \rfloor$$

$$= 1 + \left\lfloor \sum_{k=1}^n \log_{10} k \right\rfloor$$

onde a primeira equação não é computável diretamente, mas a segunda sim.

- (b) Construir um vetor  $S$  de comprimento  $m$  e utilizá-lo para armazenar a representação dígito-vetorial no número 1 (todos os dígitos iguais a zero exceto o último que recebe 1).
- (c) Utilizando a multiplicação de um escalar por um número em representação dígito-vetorial (problema-19), multiplicar  $S$  pelos números  $2, 3, 4, \dots, n$ .

Determinar a soma dos dígitos de  $100!$ .

24. Construa uma função que receba um caractere ASCII e retorne seu equivalente maiúsculo quando estiver na faixa 'a'...'z' e do contrário retorne o próprio caractere de entrada.
25. Utilizando a função do problema-24, converter em caixa alta (caracteres alfabéticos em maiúsculo) uma string dada como entrada.
26. Uma string é um *palíndromo* se a sequência dos caracteres da esquerda para direita é igual à sequência da direita para a esquerda. Por exemplo, são palíndromos "ARARA", "RADAR", "AKASAKA" e "ANA". Determinar se uma string de entrada é ou não um palíndromo.
27. Construir funções que recebam um inteiro positivo como argumento e imprimam respectivamente suas representações binária e hexadecimal.
28. Dados como entrada uma string  $S$  e um caractere  $ch$ , ambos ASCII, eliminar todas as aparições de  $ch$  em  $S$ . Por exemplo, se  $S = \text{"a_casa"}$  e  $ch = \text{'a'}$ , então  $S$  deverá se tornar  $\text{"_cs"}$ .
29. A *comparação lexicográfica* de duas strings é feita caractere a caractere e as comparações ocorrem da esquerda para a direita entre caracteres de mesma posição nas strings comparadas. Se as strings possuem mesmo comprimento e todas as comparações obtêm sucesso então as duas strings são consideradas lexicograficamente idênticas. Se uma é prefixo da outra então a mais curta é lexicograficamente menor (por ex., "casa" < "casado"). Se durante a comparação surge um caractere distinto, será lexicograficamente maior a string que possuir o caractere de maior código ASCII (por ex., "caso" > "casa" pois 'o' > 'a'). Considerando que strings sempre terminam num caractere nulo, comparar lexicograficamente duas strings dadas como entrada e reimprimí-las em ordem crescente.
30. Dada uma string de entrada, imprimir a lista de frequência de seus caracteres. Por exemplo, se a string for "xabbaaaabacxxxxaa" então a saída deverá ser,

Caractere	Frequência
a	5
x	5
c	2
b	3

31. Um número é um *palíndromo* quando a ordem natural e reversa de dígitos revela o mesmo número. Por exemplo, 121, 234432. O número 585 é palíndromo nas bases 10 e base 2 (1001001001). Determinar a soma de todos os números com esta propriedade que sejam menores que um milhão.
32. Dadas duas strings de entrada  $S$  e  $X$ , determinar quantas vezes  $X$  ocorre dentro de  $S$ . Por exemplo, se  $S = \text{"zzabaxdcabasabaqwertabar"}$  e  $X = \text{"aba"}$  então a saída deverá ser 4.

## UTILIZANDO STRINGS

21. Considerando que strings sempre terminam num caractere nulo, determinar o comprimento de uma string dada como entrada.
22. Dada uma string como entrada determinar a soma do código ASCII de seus caracteres.
23. Dada uma string ASCII como entrada, substituir todos os caracteres que não sejam letras pelo caractere "\_".