

**MC102 - Algoritmos e Programação de Computadores**  
Segundo semestre de 2009

Prof. Leonel Carlos Pereira

**Obs.:** A lista não irá compor a nota, essa lista é apenas para o acompanhamento do aluno da disciplina. A lista que irá compor a nota será liberada em algumas semanas.

## Aulas 01-04

1. Escreva um programa que efetue o cálculo do salário liquido de um professor. Os dados fornecidos são: o valor da hora aula, numero de aulas dadas no mês e o percentual de desconto do INSS.
2. Escreva um programa que leia dois números  $X$  e  $Y$  e efetue a troca dos valores de forma que a variável  $X$  passe a ter o valor da variável  $Y$  e que a variável  $Y$  passe a ter o valor da variável  $X$ . Apresente os valores trocados. (Sugestão: utilize uma variável auxiliar).
3. Escreva um algoritmo que leia três números inteiros e positivos ( $A, B, C$ ) e calcule a seguinte expressão:  $D = \frac{R+S}{2}$ , onde  $R = (A + B)^2$  e  $S = (B + C)^2$ .
4. Faça um algoritmo que leia o tempo de duração de um evento em uma fábrica expressa em segundos e mostre-o expresso em horas, minutos e segundos.

## Aulas 05-06

1. Calcule a média aritmética das 3 notas de um aluno e mostre, além do valor da média, uma mensagem de “Aprovado”, caso a média seja igual ou superior a 5, ou a mensagem “reprovado”, caso contrário.

2. Determine os resultados obtidos na avaliação das expressões lógicas seguintes, sabendo que:

$$a = 3;$$

$$b = 2.5;$$

$$c = 5;$$

$$d = 1;$$

e  $L$  é uma constante lógica cujo valor é falso ( $F$ ) e  $M$  é uma constante lógica cujo valor é verdadeiro ( $V$ ):

- (a)  $a == b * c \&\& (L || M)$
- (b)  $b > a || d == (c \% a)$

- (c)  $(!M) \parallel L \&\& (a + b) > d$
- (d)  $(a + b) \geq (c / d) \parallel M \&\& L$
- (e)  $(d - (b * c) / a) \parallel L \&\& (!M)$

Lembre-se: A ordem de prioridade dos operadores é: parênteses mais internos, operadores aritméticos, operadores relacionais, operadores lógicos. Entre os operadores lógicos a ordem de prioridade é o `!`, em seguida o `&&` e por último o `||`.

3. Elaborar um programa que lê 2 valores  $a$  e  $b$  e os escreve com a mensagem: “São múltiplos” ou “Não são múltiplos”.
4. Faça um programa que leia um número inteiro e mostre uma mensagem indicando se este número é par ou ímpar, e se é positivo ou negativo.
5. Tendo como dados de entrada a altura e o sexo de uma pessoa (‘M’ masculino e ‘F’ feminino), construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:
  - para homens:  $(72.7 * h) - 58$
  - para mulheres:  $(62.1 * h) - 44.7$
6. Elaborar um programa que lê 3 valores  $a$ ,  $b$  e  $c$  e verifica se eles formam ou não um triângulo. Supor que os valores lidos são inteiros e positivos. Caso os valores formem um triângulo, calcular e escrever a área deste triângulo. Se não formam triângulo escrever os valores lidos. (Se  $a > b + c$  não formam triângulo algum, se  $a$  é o maior).
7. Escrever um programa que lê a percentagem do IPI a ser acrescido no valor das peças, o código da peça 1, valor unitário da peça 1, quantidade de peças 1 e o código da peça 2, valor unitário da peça 2, quantidade de peças 2. O programa deve calcular o valor total a ser pago e apresentar o resultado. Fórmula :  $(valor1 * quant1 + valor2 * quant2) * (IPI/100 + 1)$
8. Escrever um algoritmo e um programa que lê um valor em reais e calcula qual o menor número possível de notas de 100, 50, 10, 5 e 1 em que o valor lido pode ser decomposto. Escrever o valor lido e a relação de notas necessárias.
9. Fornecidos os lados de um triângulo ( $L1$ ,  $L2$ ,  $L3$ ). Escrever um programa C que escreva se os lados formam um triângulo e se for verificar se é eqüilátero, isósceles ou escaleno. Traduzir o algoritmo abaixo para C.

```

inicio
  real lado1, lado2, lado3;
  escreva("Digite o lado 1º lado do triângulo: ");
  leia (lado1);
  escreva("Digite o lado 2º lado do triângulo: ");

```

```
leia (lado2);
escreva("Digite o lado 3º lado do triângulo: ");
leia (lado3);
se (lado1 + lado2 <= lado3 OU lado1 + lado3 <= lado2 OU
    lado2 + lado3 <= lado1) então
    escreva( "Estes lados não formam um triângulo!!!!");
senão se (lado1 == lado2 E lado2 == lado3 ) então
    escreva ( "Os lados ", lado1, ", ", lado2, ", ", lado3,
              " formam um triângulo retângulo");
senão se (lado1 == lado2 OU lado1 == lado3 OU
          lado3 == lado2) então
    escreva ( "Os lados ", lado1, ", ", lado2, ", ", lado3,
              " formam um triângulo isósceles");
senão
    escreva ( "Os lados ", lado1, ", ", lado2, ", ", lado3,
              " formam um triângulo escaleno");
fim
```

10. O Restaurante do Terraço Espanha tem uma opção de almoço executivo com 3 etapas: Entrada, Prato Principal e Sobremesa. Cada etapa tem alguma opção com preços diferentes.
- Entrada
- A. Sopa de escargots canadenses - R\$8.30
  - B. Yakissoba de jerimum - R\$7.25
  - C. Sushi com Torresmo - R\$4.30
- Prato Principal
- A. Língua de colibri recheada - R\$10.25
  - B. Orelha de ornitorrinco flambada - R\$11.40
  - C. Peito de perdiz asiático - R\$10.00
  - D. Pernas de rã com maracujá - R\$10.00
- Sobremesa A. Sorvete de brócolis - R\$4.00 B. Torta de doce de jiló e queijo - R\$5.95
- Caso o total da conta seja maior que R\$20.00 o cliente tem 5% de desconto. Elabore um programa C que aceite as entradas do usuário relativas as escolhas feitas e imprima o valor da conta a pagar.
11. Escreva um programa que pede para o usuário entrar um número correspondente a um dia da semana e que então apresente na tela o nome do dia. utilizando o comando *switch*.

12. Escreva um programa que converta uma temperatura expressa em graus Celsius em seu valor equivalente em graus Fahrenheit ( $C = 5/9 * (F - 32)$ ) ou vice-versa de acordo com a opção de conversão escolhida pelo usuário. As opções são ‘C’ para conversão de Fahrenheit para Celsius e ‘F’ para conversão de Celsius para Fahrenheit.
13. Faça um programa que, obtidos seis números inteiros, representando duas quantidades de tempo na forma: horas, minutos e segundos; calcule e mostre a soma destes números. Esta soma também deve estar na forma de horas, minutos e segundos.

## Aulas 07-08

1. Faça um programa que apresente na tela a tabela de conversão de graus Celsius para Fahrenheit, de  $-100C$  a  $100C$ . Use um incremento de  $10C$ . Obs.:  $Farenheit = (9/5)*(Celsius) + 32$
2. Desenvolva um programa que calcule a soma dos  $n$  primeiros números inteiros pares positivos.
3. Escreva um programa que apresente os  $n$  primeiros números primos a partir do número 2 para um valor  $n > 0$  fornecido pelo usuário.
4. Dado um número positivo  $n$ , achar todos os seus divisores inteiros diferentes de  $n$ .

Exemplo:

Se a entrada do programa for 308, então a saída deve ser: 1 2 4 7 11 22 28 44 77 154.

5. Escreva um programa que gere, para um valor  $n \geq 0$  fornecido pelo usuário, um “quadrado” de  $n$  linhas e  $n$  ‘:’ nas posições da diagonal principal e os caracteres ‘+’ nas demais posições. Por exemplo, para  $n = 5$  o programa deve gerar colunas que tenha caracteres:

```
:++++
+:+++
++:++
+++:+
++++:
```

6. Escreva um programa que descubra um número imaginado pelo usuário entre 0 e  $n > 0$ . Para cada valor sugerido pelo programa como sendo o valor imaginado pelo usuário, o usuário deve responder (honestamente) se o valor sugerido pelo programa é igual, menor ou maior do que o valor imaginado com os caracteres ‘=’, ‘<’ e ‘>’, respectivamente. A execução do programa deve terminar assim que o programa “adivinhar” o valor imaginado pelo usuário.

7. Dado um número inteiro positivo, crie um programa C que mostre este número com seus dígitos invertidos.

Exemplo:

5 - 5

32 - 23

201 - 102

2000 - 2

8. Elaborar um programa que leia do teclado uma altura  $h$  em número de linhas e desenhe na tela do computador um “X” como o exemplo, utilizando a função printf(). Exemplos:

para  $h = 4$ :

```
X  X
XX
XX
X  X
```

para  $h = 7$

```
X      X
      X  X
      X  X
      X
      X  X
      X  X
X      X
```

9. Escrever um programa que leia caracteres até que seja digitado um ‘f’ ou um ‘F’ e que ao final mostre a quantidade total de caracteres maiúsculos e minúsculos digitados, incluindo o último.
10. Um valor inteiro positivo  $n$  é chamado de quadrado perfeito se existir uma sequência de ímpares consecutivos a partir do valor 1 cuja soma seja exatamente igual a  $n$ . Exemplo: para o valor 16 temos  $16 = 1 + 3 + 5 + 7$ . Assim sendo 16 é um quadrado perfeito. Um quadrado perfeito tem a seguinte propriedade: o número de termos ímpares consecutivos  $m$  a partir do valor 1 cuja soma é igual ao quadrado perfeito corresponde à raiz quadrada do quadrado perfeito. No exemplo acima, para  $n = 16$ , o valor de  $m$  é 4, o que corresponde à raiz quadrada de 16. Escreva um programa que verifique se um valor inteiro é um quadrado perfeito e, em caso afirmativo, determine o valor de sua raiz quadrada de acordo com o procedimento descrito acima.

## Aulas 09-10

1. Reescreva o programa a seguir realizando, a cada leitura, um teste para ver se a dimensão do vetor não foi ultrapassada. Caso o usuário entre com 100 números, o programa deverá abortar o *loop* de leitura automaticamente. O uso do Flag não deve ser retirado.

```
#include <stdio.h>
int main ()
{
    int num[100]; /* Declara um vetor de inteiros de 100 posicoes */
    int count=0;
    int totalnums;
    do
    {
        printf ("\nEntre com um numero (-999 p/ terminar): ");
        scanf ("%d",&num[count]);
        count++;
    } while (num[count-1]!=-999);
    totalnums=count-1;
    printf ("\n\n\n\tOs números que você digitou foram:\n\n");
    for (count=0;count<totalnums;count++)
        printf (" %d",num[count]);
    return 0;
}
```

2. Escreva um algoritmo e depois um programa em linguagem C que leia um conjunto de pares de números, um inteiro e um real, que representam número de matrícula e a nota do aluno. Calcule e imprima:
  - (a) a média da turma;
  - (b) quantos alunos obtiveram nota maior do que a média;
  - (c) quantos alunos obtiveram nota menor do que a média;
  - (d) quantos alunos obtiveram nota igual à média;
  - (e) qual foi a maior nota e quantos e quais alunos a obtiveram.

3. Escreva um programa que recebe como entrada uma linha de caracteres, com tamanho máximo 30, e a partir de tal linha, lendo um caractere por vez, produza as subseqüências contíguas de caracteres não brancos, uma por linha. Exemplo:

Entrada:

as12; bbcdd ---987 .:.. @Alf0#

Saída:

```
as12;
bbcdd
---987
. . . .
@Alf@#
```

4. Em uma competição de ginástica olímpica a nota é determinada por um painel de seis juízes. Cada um dos juízes atribui uma nota entre zero e dez para o desempenho do atleta. Para calcular a nota final, a nota mais alta e a nota mais baixa são descartadas e é calculado a média das quatro restantes. Escreva um programa que leia 6 notas entre zero e dez e calcule a média após o descarte da maior e da menor nota.
5. Escrever um programa que, para uma dada sequência de  $3 \leq n \leq 100$  valores inteiros fornecidos em uma linha de entrada pelo usuário de comprimento maior ou igual a 3, calcule o número de triplas (subsequências de comprimento 3) com valores iguais existentes na sequência dada. Exemplo: Para a sequência

4 2 15 15 15 3 7 7 7 2

o programa deve produzir o resultado 3, pois a sequência dada contém uma tripla com valores 15 e duas (sobrepostas) com valores 7.

6. Escreva um programa que obtenha duas matrizes  $2 \times 3$ , some-os e mostre o resultado da soma na tela.
7. Suponha dois vetores (no sentido matemático)  $u$  e  $v$ . Eles podem ser representados, computacionalmente, como dois vetores. Você pode utilizar o seguinte código para referenciar com mais naturalidade os elementos do vetor de acordo com sua componente  $(x, y, z)$

```
#define x 0 // para poder referenciar com mais naturalidade
#define y 1 // os elementos do vetor de acordo com sua
#define z 2 // componente (x,y,z)
```

```
int u[3], v[3];
```

Assim,  $u[x]$  representa a componente  $x$  do vetor,  $u[y]$  representa a componente  $y$  e  $u[z]$  representa a componente  $z$ .

Escreva um algoritmo e depois um programa em linguagem C que calcule e imprima:

- (a) o produto escalar dos dois vetores;

- (b) o produto vetorial dos dois vetores.
8. O que imprime o programa a seguir? Tente entendê-lo e responder. A seguir, execute-o e comprove o resultado.

```
# include <stdio.h>

int main()
{
    int t, i, M[3][4];
    for (t = 0; t < 3; ++t)
        for (i = 0; i < 4; ++i)
            M[t][i] = (t*4) + i + 1;
    for (t = 0; t < 3; ++t)
    {
        for (i = 0; i < 4; ++i)
            printf ("%3d ", M[t][i]);
        printf ("\n");
    }
    return 0;
}
```

9. Faça um programa que inverta uma string, tamanho máximo 30: leia a string e armazene-a invertida em outra string. Use o comando for para varrer a string até o seu final.

## Aulas 11-13

1. Escreva um programa fazendo o uso de *struct's*. Você deverá criar uma *struct* chamada *Ponto*, contendo apenas a posição *x* e *y* (inteiros) do ponto. Declare 2 pontos, leia a posição (coordenadas *x* e *y*) de cada um e calcule a distância entre eles. Apresente no final a distância entre os dois pontos.
2. Considere que os seguintes dados de funcionários devem ser mantidos por registros de uma empresa: nro, nome, endereço, salário, data de contratação. Faça um programa para criar um cadastro com dados de 500 funcionários. Leia um valor de salário mínimo e imprima duas relações: uma contendo os funcionários que recebem acima de 5 salários e outra com aqueles que recebem abaixo de 2 salários.
3. Escrever uma função que determine a soma dos elementos na diagonal secundária de uma matriz quadrada de  $n > 0$  linhas e colunas. Exemplo: A soma dos elementos da diagonal secundária da matriz

$$\begin{pmatrix} 2 & 5 & 10 & 4 \\ -3 & 8 & 1 & 5 \\ 4 & 0 & 7 & 11 \\ 3 & -4 & 1 & 2 \end{pmatrix}$$

é  $4 + 1 + 0 + 3 = 8$

4. Execute o seguinte programa e diga o que será impresso:

```
#include <stdio.h>

void Z(int L);
void P(int R, int S, int *T);

int main()
{
    int A, B, C;
    A = 8;
    B = 5;
    C = 2;
    printf("%d, %d, %d", A, B, C);
    P(A, B, &C);
    printf("%d, %d, %d", A, B, C);
    A = 1;
    B = 6;
    C = 0;
    printf("%d, %d, %d", A, B, C);
    P(A, B, &C);
    printf("%d, %d, %d", A, B, C);
    A = 4;
    B = 4;
    C = 1;
    printf("%d, %d, %d", A, B, C);
    P(A, B, &C);
    printf("%d, %d, %d", A, B, C);
    return 0;
}

void Z(int L)
{
    int I;
```

```

I = 10;
L = (L / 2) * I;
printf("%d\n", L);
}

void P(int R, int S, int *T)
{
    if(R > S)
    {
        *T = R * S + 3;
        Z(*T);
    }
    else
    {
        R = R + 2;
        S = S + R;
    }
    printf("%d, %d, %d\n", R, S, *T);
}

```

5. Faça um programa que leia N matrizes quadradas de tamanho até 3x3 e escreva o valor do determinante. Coloque a leitura e o cálculo do determinante como uma função. Não se esqueça de escrever os resultados.
  6. Escreva a função *EDivisivel*(int a, int b). A função deverá retornar 1 se o resto da divisão de a por b for zero. Caso contrário, a função deverá retornar zero.
  7. Escreva uma função que recebe três números inteiros, correspondentes a dia, mês e ano. Teste os números recebidos, em caso de haver algum inválido, retorne 0 e em caso verdadeiro (dias entre 1 e 31, mês entre 1 e 12 e ano entre 1900 e 2100), retorne 1. Verifique se o mês e o número de dias batem (incluindo verificação de anos bissextos). Escreva também um programa que utilize dessa função e que seja chamada enquanto a data passada por parâmetro for inválida. Comente seu programa.
- Obs:** Um ano é bissexto se for divisível por 4 e não for divisível por 100, exceto para os anos divisíveis por 400, que também são bissextos.
8. Escrever uma função (e um programa que utilize tal função) que determine o índice da coluna de uma matriz de inteiros (composta por  $n > 0$  linhas e  $m > 0$  colunas) com o maior valor de soma de elementos por coluna. Exemplo: Para a matriz

$$\begin{pmatrix} 2 & 5 & 4 & 10 \\ -3 & 8 & 5 & 1 \\ 4 & 0 & 11 & 7 \end{pmatrix}$$

a função deve retornar o valor 2, pois a soma dos elementos da terceira coluna (20) é maior que os valores da soma dos elementos de cada uma das demais colunas (3, 13 e 18 nas colunas 0, 1 e 3, respectivamente).

9. Crie uma função que funcione igual à função *scanf* na leitura de string, isto é, que obtenha uma string a partir do teclado. Crie também um programa para testar a função. Dica: Leia a string letra a letra até que o caractere ENTER seja digitado. Não esqueça de colocar o '\0' ao final.
10. Escrever uma função que copie apenas as iniciais das palavras contidas em uma string para outra. Fazer também um programa principal para poder testá-la. A assinatura da função é *void iniciais(char \*nome, char \*ini)*. Por exemplo, para o nome “José Sebastião Costa”, a função retorna “JSC”.
11. Faça um programa que multiplique um vetor de ordem *n* por uma matriz de ordem *nxm*. Deve ser feito uma função para ler vetor, uma para imprimir vetor, uma para ler matriz, uma para imprimir matriz e uma outra que irá fazer a multiplicação. Usar variáveis locais e as passagens de parâmetros (por valor e por referência) de forma apropriada. As chamadas às funções devem ser feitas a partir do programa principal.
12. Faça um programa de conversão de base numérica. O programa deverá apresentar uma tela de entrada com as seguintes opções:

Conversao de base:

- 1: decimal para hexadecimal
- 2: hexadecimal para decimal
- 3: decimal para octal
- 4: octal para decimal
- 5: Encerra

Informe sua opção:

A partir da opção escolhida, o programa deverá pedir o número na base escolhida, lê-lo e apresentá-lo na base desejada. Em seguida, o programa deve perguntar ao usuário se ele deseja retornar ao menu principal ou finalizar o programa. O problema pode ser estendido a outras bases, de acordo com o interesse do aluno.