

# Lista de Exercícios

## MC102 - Algoritmos e Programação de Computadores

Instituto de Computação – Unicamp

Primeiro Semestre de 2014

Observações:

- Sugere-se que os exercícios sejam implementados, compilados e testados, usando a linguagem em C, uma vez que os laboratórios deverão ser realizados nesta mesma linguagem.
- Em caso de dúvida quanto a resolução dos exercícios, procure os monitores nos horários de atendimento.

### 1 Variáveis, Atribuições e Expressões Aritméticas

1. Escreva um programa que calcule a média aritmética de 3 valores inteiros.
2. Escreva um programa que troque o valor de 2 variáveis,  $a$  e  $b$ , de modo que, no fim da execução,  $b$  possua o valor de  $a$  e vice-versa.
3. Escreva um programa que calcule as raízes de uma equação de segundo grau qualquer <sup>1</sup>. Por simplicidade, considere que todas as equações possuem exatamente duas raízes reais.
4. Escreva um programa para converter temperaturas em graus Celsius para Fahrenheit e Kelvin, através das equações:

$$F = \frac{9}{5}C + 32$$

$$K = C + 273$$

5. Escreva um programa que calcule as soluções de um sistema de equações da forma

$$ax + by = c$$

$$dx + ey = f$$

cuja solução é obtida através das seguintes equações:

$$x = \frac{ce - bf}{ae - bd}$$

$$y = \frac{af - cd}{ae - bd}$$

Por simplicidade, considere que todos os sistemas de equações desta forma possuem uma solução válida.

---

<sup>1</sup>Utilize a função `double sqrt(double x)` definida na biblioteca `math.h`. Essa função recebe um valor e retorna sua raiz quadrada. Para compilar o programa usando o `gcc`, utilize a opção `-lm`.

- Escreva um programa que, dado o raio, calcule a circunferência e a área do círculo correspondente.
- Escreva um programa que troque o valor de 2 variáveis numéricas,  $a$  e  $b$  como no exercício 2, mas sem utilizar variável auxiliar.
- Escreva um programa que, dado o intervalo  $[a, b]$  com  $0 \leq a \leq b \leq 100$ , calcule a soma de todos os números ímpares contidos nesse intervalo. Exemplo: se o intervalo for  $[2, 9]$  a soma será  $3 + 5 + 7 + 9 = 24$ .
- Escreva um programa que determine as coordenadas  $(x, y)$  de um vértice de um paralelogramo, dado os outros três vértices na seguinte ordem: primeiro as coordenadas do vértice diagonalmente oposto ao vértice a ser determinado, seguido pelas coordenadas dos outros dois vértices.
- Escreva um programa que determine o  $n$ -ésimo termo da sequência

3, 5, 8, 12, 17, 23, ...

## 2 Comandos Condicionais

- Escreva um programa que tenha como entrada 3 valores reais, que representam os comprimentos de cada lado de um triângulo. Verifique se estes segmentos formam um triângulo e, caso positivo, que tipo é, ou seja, isósceles, equilátero, escaleno e/ou retângulo.
- Aprimore o item 4 da Seção 1 para que o usuário possa escolher qual conversão de escalas ele pretende fazer.
- Escreva um programa que receba como entrada um valor inteiro e verifique se ele possui 4 dígitos. Se não possuir, a execução é terminada e uma mensagem de erro deve ser exibida. Caso possua, deve-se verificar se o número atende a seguinte propriedade: a soma dos dois primeiros dígitos é igual a soma dos dois últimos dígitos.
- Escreva um programa que tenha como entrada o peso, em kg, e a altura, em m, de uma pessoa e que calcule o seu IMC a partir da fórmula:

$$IMC = \frac{M}{H^2}$$

Por fim, determinar em que faixa essa pessoa se encaixa:

- menor que 18.5 : abaixo do peso
  - entre 18.5 e 24.9 : peso normal
  - entre 25.0 e 29.9 : sobrepeso
  - entre 30.0 e 34.9 : obesidade grau I
  - entre 35.0 e 39.9 : obesidade grau II
  - maior que 40.0 : obesidade grau III
- Escreva um programa que receba como entrada três valores reais e imprima os valores do maior e do menor, assim como a ordem em que foram lidos.

6. Escreva um programa que leia os coeficientes  $a_1$ ,  $b_1$ ,  $a_2$  e  $b_2$  de duas equações de reta ( $y_1 = a_1x + b_1$  e  $y_2 = a_2x + b_2$ ) e verifique se as duas retas se interceptam e, caso positivo, imprima em qual ponto isso acontece.
7. Escreva um programa que calcule as raízes de uma equação de segundo grau qualquer. Seu programa deve lidar com o caso da equação possuir uma única raiz, assim como o caso de possuir raízes imaginárias.
8. Escreva um programa que leia um ano (valor inteiro) e imprima se ele é bissexto ou não.
9. Escreva um programa que leia um número entre 1 e 7 e imprima o dia da semana correspondente a esse número, ou seja, domingo se o valor é igual a 1, segunda-feira se 2, assim por diante.
10. Escreva um programa para verificar se um número inteiro é divisível por 3 ou por 5, mas não simultaneamente por ambos.
11. Escreva um programa que dados quatro números distintos determine e imprima a soma dos três menores.
12. Escrever um programa que lê um par de coordenadas inteiras  $(x, y)$  e imprima uma mensagem informando em qual quadrante está o ponto. O algoritmo deve também ser capaz de identificar se o ponto está sobre um dos eixos ou na origem do plano cartesiano.
13. Escreva um programa que receba a idade e o nome de um nadador e imprima o seu nome, a sua idade e a categoria do mesmo, de acordo com as regras a seguir:
  - Infantil 1: menor ou igual a 10 anos
  - Infantil 2: 11 ou 12 anos
  - Juvenil: de 13 a 16 anos
  - Junior: de 17 a 19 anos
  - Senior: maior que 19 anos
  - Master: maior que 25 anos
14. Escreva um programa que dado um número de 5 dígitos imprima se este é ou não palíndromo. Números palíndromos são aqueles que escritos da direita para esquerda ou da esquerda para direita tem o mesmo valor. Exemplos de palíndromos: 929, 44 e 97379.

### 3 Comandos de Repetição

1. Escreva um programa que solicite ao usuário uma quantidade de números a serem lidos e que informe quantos desses valores estão nos intervalos  $(-\infty, 0]$ ,  $(0, 25]$ ,  $(25, 50]$ ,  $(50, 75]$ ,  $(75, 100]$  ou  $(100, +\infty)$ .
2. Escreva um programa que calcule o fatorial de um número inteiro através de multiplicações sucessivas.
3. Escreva um único programa que solicite ao usuário que digite um número inteiro  $N$  e calcule:
  - a soma dos quadrados dos  $N$  valores que serão solicitados e digitados em seguida;

- o quadrado da soma dos  $N$  valores solicitados.
4. Faça um programa que calcule e exiba a soma dos divisores de um número inteiro maior que zero.
  5. Escreva um programa que calcule e exiba a soma dos quadrados dos 100 primeiros números inteiros.
  6. Escreva um programa que converta valores decimais para outra base numérica de 2 a 9, utilizando divisões sucessivas.
  7. O valor da função  $\cos(x)$ , para qualquer valor de  $x$  em radianos, pode ser estimada através da expressão <sup>2</sup>:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Dado um número real  $x$  e um inteiro  $k$ , estime o valor de  $\cos(x)$ , utilizando  $k$  termos desta sequência.

8. Escreva um programa que leia um número inteiro  $n$  fornecido pelo usuário e gere um *quadrado* de  $n$  linhas e  $n$  colunas que tenha caracteres  $x$  nas posições da diagonal principal e caracteres  $+$  nas demais.

Entrada: 4

Saída:

x+++

+x++

++x+

+++x

9. Um número inteiro  $n$  é dito subnúmero de um outro número  $m$ , se os dígitos de  $n$  aparecem na mesma sequência em  $m$ . Escreva um programa que leia dois números,  $n$  e  $m$ , e imprima SIM se  $n$  é um subnúmero de  $m$  e NAO caso contrário.

Exemplos:

Entrada: 21 3212

Saída: SIM

Entrada: 21 3231

Saída: NAO

10. Um número inteiro  $n$  é dito alternante se a sequência de dígitos que o forma alterna entre dígitos pares e ímpares. Exemplos de números alternantes são: 5, 12, 21, 12345 e 252. Escreva um programa que leia um número inteiro e imprima se o número é alternante ou não.
11. Supondo que a população de um país  $A$  seja de  $P_A$  de habitantes com uma taxa anual de crescimento de  $T_A$  e que a população de um país  $B$  seja de  $P_B$  habitantes com uma taxa anual de crescimento de  $T_B$  ( $P_A < P_B$  e  $T_A > T_B$ ), escreva um programa que calcule e imprima o número de anos necessários para que a população do país  $A$  ultrapasse ou iguale a população do país  $B$ , mantidas essas taxas de crescimento.
12. Escreva um programa que leia números inteiros até que se digite um número negativo. O programa deve retornar o menor e o maior número lido.

---

<sup>2</sup>Esta expressão é obtida através da série de Taylor calculada em torno de 0.

13. Escreva um programa que leia um número inteiro positivo  $n$  e imprima  $n$  linhas do chamado triângulo de Floyd.

Exemplo:  $n = 6$

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

14. O número harmônico, denotado  $H_n$ , é definido como  $n$ -ésimo termo da série harmônica, ou seja

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Escreva um programa que calcule o valor de  $H_n$  para um determinado  $n$  positivo.

15. Implemente a potenciação sem usar a função `pow` da biblioteca `math.h`. O usuário deve fornecer com dois números,  $x$  (real) e  $y$  (inteiro), e o programa deve imprimir na tela o valor de  $x^y$ .
16. Faça um programa que leia um número natural  $n$  e calcule a soma de todos os números de 1 até  $n$ . Não use fórmulas, tal como a soma de uma progressão aritmética.
17. Escreva um programa que calcule a associação em paralelo  $R_{eq}$  de dois resistores  $R_1$  e  $R_2$  cujas resistências devem ser fornecidas pelo usuário. O programa deve então a pedir um novo valor de resistência  $R_n$  e calcular a associação em paralelo de  $R_n$  com a  $R_{eq}$  previamente calculada. Este processo deve se repetir até que uma resistência nula seja fornecida, quando o programa deve imprimir a resistência final (ignorando a resistência de valor nulo).

## 4 Vetores

1. Implemente um programa que leia uma lista de até 50 números inteiros e troque o primeiro elemento da lista com o último, o segundo com o penúltimo e assim por diante. Imprima a lista antes e depois da operação.
2. Escreva um programa que conte quantos números pares e ímpares um vetor de inteiros possui.
3. Considere que, neste exercício, um vetor de inteiros represente um número, ou seja, o índice 0 representa as unidades, o 1 representa as dezenas e assim por diante. Implemente um programa que receba dois vetores de inteiros e então calcule (e imprima) um terceiro vetor de inteiros que represente a soma dos dois primeiros.
4. Dadas duas sequências de  $n$  e  $m$  valores inteiros, onde  $n \leq m$ , escrever um programa que verifique quantas vezes a primeira sequência ocorre na segunda.

Exemplo:

```
primeira sequência: 1 0 1
segunda sequência: 1 1 0 1 0 1 0 0 1 1 0 1 0
Resultado: 3
```

5. Escreva um programa para intercalar os valores de dois vetores inteiros crescentes de mesmo tamanho em um terceiro vetor, em ordem crescente.

Exemplo:

```
v1 = [1, 3, 5, 5, 7, 9, 10]
v2 = [2, 2, 4, 6, 8, 8, 10]
v3 = [1, 2, 2, 3, 4, 5, 5, 6, 7, 8, 8, 9, 10, 10]
```

6. Escreva um programa que leia o tamanho e os elementos de um vetor de inteiros positivos. O programa deve imprimir a posição e o valor do menor e do maior elemento do vetor.
7. Escreva um programa que leia um conjunto de notas, calcule a média e gere um vetor com as notas iguais ou acima da média e outro com as notas abaixo da média e imprima-os.
8. Escreva um programa que leia um vetor de inteiros e imprima a soma e o produto dos elementos desse vetor.
9. Escreva um programa que leia um vetor de inteiros e troque os elementos de índice par com os de índice ímpar imediatamente seguintes. Ao final, imprima o novo vetor.
10. Escreva um programa que leia um vetor de inteiros e conte quantos elementos pares e ímpares ele possui, calcule também a somatória dos elementos pares e ímpares e imprima os dados, dizendo qual é o maior em cada caso.

## 5 Cadeias de Caracteres

1. Escreva um programa que receba duas cadeias de caracteres e verifique se elas são idênticas. Caso não sejam, imprima o índice a partir do qual diferem.
2. Faça um programa que leia 2 cadeias de caracteres, A e B, de tamanhos máximos de 255 e 25 caracteres, respectivamente, e gere uma cadeia de caracteres C, que é dada pela união das cadeias A e B. Deve-se solicitar ao usuário que forneça a posição de A a partir da qual B será inserida. Em outras palavras, primeiro copia-se A em C até esse índice e, em seguida, copia-se toda a cadeia B. Por fim, se houver necessidade, copia-se o resto de A.
3. Faça um programa que leia duas cadeias de caracteres e elimine, da segunda cadeia, todas as ocorrências dos caracteres da primeira cadeia.
4. Um palíndromo é uma palavra ou frase que pode ser lida igualmente da esquerda para a direita ou da direita para a esquerda, desconsiderando-se os espaços em brancos. Exemplos de palíndromos: “radar”, “reviver”, “mirim”, “a sacada da casa” e “a mala nada na lama”. Escreva um programa que leia uma cadeia de caracteres de até 80 caracteres e teste se ela é um palíndromo.
5. Faça um programa que leia uma cadeia de caracteres e armazene o seu inverso em uma outra cadeia de caracteres. Exemplo: se a cadeia lida for “gustavo”, a cadeia resultante deve ser “ovatsug”.

6. Escreva um programa que leia uma cadeia de caracteres e imprima quantas consoantes foram lidas e, em seguida, a lista das consoantes lidas.
7. Escreva um programa que leia uma cadeia de caracteres. Considere que essa cadeia de caracteres é uma mensagem. Você deve aplicar sobre essa mensagem o seguinte método de codificação: cada caractere é trocado com seu vizinho. Esse processo é feito aos pares, caso a mensagem tenha um número ímpar de caracteres, o programa adiciona um '#' ao final da mensagem.  
Exemplo:  
**mensagem original:** Esta é uma mensagem.  
**mensagem codificada:** sEaté u amm neaseg.m
8. Escreva um programa que leia uma cadeia de caracteres e também outros dois caracteres  $a$  e  $b$ . O programa deve fazer duas coisas:
  - imprimir a cadeia de caracteres omitindo toda ocorrência de  $a$
  - imprimir a cadeia de caracteres e a cada ocorrência de  $a$  trocar por  $b$
9. Escreva um programa que leia uma cadeia de caracteres e reescreva-a colocando todas as suas letras em maiúsculo.
10. Em alguns lugares é comum lembrar um número do telefone associando seus dígitos a letras. Dessa maneira a expressão MY LOVE significa 69 5683. Claro que existem alguns problemas, uma vez que alguns números de telefone não formam uma palavra ou uma frase e os dígitos 1 e 0 não estão associados a nenhuma letra. Escreva um programa que leia uma expressão e encontre o número de telefone correspondente baseado na tabela abaixo. Uma expressão é composta por letras maiúsculas (A-Z), hifens (-) e os números 1 e 0.

ABC	2
DEF	3
GHI	4
JKL	5
MNO	6
PQRS	7
TUV	8
WXYZ	9

## 6 Matrizes

1. Implemente as operações de soma, subtração, multiplicação e transposição de matrizes quadradas  $n \times n$ .
2. Suponha que a matriz seguinte representa ligações entre cidades, e que, se uma posição  $(i,j)$  possui o valor 1, então há uma ligação da cidade  $i$  para a  $j$ . Exemplo:

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Neste caso, há caminhos disponíveis da cidade 0 para a 1 e 2 e da 2 para 0. Tendo essas informações, faça um programa que seja capaz de:

- Listar as cidades com entrada e sem saída;
  - Listar as cidades com saída e sem entrada;
  - Listar cidades isoladas;
  - Dizer, saindo de cada cidade, quais as que podem ser alcançadas diretamente.
3. Faça um programa que verifique se uma matriz quadrada pode ser considerada uma *quadrado mágico*. Um *quadrado mágico* é uma matriz na qual o valor da soma dos elementos de cada linha, de cada coluna e de cada diagonal é o mesmo.

Por exemplo, a matriz

$$\begin{bmatrix} 15 & 8 & 1 & 24 & 17 \\ 16 & 14 & 7 & 5 & 23 \\ 22 & 20 & 13 & 6 & 4 \\ 3 & 21 & 19 & 12 & 10 \\ 9 & 2 & 25 & 18 & 11 \end{bmatrix}$$

é quadrado mágico, cujas somas valem 65.

4. Considere o seguinte esquema de criptografia de mensagens, que é uma variação do método chamado de Cifra de Colunas. Toda mensagem é criptografada numa sequência de caracteres de tamanho equivalente a um quadrado perfeito. Assim, por exemplo, para uma mensagem de 16 caracteres usa-se um quadrado de quatro linhas e quatro colunas. Para mensagens de 25 caracteres, usa-se um quadrado de cinco por cinco. Para mensagens de 100 caracteres, é necessário um quadrado de dez por dez, e assim por diante. Para transcrever o mensagem deve-se preencher as casas do quadrado, linha por linha, e ler a mensagem coluna por coluna. Escreva um programa para ler uma cadeia de caracteres, decodificá-la (usando o método descrito a cima) e imprimir a mensagem decifrada.

Exemplo:

MEEUMOC SHMSC1T\*AGUOA\*\*\*L2\*\*\*\*T\*\*\*\*\*A

Esta mensagem deve ser transcrita em um quadrado de  $6 \times 6$ :

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

Lendo cada coluna da matriz (desconsiderando o caractere '\*'), a saída deverá ser:

MC102 ESTA EH UMA MSG OCULTA

5. Sudoku é jogado numa malha de  $9 \times 9$  quadrados, dividida em submalhas de  $3 \times 3$  quadrados, chamadas “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.

- Escreva um programa que leia uma matriz de números reais e gere uma matriz onde a primeira linha representa a soma dos elementos de cada coluna da matriz de entrada, a segunda linha representa a média dos elementos de cada coluna da matriz de entrada.
- Escreva um programa que leia a quantidade de variáveis de um sistema linear e a matriz associada a este sistema e o resolva por escalonamento. Considere apenas sistemas possíveis e determinados.
- Um conjunto de empresas está desenvolvendo um programa para controle do trabalho realizado por seus gerentes nos diversos departamentos das mesmas. Considerando que o conjunto de empresas é formado por  $n$  empresas e que cada uma delas possui  $m$  departamentos com um gerente para cada um, foi gerada uma matriz onde cada posição contém o código numérico de um gerente, sendo que cada linha corresponde a uma empresa e cada coluna a um departamento (considerando que as empresas estão codificadas com números de 1 a  $n$  e os departamentos com números de 1 a  $m$ ). Escreva um programa que solicite o número de empresas  $n$ , o número de departamentos  $m$ , a matriz e o código de um gerente a ser pesquisado na mesma. O programa deve retornar o código da empresa e código do departamento a que o gerente a ser pesquisado pertence. Considere que não existem dois gerentes com códigos iguais.
- Leia uma matriz de tamanho  $n \times m$  que se refere a  $m$  respostas de questões de múltiplas escolhas, referentes a  $n$  alunos. Leia também um vetor de  $m$  posições contendo o gabarito de respostas que podem ser A, B, C ou D. Seu programa deverá comparar as respostas de cada aluno com o gabarito e gerar um vetor contendo o número de acertos de cada aluno.
- Escreva um programa para ajudar a administrar uma enfermaria. Seu programa deve ler a quantidade de pacientes. E depois ler as medidas da pulsação de cada paciente. As medidas são feitas de hora em hora e as entradas referem-se às medidas de um dia, ou seja, 24 aferições para cada paciente. Seu programa deve informar:
  - média da pulsação de cada paciente
  - o valor e o horário da maior pulsação de cada paciente

## 7 Funções e Ponteiros

1. Considere o seguinte código em C:

```
#include <stdio.h>

int funcao_qualquer(int inteiro) {
    inteiro = inteiro + 2;
    return inteiro;
}

int main() {
    int variavel_a = 5;
    int variavel_b = funcao_qualquer(variavel_a);
    printf("%d %d\n", variavel_a, variavel_b);
    return 0;
}
```

Quais serão os valores impresso por este programa?

2. Considere o seguinte código em C:

```
#include <stdio.h>

int funcao_qualquer_modificada(int* inteiro) {
    *inteiro = *inteiro + 2;
    return *inteiro;
}

int main() {
    int variavel_a = 5;
    int variavel_b = funcao_qualquer_modificada(&variavel_a);
    printf("%d %d\n", variavel_a, variavel_b);
    return 0;
}
```

Quais serão os valores impresso por este programa?

3. Explique por que utiliza-se o símbolo & na função *scanf* e não na *printf*.
4. Considere o seguinte código:

```
void funcao_qualquer() {
    variavel_principal = variavel_principal/2;
    ...
    return;
}

int main() {
    int variavel_principal = 5;
    funcao_qualquer();
    return 0;
}
```

A compilação deste código indicará erros? Explique o motivo.

5. Mostre o valor especificado em cada item após a execução das instruções abaixo, supondo que o endereço da variável `x` é 1000 e da variável `y` é 1004.

```
int x,y;
int* p1;
int* p2;
x = 10;
y = 20;
p1 = &x;
p2 = &y;
(*p1)++;
```

- a) `x`
- b) `y`
- c) `&x`
- d) `&y`
- e) `p1`
- f) `p2`
- g) `*p1 + *p2`
- h) `*(&x)`
- i) `&(*p2)`

6. Mostre a saída do programa a seguir:

```
#include <stdio.h>

int main() {
    int i1, i2, *p1, *p2;

    i1 = 5;
    p1 = &i1;
    i2 = *p1 / 2 + 10;
    p2 = p1;

    printf("i1 = %d, i2 = %d, *p1 = %d, *p2 = %d\n", i1, i2, *p1, *p2);

    return 0;
}
```

7. Indique alguns efeitos e benefícios que podem ser causados pela separação entre a implementação das funções (arquivos `.c`) e as declarações (arquivos `.h`).
8. Considere o seguinte código. Há um erro na função `misterio`. Qual é? Dê uma solução para o erro.

```

#include <stdio.h>

void misterio (int* p, int* q) {
    int* temp;
    *temp = *p;
    *p = *q;
    *q = *temp;
}

int main() {
    int i = 6, j= 10;
    misterio(&i, &j);
    printf("%d %d\n", i, j);
    return 0;
}

```

9. Suponha que  $v$  é um vetor. Descreva a diferença conceitual entre as expressões  $v[3]$  e  $v + 3$ .
10. Suponha que os elementos do vetor  $v$  são do tipo `int` e cada `int` ocupa 8 bytes no seu computador. Se o endereço de  $v[0]$  é 55000, qual o valor da expressão  $v + 3$ ?

## 8 Algoritmos de Ordenação

1. Baseado no método *Bubble Sort*, considere um método de classificação denominado *Shake Sort*. Na primeira fase ele se comporta como o *Bubble Sort*, ou seja, passa por todos os elementos do vetor da esquerda para direita (considerando que à esquerda deverão ficar os valores menores) comparando os elementos dois a dois e quando o elemento da esquerda é maior que o da direita eles trocam de lugar. Na segunda fase, ao invés de ir da esquerda para direita, o *Shake Sort* vai da direita para esquerda, desta vez, se o da direita for menor que o da esquerda eles trocam de posição. As duas fase são feitas de forma alternada até o vetor ficar ordenado. Este método utiliza dois índices: um que começa na direita e vai decrescendo e outro que começa na esquerda e vai crescendo. Escreva uma função C que receba um vetor contendo números e um número inteiro com o tamanho do vetor e realize o *Shake Sort*.
2. Escreva um programa que leia a quantidade de alunos  $n$  e depois  $n$  linhas contendo o RA e o nome de cada aluno. O programa deve então imprimir em ordem crescente de RA a listagem dos alunos.
3. Escreva um programa que leia a quantidade de alunos  $n$  e depois  $n$  linhas contendo o RA e o nome de cada aluno. O programa deve então imprimir as informações dos alunos (nomes e RAs), em ordem alfabética dos nomes. Para comparar se um nome precede outro na ordem da alfabética utilize a função `strcmp` da biblioteca `string.h`.
4. Escreva um programa que leia quantos pacientes deram entrada na recepção de um hospital. Cada linha posterior recebe o nome do paciente, a prioridade no atendimento (quanto maior o número, menor a prioridade) e a ordem de chegada. Imprima a lista de atendimento considerando a prioridade e, em caso de empate, a ordem de chegada.

- Escreva um programa que leia a quantidade  $n$  de tipos de produto em um armazém e depois leia a porcentagem do item em estoque. Considerando que se a porcentagem em estoque for menor que 50% o armazém tem que repor o item, imprima, em ordem decrescente de porcentagem em estoque, os itens que tem que ser repostos.

## 9 Busca Sequencial e Binária

- Explique como funciona a busca binária e por que ela é mais eficiente do que a sequencial.
- Em que situação a busca sequencial passa a ser mais eficiente que a binária?

## 10 Registros e Arquivos

- Faça um programa que leia um arquivo de registros de funcionários (cada um deles contendo nome, salário e matrícula) e efetue as seguintes operações:
  - Calcular o salário total;
  - Indicar o nome, salário e matrícula dos funcionários com o menor e maior salário;
  - Ler um número de matrícula, procurá-lo no arquivo e excluir esse registro, deslocando todos os registros seguintes para a posição anterior;
  - Ler os dados de um novo funcionário (nome, salário e matrícula), procurá-lo no arquivo e, se a matrícula não existir, incluí-lo ao final do mesmo. Se existir, substituir o registro encontrado por este novo.
- Utilize a mesma estrutura de registros definida no exercício anterior e implemente um programa que leia dois arquivos, ordenados pelas matrículas dos funcionários, e realize a intercalação destes, de forma que o resultado seja um terceiro arquivo, também ordenado pelas matrículas dos funcionários, que contenha todos os registros.
- Uma empresa deseja um relatório de preços atualizados dos produtos no seu estoque. Um arquivo em disco contém os dados sobre o código de cada produto, o valor atual e o ano de fabricação. Os valores dos produtos devem ser atualizados de acordo com a tabela abaixo.

Para cada registro lido, escreva o código do produto, o ano de fabricação, o valor atual e o valor atualizado, nessa ordem. O reajuste deve ser realizado de acordo com a seguinte tabela:

Ano de Fabricação	Fator de Correção (em %)
Menor que 1986	517.00
Maior ou igual que 1986 e menor que 1988	226.43
Maior ou igual que 1988 e menor que 1990	15.93
Maior ou igual que 1990	8.50

- Escreva um programa que conte o número de linhas, palavras e caracteres em um arquivo texto (semelhante ao comando `wc` do Linux).
- Faça um programa que receba uma palavra e o nome de um arquivo texto. Seu programa então deve abrir o arquivo e procurar pela palavra fornecida. Caso seja encontrada, o programa deverá exibir na tela o número da linha do arquivo onde a palavra foi encontrada, bem com a linha completa. Se houver mais de uma linha contendo a palavra, todas as linhas devem ser exibidas, segundo o formato definido acima.

Dica: use a função `strstr`, da biblioteca `string.h`, para procurar a palavra em cada linha do arquivo.

- Escreva um programa que compare dois arquivos cujos nomes devem ser fornecidos pelo usuário. O programa deve imprimir uma mensagem se eles forem idênticos e, se forem diferentes, imprimir a linha e a coluna onde ocorreu uma disparidade pela primeira vez.
- Considere um arquivo que contenha registros de alunos composto do nome, seguido de três reais indicando as notas da primeira, segunda e terceira provas respectivamente. Escreva um programa que calcule a média  $M$  dos alunos de acordo com o critério:

$$M = \frac{3P1 + 3P2 + 4P3}{10} \quad (1)$$

... e gere um segundo arquivo indicando, para cada aluno, o seu nome, a sua média e se ele está **APROVADO** ou **REPROVADO**, considerando que a nota para aprovação é 5.

- Escreva um programa que receba um arquivo de código fonte em C e faça uma cópia desse arquivo omitindo os comentários. Considere que os comentários começam com `/*` e terminam com `*/`, e que não há, em uma mesma linha, código e comentário. Considere que as linhas não excedem 100 caracteres.
- Escreva um programa que receba um arquivo que representa o extrato de uma conta bancária. A primeira linha contém um inteiro indicando o saldo anterior. As linhas em seguida contêm um real e o nome da operação, que pode ser **DEBITO** ou **CREDITO**. Escreva um programa que leia o extrato e imprima na tela o valor do saldo após as operações realizadas na conta.

## 11 Recursão

- Implemente duas funções, uma recursiva e outra não, que encontrem o maior elemento contido em um vetor de inteiros de tamanho  $n$ .
- Escreva um programa que solicite ao usuário um número inteiro e que calcule o fatorial desse valor através da recursão.
- Escreva um programa que tenha como entrada um número inteiro e que calcule o valor do respectivo termo na sequência de Fibonacci. A sequência de Fibonacci é obtida através de:

$$F(n) = \begin{cases} 0 & \text{se } x = 0, \\ 1 & \text{se } x = 1, \\ F(n-1) + F(n-2) & \text{se } x > 1. \end{cases}$$

- Implemente uma função recursiva que, dados dois números inteiros  $x$  e  $n$ , calcule o valor de  $x^n$ .
- Implemente uma função recursiva que imprima a representação binária de um inteiro decimal passado como parâmetro.
- Implemente uma função recursiva que compute o MDC de dois inteiros positivos passados como parâmetro. Calcule o MDC através das seguintes relações:

$$MDC(x, y) = \begin{cases} x & \text{se } x = y, \\ MDC(y, x) & \text{se } x < y, \\ MDC(x - y, x) & \text{se } x > y. \end{cases}$$

Exemplo:

$$MDC(10,6) = MDC(4,6) = MDC(6,4) = MDC(2,4) = MDC(4,2) = MDC(2,2) = 2$$

7. Pode-se calcular o resto da divisão, MOD, de x por y, de dois números inteiros positivos, usando-se a seguinte definição:

$$MOD(x, y) = \begin{cases} 0 & \text{se } x = y, \\ x & \text{se } x < y, \\ MOD(x - y, y) & \text{se } x > y. \end{cases}$$

Implemente uma função recursiva que receba dois números inteiros positivos como parâmetros e retorne o resto da divisão inteira entre eles.

8. Pode-se calcular o quociente da divisão, DIV, de dois números inteiros positivos, usando-se a seguinte definição:

$$DIV(x, y) = \begin{cases} 1 & \text{se } x = y, \\ 0 & \text{se } x < y, \\ 1 + DIV(x - y, y) & \text{se } x > y. \end{cases}$$

Implemente uma função recursiva que receba dois números inteiros positivos como parâmetros e retorne o quociente da divisão inteira entre eles.

9. Escreva um algoritmo recursivo capaz de gerar todas as permutações de uma string formada por letras que não se repetem.
10. Escreva uma função recursiva que identifica se uma palavra é um palíndromo, ou seja, se ela pode ser lida da mesma forma da esquerda para direita, quanto da direita para esquerda.
11. Escreva uma função recursiva que conte as ocorrências de um número inteiro, passado como parâmetro, em um vetor de inteiros.
12. John McCarthy é um teórico famoso de ciência da computação. No seu trabalho, ele definiu uma função recursiva, chamada f91, que recebe como entrada um inteiro N e retorna um inteiro positivo definido como a seguir:

$$f91(n) = \begin{cases} n - 10 & \text{se } n > 100 \\ f91(f91(n + 11)) & \text{se } n \leq 100. \end{cases}$$

Escreva um programa que computa a função f91 de McCarthy.

13. A sequência de Fibonacci pode ser estendida para ordens maiores, assim temos a sequência de Tetranacci definida como:

$$T(n) = \begin{cases} 0 & \text{se } x = 0, 1, 2 \\ 1 & \text{se } x = 3, \\ T(n-1) + T(n-2) + T(n-3) + T(n-4) & \text{se } x > 1. \end{cases}$$

Implemente uma função que leia um número inteiro e calcule o respectivo termo da sequência de Tetranacci.

14. Um truque bem conhecido para descobrir se um inteiro  $N$  é um múltiplo de nove é computar a soma  $S$  dos seus dígitos. Se  $S$  é um múltiplo de nove, então  $N$  também é. Este é um teste recursivo, já que se  $N > 9$  então o procedimento pode ser repetido. A profundidade da recursão, ou seja quantas chamadas recursivas são necessária para obter a resposta para um número  $N$  é chamada o grau-9 de  $N$ . Sua tarefa é, dado um inteiro positivo  $N$ , determinar se ele é um múltiplo de nove e, caso ele seja, qual o seu grau-9.
15. A Torre de Hanoi é um “quebra-cabeça” que consiste em uma base contendo três pinos, em um dos quais são dispostos alguns discos uns sobre os outros, em ordem crescente de diâmetro, de tal forma que o maior esteja na parte de baixo do pino. O problema consiste em passar todos os discos (um a um) de um pino para outro qualquer, usando o terceiro pino como auxiliar, de maneira que um disco maior nunca fique em cima de outro menor.

Considere que os pinos são nomeados, da esquerda para direita, como  $A$ ,  $B$ ,  $C$  e que os discos são numerados de 1 a  $n$ , sendo 1 o disco de menor diâmetro.

Escreva um programa recursivo que leia a quantidade de discos  $n$  e transfira todos os discos de  $A$  para  $C$ , usando  $B$  como auxiliar, descrevendo todos os movimentos.

16. Implemente o algoritmo *Insertion Sort* de forma recursiva.
17. Implemente o algoritmo *Quick Sort*.