

# Introdução à Programação

## AULA 5 – Estruturas de Condição

Prof<sup>a</sup>. Glaucia M. M. Campos

[glaucia.melissa@gmail.com](mailto:glaucia.melissa@gmail.com)

# Processamento Condicional

- Algoritmos com fluxo de execução único
  - As instruções são sempre as mesmas, independente dos valores introduzidos
- Procedimentos podem ser executados dependendo de uma série de situações/condições
  - Muitas vezes, é necessário executar um conjunto de instruções se uma condição for verdadeira e, caso contrário, um outro conjunto de instruções.
  - Quando um programa executa ou deixa de executar instruções com base no valor de uma condição, o programa realiza um **processamento condicional**.

# Situações que geram condições

- Exemplos (vida real):

...

**Digite sua senha no caixa eletrônico**

Se senha digitada for correta então

Entrar no sistema

...

# Situações que geram condições

- Exemplos (vida real):

...

Escolha um produto que quer comprar

Se tem dinheiro para comprá-lo então

Coloque no carrinho

Senão

Devolva para a prateleira

Escolha uma marca mais barata

...

# Situações que geram condições

- Exemplos (vida real):

...

**Escolha uma família para entrega de cestas básicas gratuitas**

Se a família tiver mais do que 4 componentes então

Não entregar cesta básica gratuita

Explicar o motivo da não entrega

Senão

Entregar cesta básica gratuita

...

# Processamento Condicional

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int a = 2, b = 3, c = 1;
```

```
    float delta, x1, x2;
```

```
    delta = b*b - 4*a*c;
```

```
    printf("A equacao %s\n", (delta >= 0) ? "possui raizes reais" :  
        "nao possui raizes reais");
```

```
    if (delta >= 0)
```

```
    {
```

```
        printf("As raizes sao %s\n", (delta > 0) ? "diferentes" : "iguais");
```

```
        x1 = (-b + sqrt(delta)) / (2*a);
```

```
        x2 = (-b - sqrt(delta)) / (2*a);
```

```
        printf("Raiz x1 = %f\n", x1);
```

```
        printf("Raiz x2 = %f\n", x2);
```

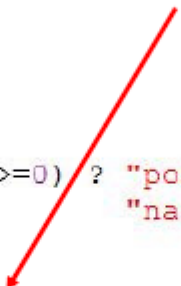
```
    }
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

Estas instruções serão executadas  
somente se  $\text{delta} \geq 0$ .



# Processamento Condicional

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

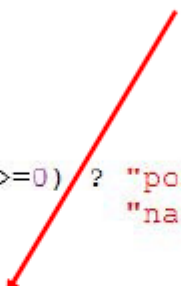
int main(int argc, char *argv[])
{
    int a = 2, b = 3, c = 1;
    float delta, x1, x2;

    delta = b*b - 4*a*c;
    printf("A equacao %s\n", (delta >= 0) ? "possui raizes reais" :
        "nao possui raizes reais");

    if (delta >= 0)
    {
        printf("As raizes sao %s\n", (delta > 0) ? "diferentes" : "iguais");
        x1 = (-b + sqrt(delta)) / (2*a);
        x2 = (-b - sqrt(delta)) / (2*a);
        printf("Raiz x1 = %f\n", x1);
        printf("Raiz x2 = %f\n", x2);
    }

    system("PAUSE");
    return 0;
}
```

Estas instruções serão executadas somente se  $\text{delta} \geq 0$ .



# Processamento Condicional

- Para executar um processamento condicional, um programa precisa utilizar o comando if.
- Todo comando if requer uma condição. O valor de uma condição pode ser verdadeiro ou falso.
- Em C, não existe um tipo de dados específico para representar valores lógicos (V ou F).
- Quando existe um conjunto de instruções a ser executado, caso o valor da condição seja falso, utiliza-se o comando if-else

Qualquer valor diferente de zero é interpretado como verdadeiro, enquanto zero é falso.



# Operadores Relacionais

- Para escrever condições, são utilizados os operadores relacionais e os operadores lógicos

Operador	Significado
>	Maior do que.
<	Menor do que.
>=	Maior do que ou igual a.
<=	Menor do que ou igual a.
==	Igual a.
!=	Diferente de.

Condição	Valor lógico
(a != x)	Verdadeiro.
(a/2.0 == x)	Verdadeiro.
(a/2 == x)	Falso.
(a/x < 2)	Falso.
(a)	Verdadeiro.
(a - 2*x)	Falso.

```
int a = 3; float x = 1.5;
```

# Operadores Relacionais

- Para escrever condições, são utilizados os operadores relacionais e os operadores lógicos

Operador	Significado
&&	Conjunção lógica ("and")
	Disjunção lógica ("or")
!	Negação lógica ("not")

Expressão	Valor Lógico
$((a/2 == x) \&\& (a > 2))$	Falso.
$((x \leq a) \&\& (a \geq 2 * x))$	Verdadeiro.
$!(a/3 \leq x)$	Falso.
$(a \&\& x)$	Verdadeiro.
$((a - 2 * x)    (x < a/2))$	Falso.

```
int a = 3; float x = 1.5;
```

# Operador Condicional

- O operador condicional na linguagem C tem a seguinte sintaxe:

```
(condição) ? resultado-se-condição-verdadeira : resultado-se-condição-falsa
```

- Os resultados podem ser de qualquer tipo (int, float, char, double) e mesmo strings.
- Exemplos:

```
(b != 0) ? a/b : 0  
(peso <= 75) ? "ok" : "deve emagrecer"
```

# Operador Condicional

- O operador condicional pode ser usado em atribuições.
- Exemplo:

```
float nota1 = 5.5, nota2 = 4.3;  
  
media = ((nota1 >= 3) && (nota2 >= 5)) ?  
        (nota1 + 2*nota2)/3 :  
        (nota1 + nota2)/2;
```

↓  
**media** recebe o valor 4.9.

# Operador Condicional

- No programa abaixo, o operador condicional é usado dentro da função printf.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char *argv[])
{
    int a = 2, b = 3, c = 1;
    float delta, x1, x2;

    delta = b*b - 4*a*c;
    printf("A equacao %s\n", (delta >= 0) ? "possui raizes reais" :
                                                "nao possui raizes reais");
    if (delta >= 0)
    {
        printf("As raizes sao %s\n", (delta > 0)? "diferentes" : "iguais");
        x1 = (-b + sqrt(delta))/(2*a);
        x2 = (-b - sqrt(delta))/(2*a);
        printf("Raiz x1 = %f\n", x1);
        printf("Raiz x2 = %f\n", x2);
    }

    system("PAUSE");
    return 0;
}
```

# Comando if

- Exemplo:

```
if (delta >=0)
{
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
}
```

- Um conjunto de instruções começa com um símbolo { e termina com o símbolo }. Caso o conjunto tenha apenas uma instrução, as chaves são opcionais

# Comando if-else

- Exemplo:

```
if (delta >=0)
{
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
}
else
{
    printf("Sem raízes reais.");
}
```

- O comando se pode ser acompanhado pela cláusula else, que será executada quando o valor booleano avaliado na condição for falso (F)

# Comando if-else

- Qualquer instrução pode fazer parte de um conjunto de instruções, inclusive um comando if ou mesmo um comando if-else

```
if (delta >=0)
{
    x1 = (-b + sqrt(delta))/(2*a);
    if (delta == 0)
        x2 = x1;
    else
        x2 = (-b - sqrt(delta))/(2*a);
}
else
{
    printf("Sem raízes reais.");
}
```

O comando **if-else** corresponde a uma única instrução.



# A Importância dos Recuos

- Programas mais complexos são mais difíceis de ler e compreender.
- Uma forma de melhorar a legibilidade do programa é usar recuos.
- Os recuos devem ser usados sempre após o símbolo {, sendo as instruções recuadas à direita.
- O símbolo } deve estar alinhado ao abre-chaves correspondente.

# A Importância dos Recuos

- Exemplo:

```
if (nota >= 9)
    if (nota_anterior < nota)
        printf("Você está melhorando.");
else
    printf("Sem estudo é difícil ser aprovado.");
```

- De quem é o else acima?
  - O compilador sempre associa um else ao “if anterior mais próximo que ainda não possui um else.”
- Como associar o else à instrução if (nota >= 9)?

# A Importância dos Recuos

- Exemplo:

```
if (nota >= 9)
{
    if (nota_anterior < nota)
        printf("Você está melhorando.");
}
else
    printf("Sem estudo é difícil ser aprovado.");
```

- Neste caso, as chaves, em vez de opcionais, serão obrigatórias, pois apenas os recuos não resolvem.

# Condições Compostas

- As condições vistas até o momento são únicas. Conforme aumenta a complexidade dos problemas, é necessária a utilização de condições mais elaboradas

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n1, n2, n3;
    printf("Digite os valores de n1, n2 e n3");
    scanf("%d %d %d", &n1, &n2, &n3);
    if((n1==n2) && (n2==n3))
        printf("Os numeros sao iguais \n");
    if((n1!=n2)&&(n2!=n3)&&(n1!=n3))
        printf("Os numeros são diferentes \n");
    if ((n1==n2)&&(n1!=n3)) || ((n1==n3)&&(n1!=n2)) || ((n2==n3)&&(n1!=n2))
        printf("Há dois valores iguais e um diferente \n");
    system("PAUSE");
    return 0;
}
```

# Comandos if-else relacionados

- Exemplo:

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int dias, mes;
    printf("Digite o mês:");
    scanf("%d", &mes);
    if( (mes==1) || (mes==3) || (mes==5) || (mes==7) || (mes==8) || (mes==10) || (mes==12) )
        dias=31;
    else if(mes==2)
        dias=28;
    else if( (mes==4) || (mes==6) || (mes==9) || (mes==11) )
        dias=30;
    else{
        dias=0;
        printf("O numero do mes esta incorreto?");
    }
    printf("O mes %d tem %d dias \n", mes, dias);
    system("PAUSE");
    return 0;
}
```

# Atribuição e Teste de Igualdade

- Atenção!
  - Na linguagem C, o operador de atribuição (=) e o operador relacional (==) são diferentes

```
int fator = 3;

if (fator == 1)
{
    printf("O fator e unitario.\n");
}
printf("fator = %d\n", fator);
```

Imprimi: fator = 3

(fator == 1) é falso!

```
int fator = 3;

if (fator = 1)
{
    printf("O fator e unitario.\n");
}
printf("fator = %d\n", fator);
```

Imprimi: O fator e unitario.  
fator = 1

(fator = 1) é verdadeiro!

# Comando switch

- Sintaxe para comandos if-else relacionados

```
switch (expressão)
{
    case constante-1:
        comandos-1;
    case constante-2:
        comandos-2;
    ...
    default:
        comandos-n;
}
```

# Comando switch

- Este comando permite que, de acordo com o valor de uma expressão, seja executado um ou mais comandos dentre uma série de alternativas.
- O caso cuja constante for igual ao valor da expressão será selecionado para execução.
- Atenção!
  - Os comandos associados a este caso e todos os comandos seguintes serão executados em sequência até o final do comando switch.
  - Para evitar a execução de todos os comandos seguintes, usa-se o comando break.



# Comando switch

- Como assim?

Para mes = 2:

- Com o uso de **break**:

dias = 28;

- Sem o uso de **break**:

dias = 28;

dias = 30;

dias = 0;

printf("O número do mês  
estah incorreto");

```
int main(){
    int dias, mes;
    printf("Digite o mês:");
    scanf("%d", &mes);
    switch (mes) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            dias=31;
            break;
        case 2:
            dias=28;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            dias=30;
            break;
        default:
            dias=0;
            printf("O numero do mes esta incorreto");
    }
    printf("O mes %d tem %d dias \n", mes, dias);
    system("PAUSE");
    return 0;
}
```

# Comando break

- Em alguns programas, durante um processamento iterativo, pode ser necessário:

- Encerrar o processamento iterativo independentemente do valor da condição do laço;
- Executar apenas parcialmente uma iteração, ou seja, executar somente algumas das instruções do laço da repetição.

- Para encerrar um processamento iterativo, independentemente do valor da condição do laço, deve-se usar o comando break.

# Comando break

- Exemplo: dados os valores N (int) e A (float), determine a partir de qual termo o valor de:

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

é maior do que A. Suponha N = 10 e A = 2.

Instante	Valor de s
1º termo	1.000000
2º termo	1.500000
3º termo	1.833333
4º termo	2.083333

← A partir do quarto termo  $s > A$ .

# Problemas (1/11)

```
//Algoritmo para identificar se o numero digitado foi 5
#include <stdio.h>
#include <stdlib.h>

int main() {
    int num;
    printf("Digite numero:");
    scanf("%d", &num);
    if(num==5) {
        printf("O numero e igual a 5");
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

# Problema (2/11)

```
//Algoritmo que determina o maior numero entre dois digitados
#include <stdio.h>
#include <stdlib.h>

int main(){
    int num1, num2;
    printf("Digite numero 1:");
    scanf("%d", &num1);
    printf("Digite numero 2:");
    scanf("%d", &num2);
    if(num1>num2){
        printf("O numero 1 e maior do que o numero 2");
    }else{
        printf("O numero 2 e maior do que o numero 1");
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

# Problemas (3/11)

```
//Algoritmo que identifica se um numero é par e positivo
#include <stdio.h>
#include <stdlib.h>

int main(){
    int num1;
    printf("Digite numero 1:");
    scanf("%d", &num1);
    if((num1%2==0)&&(num1>0)){
        printf("O numero e par e positivo");
    }else if((num1%2==0)&&(num1<0)){
        printf("O numero e par e negativo");
    }else if((num1%2!=0)&&(num1>0)){
        printf("O numero nao e par, mas e positivo");
    }else if((num1%2==0)&&(num1==0)){
        printf("O numero e par e igual a zero");
    }else{
        printf("Numero nao e par e nem eh positivo");
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

# Problemas (4/11)

```
#include <stdio.h>
#include <stdlib.h>
//Calculadora com if e else

int main(){
    float n1, n2, result;
    char op;
    printf("Digite os valores de n1 e n2: ");
    scanf("%f %f", &n1, &n2);
    printf("Digite o operador:");
    op=getche();

    printf("\n");
    if(op=='+'){
        result=n1+n2;
    } else if (op=='-') {
        result=n1-n2;
    } else if (op=='*'){
        result=n1*n2;
    } else if (op=='/'){
        if(n2!=0)
            result=n1/n2;
        else
            printf("Não pode dividir numero por 2 \n");
    } else
        printf("Operador não existente \n");

    printf("Resultado: %f", result);
    printf("\n");
    system ("PAUSE");
    return 0;
}
```

# Problemas (5/11)

```
//Algoritmo que calcula o valor de  $f(x)=8/(2-x)$ 
#include <stdio.h>
#include <stdlib.h>

int main(){
    float x, fx;
    printf("Digite o valor de x:");
    scanf("%f", &x);
    if(x!=2){
        fx=(8/(2-x));
        printf("Valor de fx: %f", fx);
    } else
        printf("Operacao nao pode ser feita");
    printf("\n");
    system("PAUSE");
    return 0;
}
```



# Problemas (6/11)

```
//Algoritmo que imprime o mes correspondente
#include <stdio.h>
#include <stdlib.h>
int main(){
    int num;
    printf("Digite numero de 1 a 12:");
    scanf("%d", &num);
    if(num==1)
        printf("Janeiro");
    else if(num==2)
        printf("Fevereiro");
    else if(num==3)
        printf("Marco");
    else if(num==4)
        printf("Abril");
    else if(num==5)
        printf("Maio");
    else if(num==6)
        printf("Junho");
    else if(num==7)
        printf("Julho");
    else if(num==8)
        printf("Agosto");
    else if(num==9)
        printf("Setembro");
    else if(num==10)
        printf("Outubro");
    else if(num==11)
        printf("Novembro");
    else if(num==12)
        printf("Dezembro");
    else printf("Nao existe mes correspondente!");
    printf("\n");
    system("PAUSE");
    return 0;
}
```

# Problema (7/11)

```
#include <stdio.h>
#include <stdlib.h>
int main(){ //Calculadora com switch
    float n1, n2, result;
    char op;
    printf("\n\n\t\t\t\t\t*****\n");
    printf("\t\t\t\t\t*CALCULADORA*\n");
    printf("\t\t\t\t\t*****\n");
    printf("\n\t\t\t\t\t+ para somar\n");
    printf("\n\t\t\t\t\t- para subtrair\n");
    printf("\n\t\t\t\t\t* para multiplicar\n");
    printf("\n\t\t\t\t\t/ para dividir\n");
    printf("Digite o operador:");
    op=getche();
    printf("\n");
    printf("Digite o valor de n1: ");
    scanf("%f", &n1);
    printf("Digite o valor de n2: ");
    scanf("%f", &n2);
    printf("\n");
    switch(op){
        case '+': result=n1+n2; break;
        case '-': result=n1-n2; break;
        case '*': result=n1*n2; break;
        case '/':
            if(n2!=0){
                result=n1/n2; break;
            } else
                printf("Nao e possivel fazer divisao por zero!");
        default:
            printf("Opcao invalida!");
    }
    printf("Resultado: %f", result);
    printf("\n");
    system ("PAUSE");
    return 0;
}
```

# Problemas (8/11)

```
//Mes com switch
#include <stdio.h>
#include <stdlib.h>

int main(){
    int num;
    printf("Digite numero de 1 a 12:");
    scanf("%d", &num);
    switch(num) {
        case 1: printf("Janeiro");
        case 2: printf("Fevereiro");
        case 3: printf("Marco");
        case 4: printf("Abril");
        case 5: printf("Maio");
        case 6: printf("Junho");
        case 7: printf("Julho");
        case 8: printf("Agosto");
        case 9: printf("Setembro");
        case 10: printf("Outubro");
        case 11: printf("Novembro");
        case 12: printf("Dezembro");
        default:
            printf("Numero nao existe mes corresponde!");
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

# Problema (9/11)

```
#include <stdio.h>
#include <stdlib.h>
int main(){ //Cardápio de um restaurante
    char op;
    printf("\n\n\t\t\t\t\t*****\n");
    printf("\t\t\t\t\t+CARDAPIO+\n");
    printf("\t\t\t\t\t*****\n");
    printf("\n\t\t\t\t\t 1.PEIXES\n");
    printf("\n\t\t\t\t\t 2.SALADAS\n");
    printf("\n\t\t\t\t\t 3.FRANGO\n");
    printf("\n\t\t\t\t\t 4.CARNES\n");
    printf("\n\n\n");
    printf("Digite sua opcao:");
    op=getche();
    printf("\n");
    if(op=='1'){
        printf("\n\t\t\t\t\tPORCOES SERVEM 4 PESSOAS \n");
        printf("\t\t\t\t\tPeixe grelhado, acompanha arroz e batata doce - R$ 45,00 \n");
        printf("\t\t\t\t\tPeixe cozido, acompanha arroz e pirão - R$ 43,00 \n");
        printf("\t\t\t\t\tPeixe frito, acompanha tapioca e fritas - R$ 38,00 \n");
    } else if(op=='2'){
        printf("\n\t\t\t\t\tPORCOES INDIVIDUAIS \n");
        printf("\t\t\t\t\tAlface, rúcula, alcaparras, molho de mostarda - R$ 15,00 \n");
        printf("\t\t\t\t\tManjericao, tomate seco, rucula, alface, molho agrião - R$ 18,00 \n");
        printf("\t\t\t\t\tAlface, tomate seco, manjerona, alcaparras, molho de maracujá - R$ 22,00 \n");
    } else if(op=='3'){
        printf("\n\t\t\t\t\tPORCOES SERVEM 4 PESSOAS \n");
        printf("\t\t\t\t\tFrango grelhado, acompanha arroz e batata doce - R$ 45,00 \n");
        printf("\t\t\t\t\tFrango cozido, acompanha arroz e pirão - R$ 43,00 \n");
        printf("\t\t\t\t\tFrango frito, acompanha tapioca e fritas - R$ 38,00 \n");
    } else if(op=='4'){
        printf("\n\t\t\t\t\tPORCOES SERVEM 4 PESSOAS \n");
        printf("\t\t\t\t\tCarne bovina grelhada, acompanha arroz e batata doce - R$ 45,00 \n");
        printf("\t\t\t\t\tCarne bovina cozida, acompanha arroz e pirão - R$ 43,00 \n");
        printf("\t\t\t\t\tCarne de porco frita, acompanha tapioca e fritas - R$ 38,00 \n");
    }
```

# Problema (9/11)

- Continuação...

```
}else  
    printf("Nao temos esta opcao no cardapio. Escolha outra!");  
  
printf("\n");  
system ("PAUSE");  
return 0;  
}
```

# Problema (10/11)

```
#include <stdio.h>
#include <stdlib.h>

int main(){ //Cardápio de um restaurante
    char op;
    printf("\n\n\t\t\t\t\t*****\n");
    printf("\t\t\t\t\t*CARDAPIO*\n");
    printf("\t\t\t\t\t*****\n");
    printf("\n\t\t\t\t\t1.PEIXES\n");
    printf("\n\t\t\t\t\t2.SALADAS\n");
    printf("\n\t\t\t\t\t3.FRANGO\n");
    printf("\n\t\t\t\t\t4.CARNES\n");
    printf("\n\n\n");
    printf("Digite sua opcao:");
    op=getche();
    printf("\n");
    switch(op){
        case '1':
            printf("\n\t\t\t\t\tPORCOES SERVEM 4 PESSOAS \n");
            printf("\t\tPeixe grelhado, acompanha arroz e batata doce - R$ 45,00 \n");
            printf("\t\tPeixe cozido, acompanha arroz e pirão - R$ 43,00 \n");
            printf("\t\tPeixe frito, acompanha tapioca e fritas - R$ 38,00 \n");
            break;
        case '2':
            printf("\n\t\t\t\t\tPORCOES INDIVIDUAIS \n");
            printf("\t\tAlface, rúcula, alcaparras, molho de mostarda - R$ 15,00 \n");
            printf("\t\tManjericao, tomate seco, rucula, alface, molho agrião - R$ 18,00 \n");
            printf("\t\tAlface, tomate seco, manjerona, alcaparras, molho de maracujá - R$ 22,00 \n");
            break;
        case '3':
            printf("\n\t\t\t\t\tPORCOES SERVEM 4 PESSOAS \n");
            printf("\t\tFrango grelhado, acompanha arroz e batata doce - R$ 45,00 \n");
            printf("\t\tFrango cozido, acompanha arroz e pirão - R$ 43,00 \n");
            printf("\t\tFrango frito, acompanha tapioca e fritas - R$ 38,00 \n");
            break;
```

# Problema (10/11)

- Continuação...

```
        case '4':
            printf("\n\t\t\tPORCOES SERVEM 4 PESSOAS \n");
            printf("\tCarne bovina grelhada, acompanha arroz e batata doce - R$ 45,00 \n");
            printf("\tCarne bovina cozida, acompanha arroz e pirão - R$ 43,00 \n");
            printf("\tCarne de porco frita, acompanha tapioca e fritas - R$ 38,00 \n");
            break;
        default:
            printf("Nao temos esta opcao no cardapio. Escolha outra!");
    }

    printf("\n");
    system ("PAUSE");
    return 0;
}
```

# Problema (11/11)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    float angulo, rang, result;
    int base, potencia, n1;
    char op;
    printf("\n\n\t\t\t\t\t*****\n");
    printf("\t\t\t\t\t OPERACOES\n");
    printf("\t\t\t\t\t MATEMATICAS\n");
    printf("\t\t\t\t\t*****\n");
    printf("\n\t\t\t\t\t1 raiz quadrada");
    printf("\n\t\t\t\t\t2 potencia");
    printf("\n\t\t\t\t\t3 seno");
    printf("\n\t\t\t\t\t4 tangente");
    printf("\n\n");
    printf("Digite o operador:");
    op=getche();
    printf("\n");
    if(op=='1'){
        printf("Entre com o valor para determinar raiz quadrada: ");
        scanf("%d", &n1);
        result=sqrt(n1);
    } else if(op=='2'){
        printf("Digite a base e a potência: ");
        scanf("%d %d", &base, &potencia);
        result=pow(base,potencia);
    } else if((op=='3') || (op=='4')){
        printf("Determine o angulo: ");
        scanf("%f", &angulo);
        rang=angulo*M_PI/180;
        if(op=='3')
            result=sin(rang);
        else if ((angulo!=90) || (angulo!=270))
            result=tan(rang);
    }
```



# Problema (11/11)

- Continuação...

```
else
    printf("Nao existe tangente deste angulo!");

}else
    printf("Operacao nao existe!");

printf("Resultado: %.2f", result);
printf("\n");
system ("PAUSE");
return 0;
}
```

# Lista de Exercícios

- 1. Faça um programa que leia dois valores inteiros A e B. Se os valores forem iguais, deverá se somar os dois, caso contrário, multiplique A por B. Ao final do cálculo escrever o resultado.
- 2. Construa um programa que entre com um número e informar se ele é divisível por 10, por 5, por 2 ou se não é divisível por nenhum destes.
- 3. A prefeitura abriu uma linha de crédito para os funcionários estatutários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Faça um programa em C que permita entrar com o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.

# Lista de Exercícios

- 4. O peso ideal para uma pessoa é calculado pelas seguintes fórmulas:

Sexo Masculino:  $(72.7 * \text{altura}) - 58$

Sexo Feminino:  $(62.1 * \text{altura}) - 44.7$

Escreva um programa que receba como entrada: sexo (f ou m), altura e peso; calcule o peso ideal usando a fórmula que for pertinente. Como resultado o programa deve exibir na tela: a diferença entre eles e uma mensagem de alerta:

“\*\*\* Peso normal ou abaixo do ideal \*\*\*” ou “\*\*\* Peso acima do Ideal \*\*\*”.

- 5. Construa um programa que calcule e apresente quanto deve ser pago por um produto considerando a leitura do preço da etiqueta (PE) e o código de pagamento (CP). Utilize para os cálculos a tabela de condições de pagamento a seguir:

# Lista de Exercícios

Código de pagamento	Condição de pagamento
1	À vista em dinheiro ou cheque, com 10% de desconto
2	À vista com cartão de crédito, com 5% de desconto
3	Em 2 vezes, preço normal de etiqueta sem juros
4	Em 3 vezes, preço de etiqueta com acréscimo de 10%

- 6. Construa um programa que leia os comprimentos dos três lados (A, B e C) de um triângulo, verifique se esses valores correspondem aos lados de um triângulo. Em caso afirmativo, informar ao usuário se o triângulo é equilátero, isóscele ou escaleno. Em caso negativo informar ao usuário que os valores não correspondem a um triângulo. Sabe-se que:
  - Para formar um triângulo, o valor de cada lado deve ser menor que a soma dos outros 2
  - Triângulo do tipo Equilátero – possui os três lados iguais
  - Triângulo do tipo Isóscele – possui dois lados iguais
  - Triângulo do tipo Escaleno – possui os três lados diferentes

# Lista de Exercícios

- 7. Construa um programa que leia as informações de: horas trabalhadas (HT) e valor da hora trabalhada (VH). Calcule e escreva o salário líquido, o valor do INSS descontado e o valor do IRPF, baseado nas tabelas abaixo.

**OBS:** Salário Bruto = Horas trabalhadas \* Valor da hora trabalhada

INSS = Salário Bruto \* Alíquota correspondente da tabela abaixo

Imposto de Renda = ((Salário Bruto – INSS) \* Alíquota correspondente da tabela abaixo)  
- parcela a deduzir na tabela abaixo

Salário Líquido = Salário Bruto – ( INSS + Imposto de Renda)

Tabela INSS	Alíquota
Até \$868,29	7,65%
De \$868,30 até \$1.447,14	8,00%
De \$1.447,15 até \$2.894,28	9,00%
Mais que \$2.894,28	11,00%

Tabela IRPF	Alíquota	Deduzir
Até \$1.372,81	Isento (0%)	-
De \$1.372,82 até \$2.743,25	15%	205,92
Mais que \$2.743,25	27,5%	548,82

# Lista de Exercícios

► 8. Dado o programa abaixo, responda:

- a) Se A= verdade, B= verdade, C= falsidade, quais comandos serão executados?
- b) Se A= falsidade, B= verdade, C= falsidade, quais comandos serão executados?
- c) Se A= falsidade, B= verdade, C= verdade, quais comandos serão executados?
- d) Quais os valores de A, B e C para que somente os comandos 5 e 6 sejam executados?
- e) Quais os valores de A, B e C para que somente o comando 6 seja executado?

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int A, B, C;
    if(A){
        printf("Comando 1!");
    } else if(B){
        if(C){
            printf("Comando 2!");
        } else{
            printf("Comando 3!");
            printf("Comando 4!");
        }
    }
    printf("Comando 5!");
}
printf("Comando 6!");
system("PAUSE");
return 0;
```

# Lista de Exercícios

- 9. Escreva um algoritmo que leia três valores inteiros e diferentes e mostre-os em ordem decrescente. Utilize para tal uma seleção encadeada.
- 10. Criar um programa que informe a quantidade total de calorias de uma refeição a partir da escolha do usuário que deverá informar o prato, a sobremesa e a bebida, conforme tabela abaixo:

PRATO	SOBREMESA	BEBIDA
Vegetariano 180cal	Abacaxi 75cal	Chá 20cal
Peixe 230cal	Sorvete diet 110cal	Suco de laranja 70cal
Frango 250cal	Mousse diet 170cal	Refrigerante diet 65cal
Carne 350cal	Bolo de chocolate 200cal	Suco de melão 100cal

- 11. A polícia rodoviária resolveu fazer cumprir a lei e cobrar dos motoristas o DUT. Sabe-se que o mês em que o emplacamento do carro deve ser renovado é determinado pelo último número da placa do mesmo. Criar um programa que, a partir da leitura do número da placa do carro, informe o mês em que o emplacamento deve ser renovado.

# Lista de Exercícios

- 12. Fazer um programa que leia uma data (no formato ddmmaaaa) e imprimir se a data é ou não válida.
- 13. Entrar com o valor de x e imprimir y:

$$Y = \begin{cases} 1, & \text{se } x \leq 1 \\ 2, & \text{se } 1 < x \leq 2 \\ x^2, & \text{se } 2 < x \leq 3 \\ x^3, & \text{se } x > 3 \end{cases}$$

- 14. Efetuar a leitura de cinco números reais diferentes e identificar o maior e o menor valor.
- 15. Ler um número inteiro de 4 casas e imprimir se é ou não múltiplo de quatro o número formado pelos algarismos que estão nas casas das unidades de milhar e das centenas.
- 16. Escrever um algoritmo 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.



# Referências

- Medina, Marco; Fertig, Cristina. Algoritmos e Programação: teoria e prática. São Paulo: Novatec Editora, 2006.
- Lopes, Anita; Garcia, Guto. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Editora Campus, 2002.
- Mizrani, Victorine Viviane. Treinamento em Linguagem C, Módulo 1. Editora Makron Books.
- Transparências modificadas do professor Dr. Flavio Luiz Cardeal Pádua, do Centro Federal de Educação Tecnológica de Minas Gerais