



Universidade Federal do Ceará  
Departamento de Computação  
Curso de Ciência da Computação

<b>Disciplina</b>	Programação - CK0226	<b>Semestre</b>	2018/2
<b>Professor</b>	Lincoln Souza Rocha		
Laboratório de Programação 03 – Vetores, Alocação Dinâmica e Cadeia de Caracteres			

- 1) Dado dois vetores,  $a$  (5 elementos) e  $b$  (8 elementos), faça um programa em C que imprima todos os elementos comuns aos dois vetores.
- 2) Faça um programa que determina o máximo e o mínimo de um conjunto de  $n$  números inteiros armazenados num vetor  $a$  de 10 elementos. Faça duas funções, uma que retorna o máximo e outra que retorne o mínimo.
- 3) Durante uma corrida automobilística com  $N$  voltas de duração, foram anotados para um piloto, na ordem, os tempos registrados em cada volta. Fazer um programa em C para ler os tempos das  $N$  voltas, guardar em um vetor, implementar funções para calcular e imprimir: (i) melhor tempo; (ii) a volta em que o melhor tempo ocorreu; (iii) pior tempo; (iv) a volta em que o pior tempo ocorreu; e (v) o tempo médio das  $N$  voltas.
- 4) Crie um programa em C que possua dois vetores de inteiros ( $a[]$  e  $b[]$ ) cada um com a capacidade de 10 elementos. O programa deve permitir que seja lido via teclado os dez elementos de cada vetor. Além disso, o programa possui uma função que calcula o produto escalar dos dois vetores de acordo com a formula:

$$escalar = \sum_{i=1}^{10} a[i] * b[i]$$

- 5) Dadas duas sequências com  $n$  números inteiros entre 0 e 9, interpretadas como dois números inteiros de  $n$  algarismos, calcular a sequência de números que representa a soma dos dois inteiros. Veja o exemplo abaixo para um  $n = 8$ :

Sequência 1	8	2	4	3	4	2	5	1
Sequência 2	(+)	3	3	7	5	2	3	7
Sequência 3		1	1	6	1	8	5	8

- 6) Crie um vetor  $v$  com  $n$  inteiros, onde  $n$  é um valor inteiro fornecido pelo usuário. O vetor só deve ser alocado na memória depois que o usuário fornecer o valor de  $n$  (utilize a função `malloc()` pra isso). Depois disso, leia do teclado  $n$  inteiros e armazene em  $v$ . Depois disso, pergunte se o usuário deseja inserir mais elementos, caso a resposta dele seja sim, leia a quantidade de elementos que ele deseja adicionar e aumente o tamanho do vetor sem, no entanto, perder os valores já lidos (utilize a função `realloc()` pra isso). Esse processo deve se repetir até que o usuário informe que não deseja mais adicionar elementos ou a memória se esgote (faça um código que trate essa ocorrência). Ao final, imprima todos os elementos fornecidos pelo usuário e libere a memória (utilize a função `free()` pra isso).

- 7) Crie um programa em C que lê uma sequência de  $n$  inteiros, onde  $n$  é um valor inteiro fornecido pelo usuário. Depois disso, implemente uma função de busca (`int buscar(int numero, int* sequencia)`) que recebe um número fornecido pelo usuário e verifica se esse número encontra-se a sequência. Caso o número se encontre na sequência, a função deve retornar seu índice, caso contrário deve retornar -1. Com base no retorno da função de busca, informe se o número buscado encontra-se na lista e, se afirmativo, em qual posição.
- 8) Considere o programa desenvolvido na questão anterior e crie um função que ordene os elementos da sequência de forma crescente (`void ordena(int* sequencia)`). Para auxiliar seu trabalho, crie também uma função que faz a troca do conteúdo de dois elementos da sequência (`void troca(int* x, int* y)`).
- 9) Dada uma matriz  $A_{m \times n}$  de inteiros, escreva uma função que diga quantas vezes um dado número aparece na matriz (`int repeticoes(int numero)`).
- 10) Escreva um programa que, dados uma matriz  $A_{m \times n}$  de inteiros e um escalar  $\alpha$  inteiro, calcule o produto de  $A \times \alpha$ .
- 11) Escreva um programa que, dados uma matriz  $A_{m \times n}$  e um vetor  $V_{n \times 1}$  de inteiros, calcule o produto de  $A \times V$ . Lembre-se que  $A_{m \times n} \times V_{n \times 1} = R_{m \times 1}$ , ou seja:
- $$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \times \begin{bmatrix} v_{1,1} \\ \vdots \\ v_{n,1} \end{bmatrix} = \begin{bmatrix} r_{1,1} \\ \vdots \\ r_{m,1} \end{bmatrix}, \text{ onde } r_{i,j} = \sum_{k=1}^n a_{i,k} \times v_{k,j}$$
- 12) O Jogo da Velha é um jogo típico de tabuleiro, onde dois jogadores intercalam jogadas. Um jogador usa a marca X e outro jogador usa a marca O para indicar a sua jogada em um tabuleiro de  $A_{3 \times 3}$ . Vence o jogo o jogador que conseguir preencher uma fileira completa, seja na vertical, na horizontal ou na transversal. Escreva um programa que implemente a dinâmica do jogo utilizando uma matriz. O programa deve permitir que os dois jogadores intercalem suas jogadas e avise quando houve um ganhador.
- 13) Faça um programa para corrigir uma prova com 10 questões de múltipla escolha (a, b, c, d ou e), em uma turma com 3 alunos. Cada questão vale 1 ponto. Leia o gabarito, e para cada aluno leia sua matrícula (número inteiro) e suas respostas. Calcule e escreva:
- Para cada aluno, escreva sua matrícula, suas respostas, e sua nota.
  - O percentual de aprovação, assumindo média 7.0.
- 14) Escreva um programa que implemente as seguintes funções:
- `void troca(char *string)`: que transforma todos os caracteres de uma *string* em minúsculos em maiúsculos e os maiúsculos em minúsculos.
  - `int ocorrencia(char *string, char caractere)`: que recebe uma *string* e um caractere, e retorne o número de vezes que esse caractere aparece na *string*.
  - `void remove(char *string, char caractere)`: que recebe uma *string* e um caractere, e apague todas as ocorrências desses caractere na *string*.
  - `void titulo(char *string)`: que altera a primeira letra de cada uma das palavras da *string* deixando-a em maiúsculo.
  - `void inverte(char *string)`: que receba uma *string* e a retorne escrita de trás pra frente.

- 15) Escreva um programa que leia uma *string* do teclado e diga se ela é um palíndromo. Uma *string* é um palíndromo quando pode ser lida tanto de trás pra frente quanto de frente para trás e possui exatamente a mesma sequência de caracteres. Exemplos: ASA e SUBI NO ONIBUS. Desconsidere os espaços em branco. Defina uma função chamada `int palindromo(char *string)` que receba uma *string* como parâmetro e retorne 1 se ela for um palíndromo ou 0, caso contrário. Dica: use a função do item (e) da questão anterior .
- 16) Escreva um programa que implemente as seguintes funções:
- a) `int ocorrencia(char *frase, char *palavra)`: que recebe duas *strings*, e retorne o 1 se a segunda *string* (palavra) ocorre na primeira *string* (frase) ou 0, caso contrário. Se uma *string* A é parte de uma *string* B, diz-se que A é *substring* de B.
  - b) `int nocorrencias(char *frase, char *palavra)`: que recebe duas *strings*, e retorne o número de vezes que a segunda *string* (palavra) ocorre na primeira *string* (frase).
  - c) `int pocorrencia(char *frase, char *palavra)`: que recebe duas *strings*, e retorne a posição da primeira ocorrência da segunda *string* (palavra) na primeira *string* (frase). Exemplo para frase = "Eu gosto do curso de Ciência de Computação e de programar!", palavra = "de", a resposta deve ser: 18.
  - d) `void remove(char *frase, char *palavra)`: que remove todas as ocorrências da *string* palavra da *string* frase. Exemplo: Se frase = "Abra a porta para entrar para descansar calmamente", palavra = "para", a resposta deve ser: "Abra a porta entrar e descansar calmamente".
  - e) `void insercao(char *frase, char *palavra, int pos)`: que recebe duas *strings* e um inteiro, e insere a segunda *string* (palavra) na primeira *string* (frase) a partir da posição pos. Exemplo para frase = "Alo Mundo", palavra = " para o" e pos = 3. a resposta deve ser: "Alo para o Mundo".
  - f) `void substitui(char *frase, char *palavra, char *nova)`: que faça substituição de todas as ocorrências da *string* palavra pela *string* nova dentro da *string* frase. Exemplo: Se frase = "Abra a porta para entrar para descansar calmamente", palavra = "para" e nova = "e tente", a resposta deve ser: "Abra a porta e tente entrar e tente descansar calmamente".