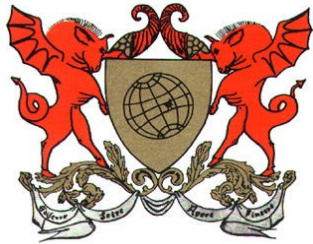


Universidade Federal de Viçosa  
Campus Rio Paranaíba  
Instituto de Ciências Exatas e Tecnológicas

# **SIN 110**

## **Programação**

Sistemas de Informação  
Prof. Rachel Reis  
[rachel.reis@ufv.br](mailto:rachel.reis@ufv.br)



Universidade Federal de Viçosa  
Campus Rio Paranaíba  
Instituto de Ciências Exatas e Tecnológicas

Aula de Hoje

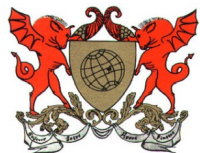
# Estruturas de Decisão em C

Créditos:

**Prof. Guilherme Pena**

# Indentação de Código

Na construção de programas em qualquer linguagem de programação, inclusive em C, um conceito muito importante é o de **INDENTAÇÃO** dos comandos daquele programa.



# Indentação de Código

A **indentação** trata-se do recuo de um texto em relação à sua margem e isso é uma obrigação na etapa de codificação visando a **legibilidade** do código e posterior **facilitação de sua manutenção**.

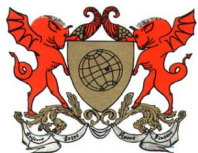
```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] == $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```



## Estética - Indentação

```
se (ano2 < ano1) entao
  escreva("data 2 menor que data 1")
senao
  se (ano2 > ano1) entao
    escreva("data 1 menor que data 2")
  senao
    se (mes2 < mes1) entao
      escreva("data 2 menor que data 1")
    senao
      se (mes2 > mes1) entao
        escreva("data 1 menor que data 2")
      senao
        se (dia2 < dia1) entao
          escreva("data 2 menor que data 1")
        senao
          se (dia2 > dia1) entao
            escreva("data 1 menor que data 2")
          senao
            escreva("Datas iguais")
          fimse
        fimse
      fimse
    fimse
  fimse
```

```
se (ano2 < ano1) entao
  escreva("data 2 menor que data 1")
senao
  se (ano2 > ano1) entao
    escreva("data 1 menor que data 2")
  senao
    se (mes2 < mes1) entao
      escreva("data 2 menor que data 1")
    senao
      se (mes2 > mes1) entao
        escreva("data 1 menor que data 2")
      senao
        se (dia2 < dia1) entao
          escreva("data 2 menor que data 1")
        senao
          se (dia2 > dia1) entao
            escreva("data 1 menor que data 2")
          senao
            escreva("Datas iguais")
          fimse
        fimse
      fimse
    fimse
  fimse
fimse
```



# Indentação de Código

Em geral, a indentação é feita sempre que houver a abertura de **chaves { }** ao longo do código e usamos um espaçamento aceitável. (O espaço de uma tabulação por exemplo |     | )

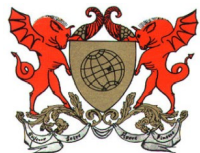
Exemplo:

## Código Não-Indentado.

```
#include <stdio.h>
int main() {
float nota;
printf("Digite a nota do aluno:");
scanf("%f", &nota);
printf("A nota digitada e : %f\n", nota);
system("pause");
return 0;
}
```

## Código Indentado.

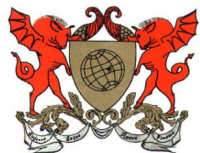
```
#include <stdio.h>
int main() {
    float nota;
    printf("Digite a nota do aluno:");
    scanf("%f", &nota);
    printf("A nota digitada e : %f\n", nota);
    system("pause");
    return 0;
}
```



# Estruturas de Decisão

Uma **estrutura de decisão**, também chamada de **estrutura de seleção**, é a primeira forma que altera o fluxo de ações permitindo a construção de algoritmos mais poderosos.

Ela permite a escolha de um grupo de ações (instruções, comandos) que são executados quando determinadas condições são ou não satisfeitas.



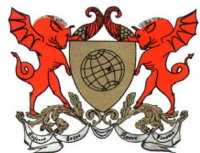
# Estruturas de Decisão

Exemplo:

"se tiver R\$ 20,00 sobrando então irei ao cinema hoje à noite."

Qual a condição nessa frase?

- ter R\$ 20,00 sobrando..

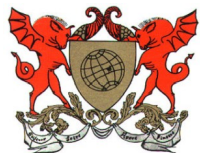


# Estruturas de Decisão

Exemplo:

"se tiver R\$ 20,00 sobrando então irei ao cinema hoje à noite."

Ela é uma **expressão lógica**, pois a pergunta "Tenho R\$ 20,00 sobrando?" Pode (tem que) ser respondida com "**Sim**" ou "**Não**".



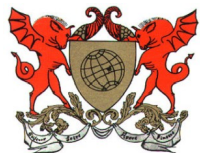


# Estruturas de Decisão

Exemplo:

"se tiver R\$ 20,00 sobrando então irei ao cinema hoje à noite."

Lembre-se, então: em um algoritmo, toda condição tem que ser uma expressão lógica, algo que possa-se pensar como "isto é **VERDADEIRO**" ou "isto é **FALSO**".



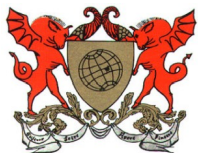
# Estruturas de Decisão

Exemplo:

"se tiver R\$ 20,00 sobrando então irei ao cinema hoje à noite."

Se a condição for **verdadeira** ou a resposta ser "**sim**", a ação a ser executada é "irei ao cinema hoje a noite".

Então, em um algoritmo, as ações são um ou mais comandos que serão realizados apenas se a avaliação da condição resulta **VERDADEIRO**.



# Estruturas de Decisão

## Exemplo:

A mãe falou para o filho estudante de SIN110:

- “Filho, vá ao mercado e traga 1 garrafa de leite. Se eles tiverem ovos, então traga 6.”

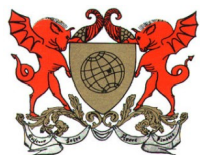
O menino volta do mercado com 6 garrafas de leite.

A mãe pergunta:

- “Porque você trouxe 6 garrafas de leite ??”

Ele respondeu:

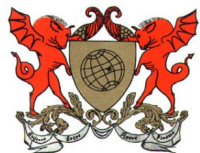
- “Porque eles tinham ovos!”



# Estruturas de Decisão em C

## **Operadores mais usados na decisão:**

- Operadores Relacionais
- Operadores Lógicos
- Combinação de ambos

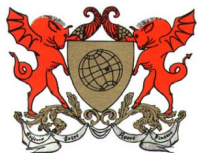


# Operadores Relacionais

Aqueles usados para **comparar valores** de um mesmo tipo primário (int, char, float).

Operadores Relacionais	Em linguagem C
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

O resultado é um valor **0 (FALSO)** ou **1 (VERDADEIRO)**.

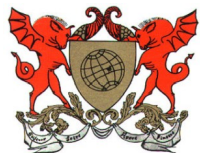


# Operadores Relacionais

Exemplos:

Operadores Relacionais	C	Exemplos
Maior	>	$3 > 3$ 0 (Falso)
Menor	<	$10 < 20$ 1 (Verdadeiro)
Maior ou igual	>=	$32.5 >= 5.7$ 1 (Verdadeiro)
Menor ou igual	<=	$4 <= 2$ 0 (Falso)
Igual	==	$'a' == 'g'$ 0 (Falso)
Diferente	!=	$'w' != 'r'$ 1 (Verdadeiro)

O resultado é um valor **0 (FALSO)** ou **1 (VERDADEIRO)**.

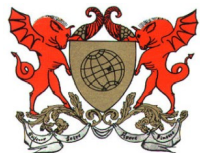


# Operadores Lógicos

Aqueles usados para **conectar expressões relacionais ou lógicas**.

Operadores Lógicos	Em linguagem C
E	&&
OU	
NÃO	!

O resultado é um valor **0 (FALSO)** ou **1 (VERDADEIRO)**.

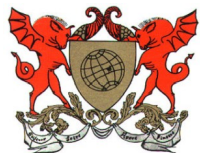


# Operadores Lógicos

Operador lógico “E”:

Se **fizer sol** **e** eu **tiver roupa de banho**, então eu vou para a praia.

Quando eu vou para a praia?





# Operadores Lógicos (E)

## Tabela-Verdade (E):

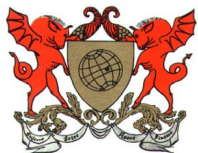
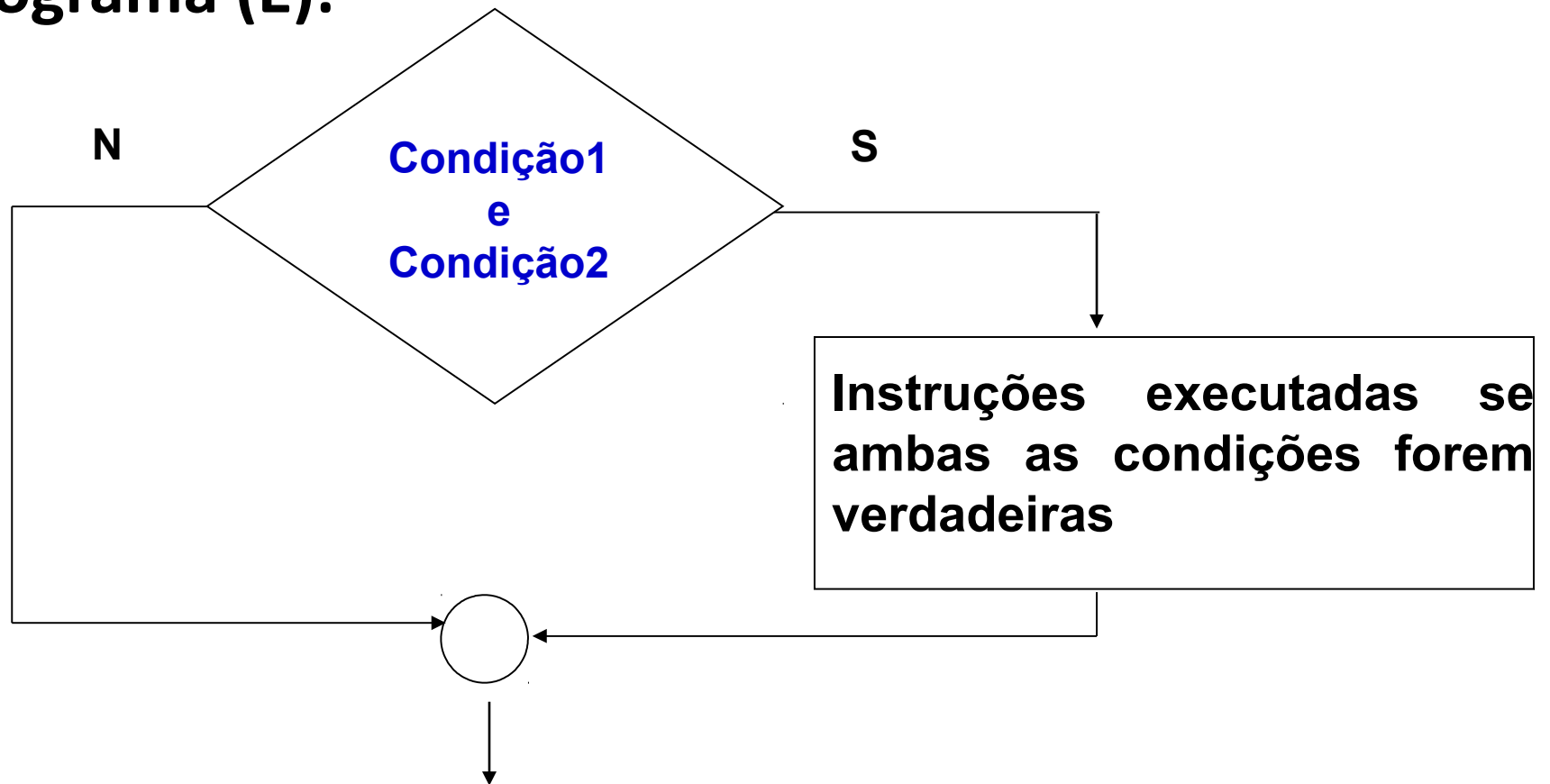
Condição 1	Condição 2	Resultado
Verdadeira	Verdadeira	Verdadeiro
Verdadeira	Falsa	Falso
Falsa	Verdadeira	Falso
Falsa	Falsa	Falso

Se **fizer sol** **e** **eu tiver roupa de banho**, então eu vou para a praia.



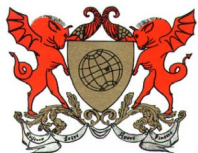
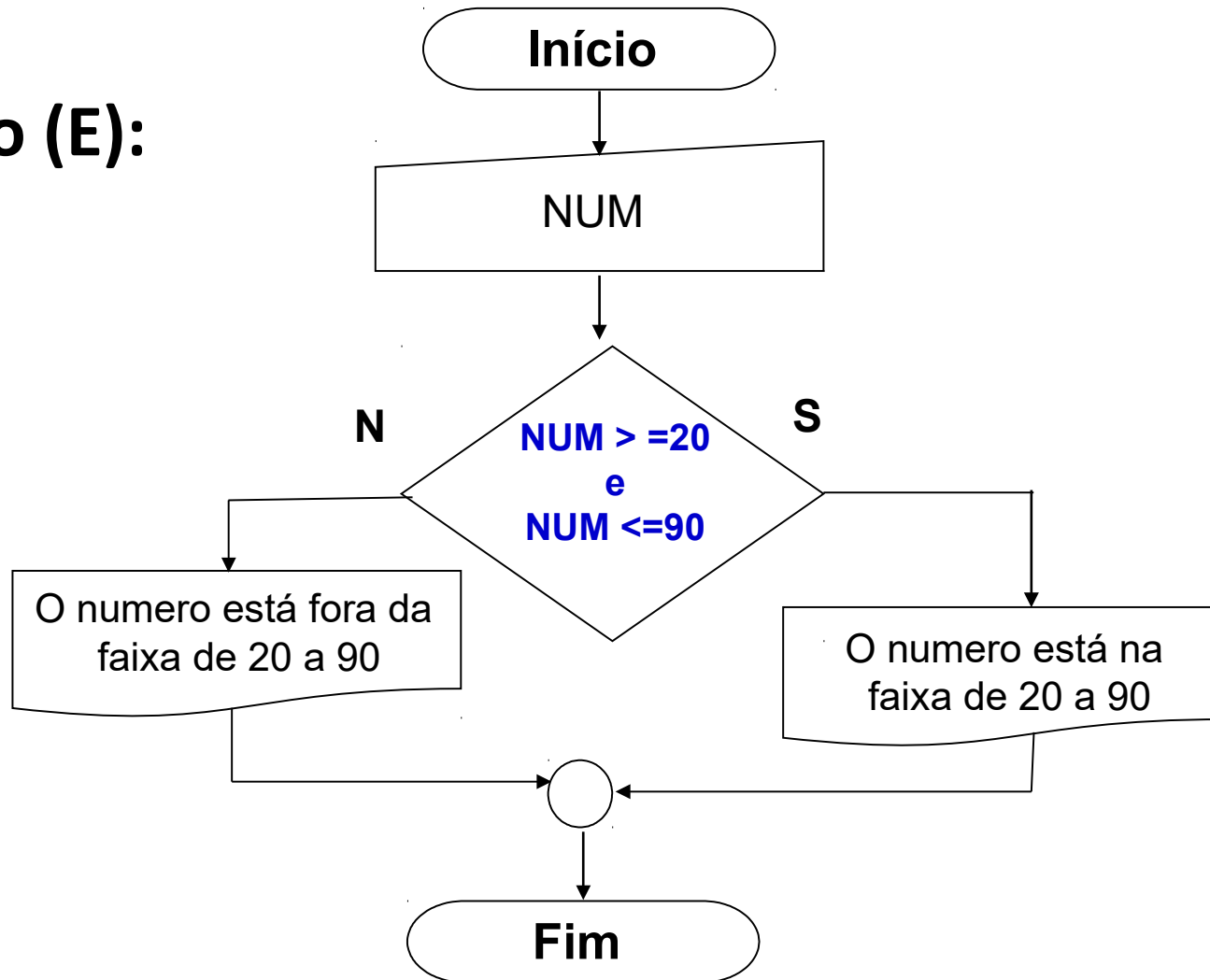
# Operadores Lógicos (E)

## Fluxograma (E):



# Operadores Lógicos (E)

Exemplo (E):



# Operadores Lógicos (E)

## Exemplo (E):

```
int main(){
    int num;
    scanf("%d", &num);
    if( (num >= 20) && (num <=90) ){
        printf("O número está na faixa de 20 a 90");
    }else{
        printf("O número está fora da faixa de 20 a 90");
    }
    return 0;
}
```

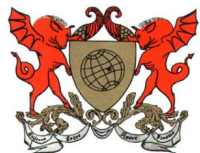


# Operadores Lógicos

Operador lógico “**OU**”:

Se **fizer sol** **ou** eu **tiver roupa de banho**, então eu vou para a praia.

Quando eu vou para a praia?

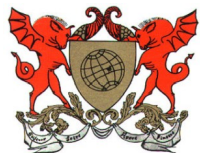


# Operadores Lógicos (OU)

## Tabela-Verdade (OU):

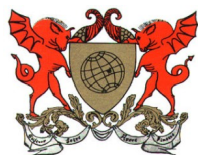
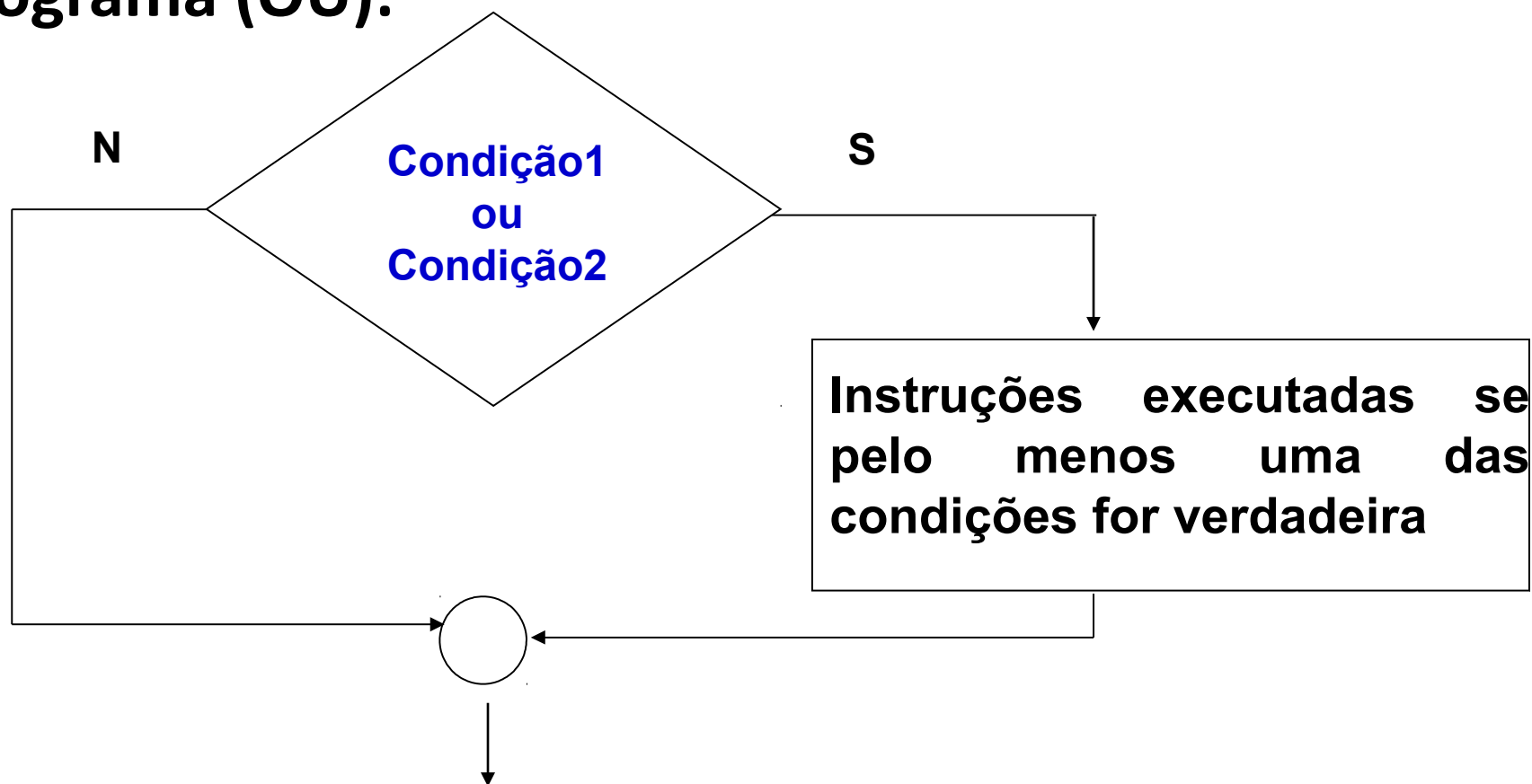
Condição 1	Condição 2	Resultado
Verdadeira	Verdadeira	Verdadeiro
Verdadeira	Falsa	Verdadeiro
Falsa	Verdadeira	Verdadeiro
Falsa	Falsa	Falso

Se **fizer sol** **ou** **eu tiver roupa de banho**, então eu vou para a praia.



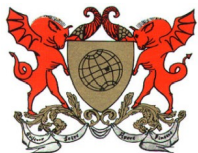
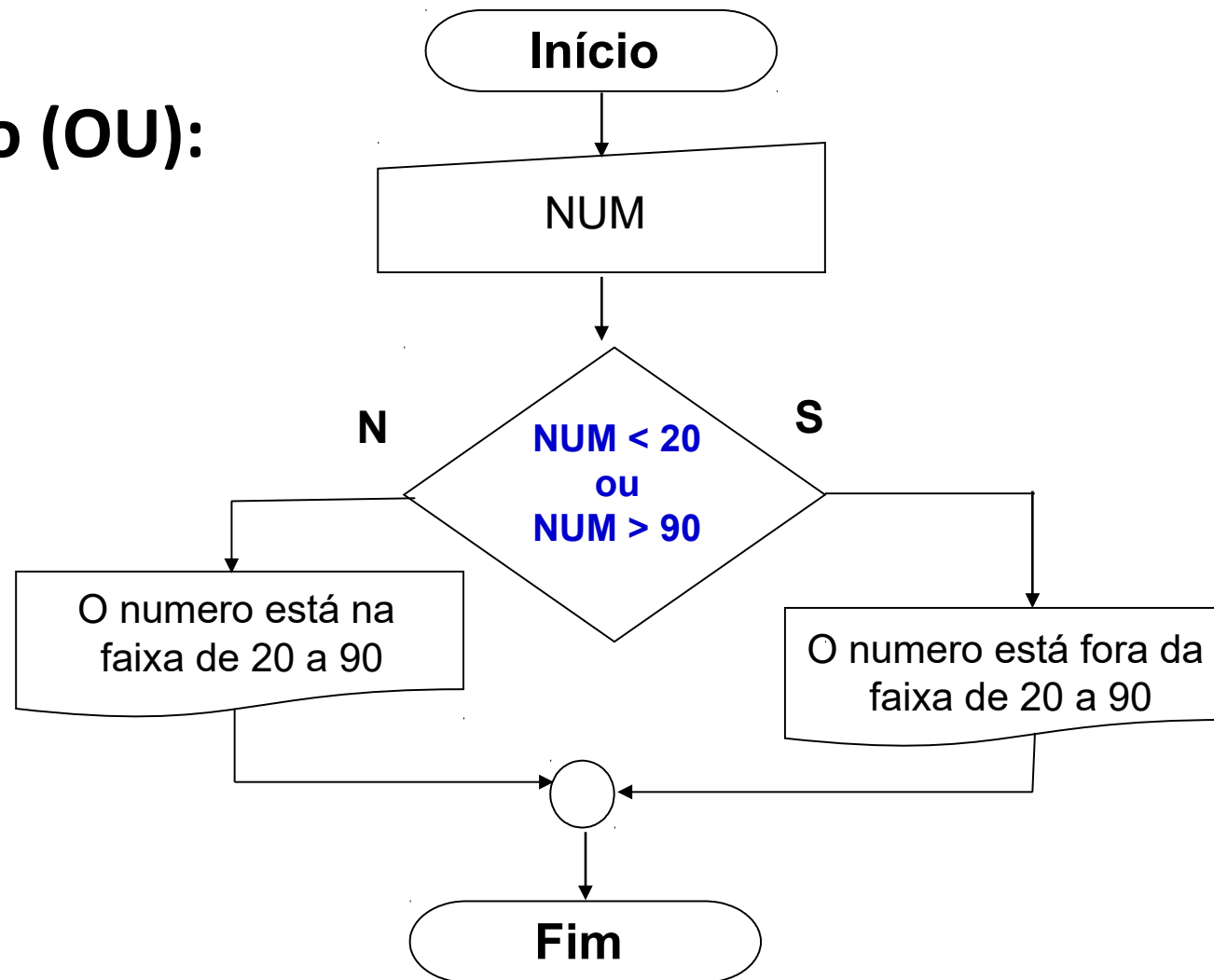
# Operadores Lógicos (OU)

## Fluxograma (OU):



# Operadores Lógicos (E)

Exemplo (OU):





# Operadores Lógicos (E)

## Exemplo (OU):

```
int main(){
    int num;
    scanf("%d", &num);
    if( (num < 20) || (num > 90) ){
        printf("O número está fora da faixa de 20 a 90");
    }else{
        printf("O número está na faixa de 20 a 90");
    }
    return 0;
}
```



# Operadores Lógicos

Operador lógico “**NAO**”:

Se **nao** chover, então eu vou para a praia.

Quando eu vou para a praia?



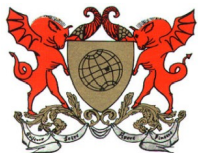
# Operadores Lógicos (NAO)

O operador do tipo **não** é utilizado quando tem necessidade de estabelecer a inversão do resultado lógico de uma determinada condição.

**Tabela-Verdade (NAO):**

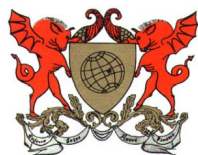
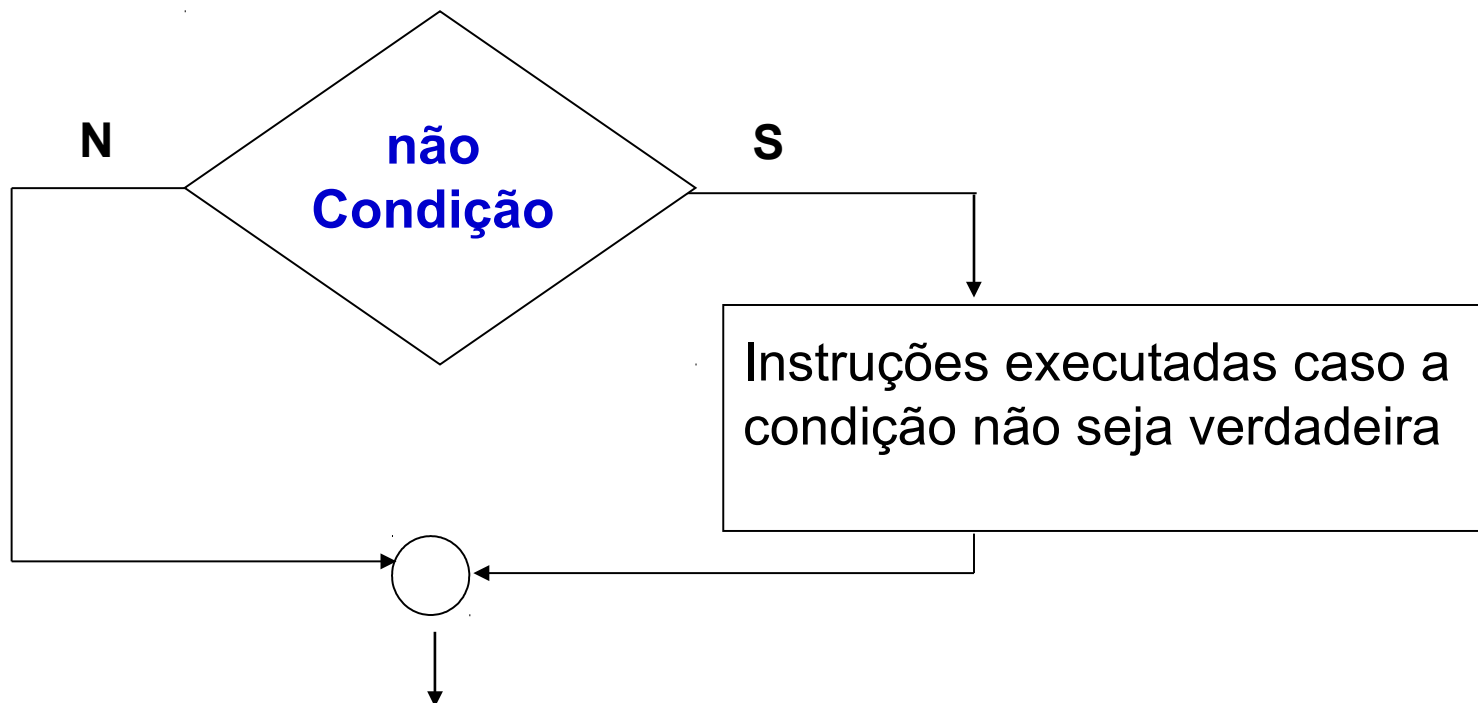
Condição	Resultado
Verdadeira	Falso
Falsa	Verdadeiro

Se **nao** chover, então eu vou para a praia.



# Operadores Lógicos (NAO)

## Fluxograma (NAO):



# Operadores Lógicos (NAO)

## Exemplo (NAO):

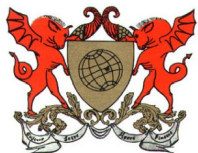
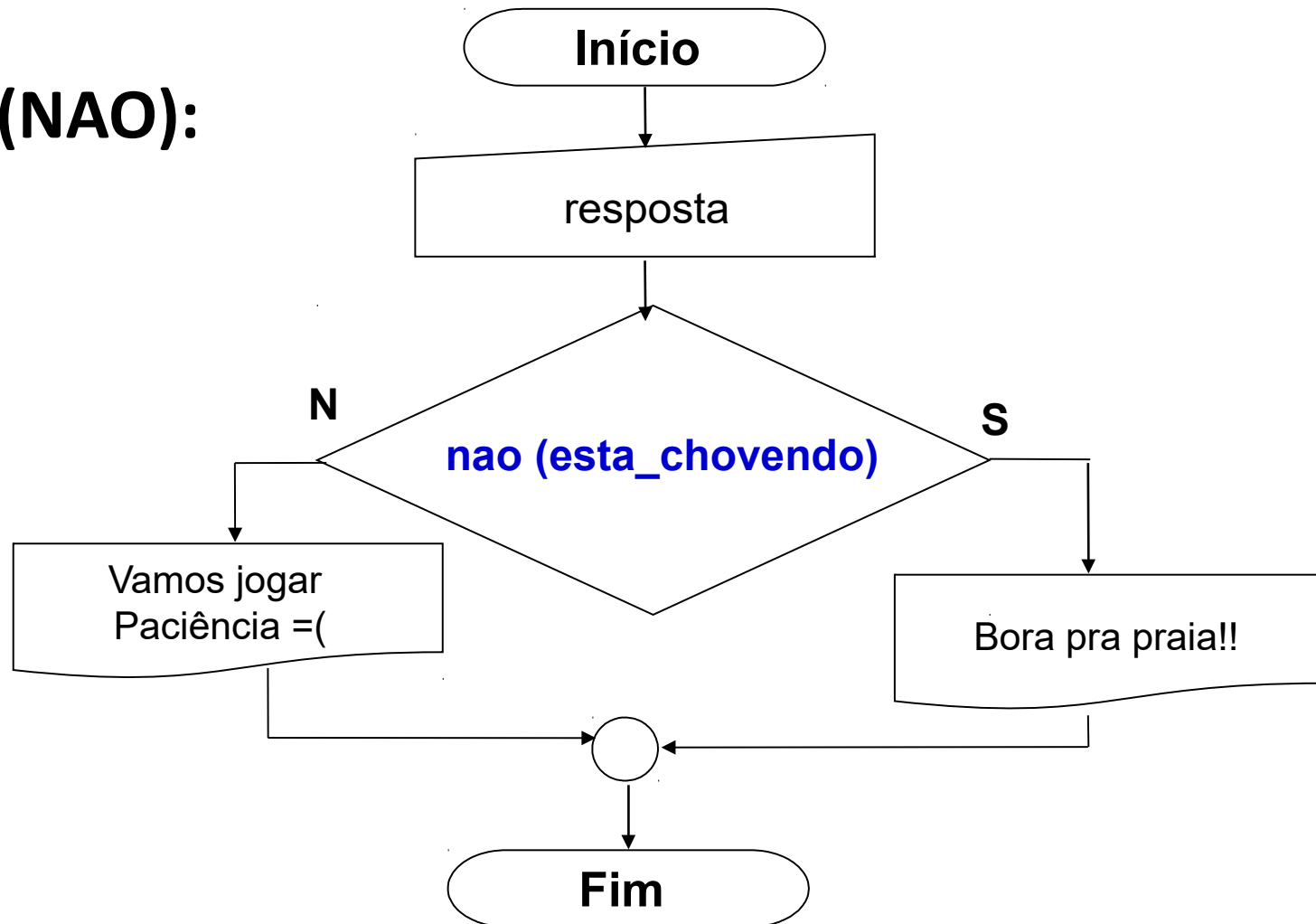
### Programa **Computador Brother**

```
int main(){
    int esta_chovendo;
    char resposta;
    printf("Está chovendo? Responda S ou N\n");
    scanf("%c", &resposta);
    esta_chovendo = (resposta == 'S');
    if (!(esta_chovendo)) {
        printf("Bora pra praia!!\n");
    }else{
        printf("Vamos jogar Paciência =((( \n");
    }
    return 0;
}
```



# Operadores Lógicos (NAO)

Exemplo (NAO):



# Estruturas de Decisão

É importante frisar que sempre à direita do comando **if** deverá parecer uma **expressão lógica**, ou seja, uma expressão cujo resultado é **VERDADEIRO** ou **FALSO**.

**Exemplo: (Incorreto)**

```
if (A = B) { // É uma atribuição e não uma expressão
```

```
...  
}
```

```
if (A + B) { // Expressão aritmética e não lógica
```

```
...  
}
```



# Estruturas de Decisão

É importante frisar que sempre à direita do comando **se** deverá parecer uma **expressão lógica**, ou seja, uma expressão cujo resultado é **VERDADEIRO** ou **FALSO**.

**Exemplo: (Correto)**

```
if ( (A > B) && (A > C) && (B != C) ){
```

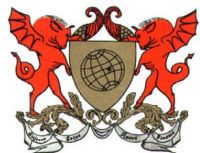
```
...
```

```
}
```

```
if ( ! Achou ){ // Correto se Achou é tratada como tipo logico
```

```
...
```

```
}
```





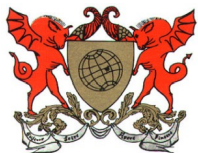
# Operadores Lógicos

## Tabela Verdade:

Vamos combinar o seguinte:

- **VERDADEIRO** → 1(um)
- **FALSO** → 0 (zero)

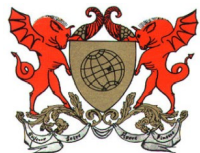
A	B	A && B	A    B	!A	!B
1	1	1	1	0	0
1	0	0	1	0	1
0	1	0	1	1	0
0	0	0	0	1	1



# Operadores Lógicos

## Exemplos:

- **`(4 > 3) && ('a' != 'b)`** 1 (Verdadeiro)
- `num = 10;`  
**`(num < 0 || num > 100)`** 0 (Falso)
- `valor = -10;`  
**`(!valor)`** 0 (Falso)



# Estruturas de Decisão em C

## **Tipos de Estruturas de Seleção em C:**

- Seleção Simples
- Seleção Composta e Operador Ternário
- Seleção Encadeada (Aninhada)
- Múltipla escolha



# Estruturas de Decisão

"**se** tiver R\$ 20,00 sobrando **então** irei ao cinema hoje à noite."

"... **Se** eles tiverem ovos, **então** traga 6. ..."

**se** tiver R\$ 20,00 sobrando **então**  
irei ao cinema hoje à noite.  
**fimse**

**se** eles tiverem ovos **então**  
traga 6.  
**fimse**



# Estrutura Condicional Simples

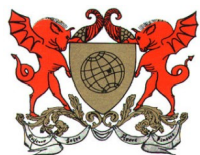
Linguagem C:

Comando if

```
if (<condição>)  
    instrução
```

O comando **if** instrui o programa a tomar uma decisão simples.

A **instrução** (ou **instruções**) só será executada se a **condição** for **verdadeira**.

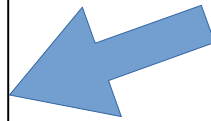


# Estrutura Condicional Simples

Linguagem C:

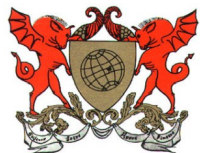
Comando if

```
if (<condição>)  
    instrução;
```



O comando **if** instrui o programa a tomar uma decisão simples.

Quando se têm apenas uma instrução, o uso de chaves é **opcional**.



# Estrutura Condicional Simples

Linguagem C:

Comando if

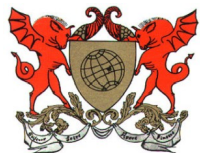
```
if (<condição>)
```

```
    instrução 1
```

```
    instrução 2
```

```
    ...
```

```
    instrução n
```



# Estrutura Condicional Simples

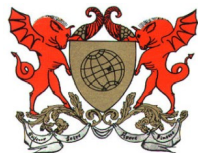
## Linguagem C:

```
if (<condição>) {  
    instrução 1;  
    instrução 2;  
    ...  
    instrução n;  
}
```

## Comando if

```
if (<condição>)  
{  
    instrução 1;  
    instrução 2;  
    ...  
    instrução n;  
}
```

Quando se tem **mais de uma instrução** para o mesmo **if** é **obrigatório** o uso de **chaves**.

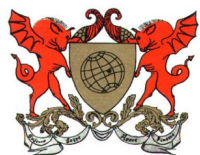




# Estrutura Condicional Simples

## Exemplo (**Erro comum**):

```
#include <stdio.h>
int main( )
{
    int num, den;
    printf("Digite um numerador e um denominador:");
    scanf("%d %d", &num, &den);
    if (den > 0)
        printf("A divisao pode ser feita.\n");
        printf("%d/%d = %d\n", num, den, num/den);
    system("pause");
    return 0;
}
```

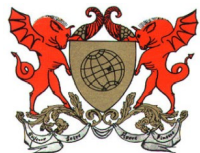


# Estrutura Condicional Simples

## Exemplo:

```
#include <stdio.h>

int main( )
{
    int num, den;
    printf("Digite um numerador e um denominador:");
    scanf("%d %d", &num, &den);
    if (den > 0){
        printf("A divisao pode ser feita.\n");
        printf("%d/%d = %d\n", num, den, num/den);
    }
    system("pause");
    return 0;
}
```



# Estrutura Condicional Simples

## Exemplo (**Erro Comum**):

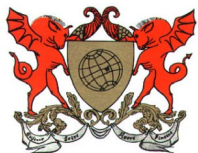
```
#include <stdio.h>
int main( )
{
    float nota;
    printf("Digite uma nota de 0 a 100: ");
    scanf("%f", &nota);
    if ( (nota < 0) || (nota > 100) );{
        printf("Nota invalida.\n");
    }
    system("pause");
    return 0;
}
```



# Estrutura Condicional Simples

## Exemplo:

```
#include <stdio.h>
int main( )
{
    float nota;
    printf("Digite uma nota de 0 a 100: ");
    scanf("%f", &nota);
    if ( (nota < 0) || (nota > 100) ){
        printf("Nota invalida.\n");
    }
    system("pause");
    return 0;
}
```

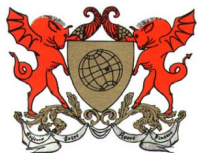


# Seleção Simples

## Exemplo:

Seja um algoritmo que calcule a média aritmética entre quatro notas fornecidas por um aluno. O aluno será aprovado se a média for maior ou igual a 7.

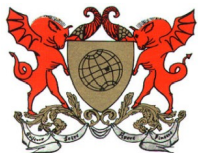
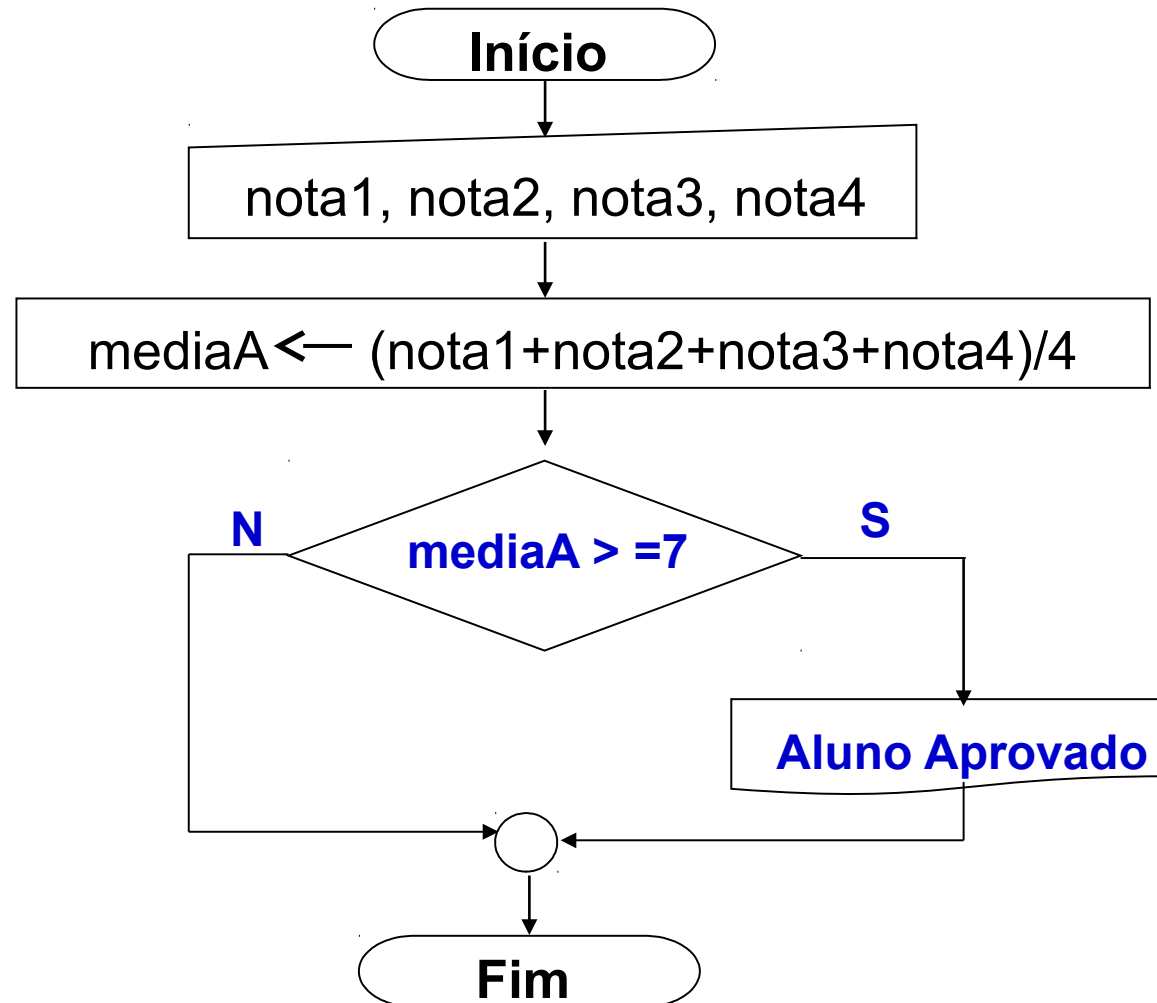
```
var
    nota1, nota2, nota3, nota4, mediaA: real
inicio
    leia(nota1, nota2, nota3, nota4)
    mediaA ← (nota1+nota2+nota3+nota4)/4
    se (mediaA >= 7) então
        escreva("Aluno Aprovado")
    fimse
finalgoritmo
```



# Seleção Simples

**Exemplo:**

**Fluxograma ->**

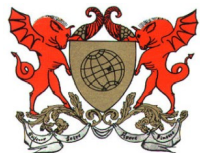


# Seleção Simples

## Exercício:

Fazer um programa para calcular e imprimir a área de um círculo cujo raio o usuário digita. O cálculo e a impressão só é realizado se o raio for positivo, caso contrário uma mensagem deve ser impressa.

$$\text{Área} = \text{PI} * \text{raio}^2$$



# Estruturas de Decisão

## Seleção Composta:

Usado quando tivermos situações em que **duas alternativas dependem de uma mesma condição.**

Onde quando a condição for verdadeira o programa toma uma ação ou quando a condição é falsa o programa toma outra ação.

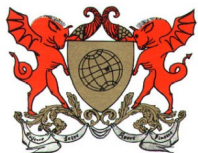




# Seleção Composta

**Exemplo:** ... aprovado se a média for maior ou igual a 7 e reprovado caso contrário.

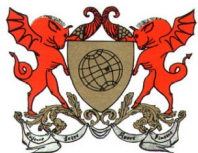
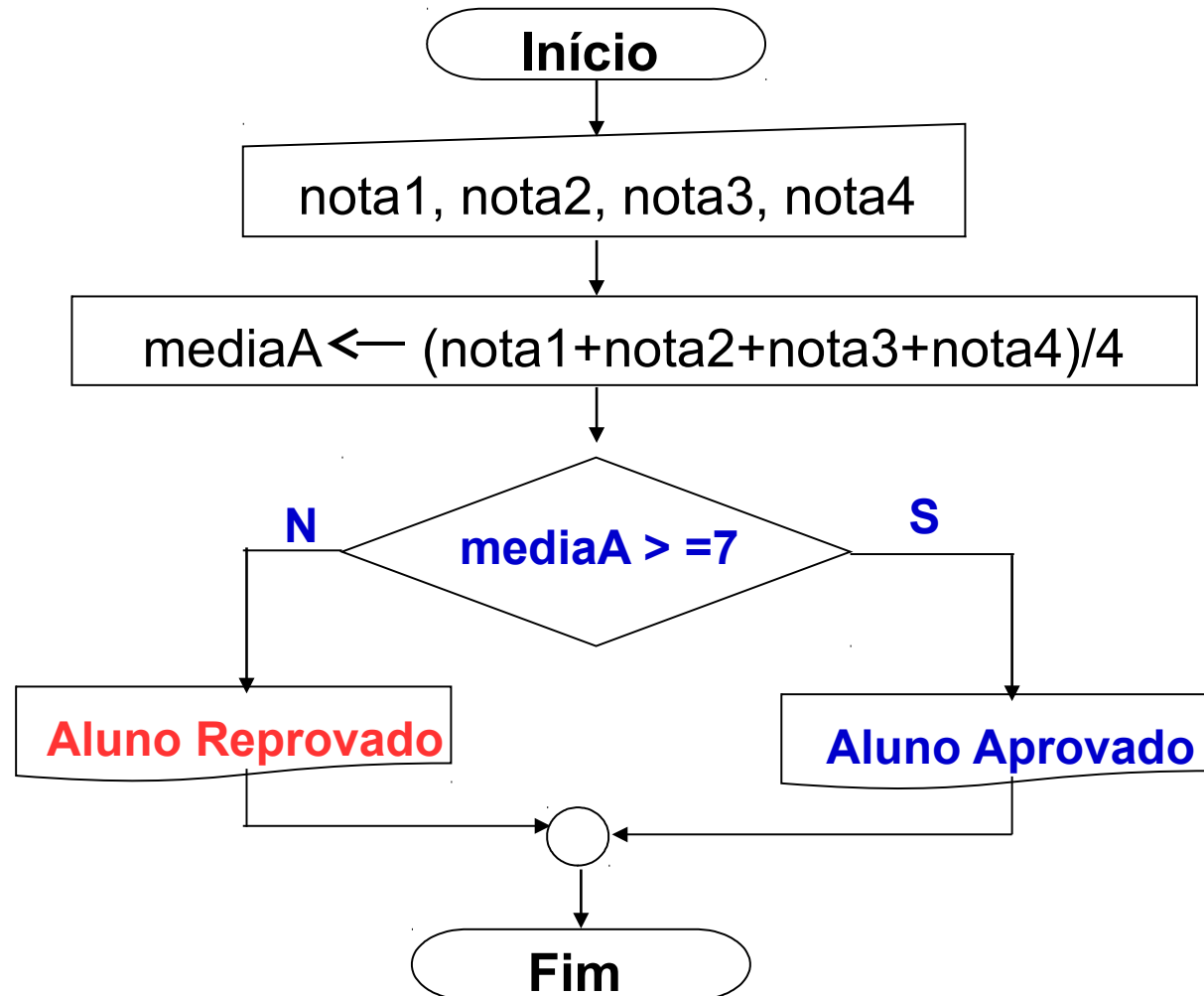
```
var
    nota1, nota2, nota3, nota4, mediaA: real
inicio
    leia(nota1, nota2, nota3, nota4)
    mediaA ← (nota1+nota2+nota3+nota4)/4
    se (mediaA >= 7) então
        escreva("Aluno Aprovado")
    senao
        escreva("Aluno Reprovado")
    fimse
finalgoritmo
```



# Seleção Composta

**Exemplo:**

**Fluxograma ->**



# Estrutura Condicional Composta

Linguagem C:

Comando if-else

```
if (<condição>)
```

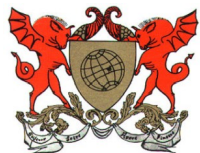
```
    instrução 1;
```

```
else
```

```
    instrução 2;
```

Caso **condição** seja **VERDADEIRA**, então **instrução 1** será executada.

Caso **condição** seja **FALSA**, então **instrução 2** será executada.



# Estrutura Condicional Composta

Linguagem C:

Comando if-else

```
if (<condição>)
```

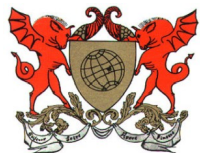
```
    instrução 1;
```

```
else
```

```
    instrução 2;
```

O comando **else** não requer condição, pois a mesma já é avaliada no respectivo **if**.

Pode-se dizer que é um **complemento** do comando **if**.



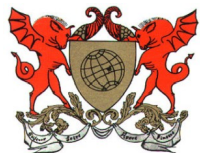
# Estrutura Condicional Composta

## Linguagem C:

## Comando if-else

```
if (<condição>) {  
    instrução 1;  
} else {  
    instrução 2;  
}
```

O uso das chaves segue o mesmo princípio do **if**, caso o bloco possua **mais de um comando** deve-se **usar chaves**.

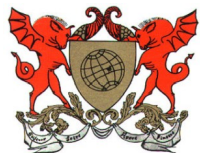


# Estrutura Condicional Composta

## Linguagem C:

## Comando if-else

```
if (<condição>)  
{  
    instrução 1;  
    ...  
    instrução n;  
}  
else  
{  
    instrução 2;  
    ...  
    instrução m;  
}
```



# Estrutura Condicional Composta

Linguagem C:                      Comando if-else

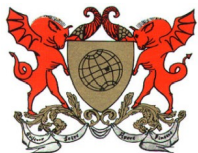
**Observação:** Na ausência de chaves para indicar o respectivo **if**, o comando **else** se ligará ao **if** mais próximo.



# Estrutura Condicional Composta

## Exemplo:

```
#include <stdio.h>
int main( )
{
    int num;
    printf("Digite um numero inteiro: ");
    scanf("%d", &num);
    if (num > 0)
        if (num%2 == 0)
            printf("Numero digitado e positivo e par.\n");
    else
        printf("Numero digitado e nulo ou negativo.\n");
    system("pause");
    return 0;
}
```

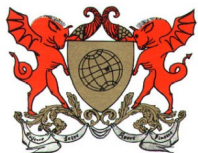




# Estrutura Condicional Composta

## Exemplo:

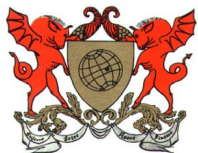
```
#include <stdio.h>
int main( )
{
    int num;
    printf("Digite um numero inteiro: ");
    scanf("%d", &num);
    if (num > 0){
        if (num%2 == 0)
            printf("Numero digitado e positivo e par.\n");
        }else
            printf("Numero digitado e nulo ou negativo.\n");
    system("pause");
    return 0;
}
```



# Estrutura Condicional Composta

## Exemplo (**Erro Comum**):

```
#include <stdio.h>
int main( )
{
    int num;
    printf("Digite um numero inteiro: ");
    scanf("%d", &num);
    if (num > 0){
        if (num%2 == 0)
            printf("Numero digitado e positivo e par.\n");
    }else
        printf("Numero digitado e nulo ou negativo.\n");
    system("pause");
    return 0;
}
```



# Estrutura Condicional Composta

## Observações:

- Fiquemos atentos ao uso dos operadores corretos.
- Fiquemos atentos ao uso de ponto-e-vírgula.
- Na dúvida sobre usar **chaves**, sempre use-as.



# Estrutura Condicional Composta

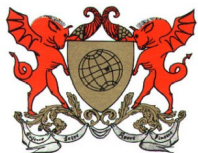
## Operador Condicional Ternário:

O operador condicional ternário é equivalente ao comando if-else.

Ele apresenta uma forma compacta de escrever uma operação simples no if-else.

**Forma Geral: ( ? : )**

```
(<condição>) ? expressão_1: expressão_2;
```



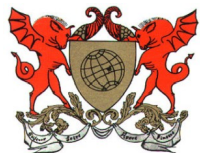
# Estrutura Condicional Composta

## Forma Geral:

```
(<condição>) ? expressão_1 : expressão_2 ;
```

A **condição** equivale a uma expressão lógica avaliada como **verdadeiro** e **falso**.

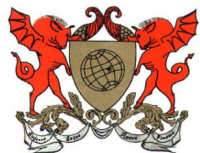
Se a **condição** é verdadeira (não zero) a **expressão\_1** é executada; se falsa a **expressão\_2** é executada.



# Estrutura Condicional Composta

## Exemplo:

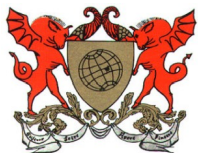
```
#include <stdio.h>
int main( )
{
    int n1, n2, maior;
    printf("Digite dois numeros inteiros: ");
    scanf("%d %d", &n1, &n2);
    if (n1 > n2){
        maior = n1;
    }else{
        maior = n2;
    }
    printf("O maior e: %d\n", maior);
    system("pause");
    return 0;
}
```



# Estrutura Condicional Composta

## Exemplo:

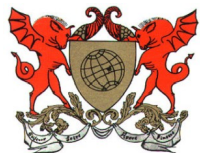
```
#include <stdio.h>
int main( )
{
    int n1, n2, maior;
    printf("Digite dois numeros inteiros: ");
    scanf("%d %d", &n1, &n2);
    maior = (n1 > n2) ? n1 : n2 ;
    printf("O maior e: %d\n", maior);
    system("pause");
    return 0;
}
```



# Estrutura Condicional Composta

## Exemplo:

```
#include <stdio.h>
int main( )
{
    int n;
    printf("Digite um numero inteiro: ");
    scanf("%d", &n);
    if(n%2 == 0)
        printf("Numero e PAR.\n");
    else
        printf("Numero e IMPAR.\n");
    system("pause");
    return 0;
}
```



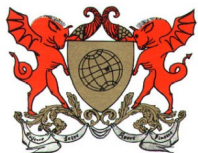


# Estrutura Condicional Composta

## Exemplo:

```
#include <stdio.h>

int main( )
{
    int n;
    printf("Digite um numero inteiro: ");
    scanf("%d", &n);
    printf("Numero e ");
    (n%2 == 0) ? printf("PAR.\n") : printf("IMPAR.\n");
    system("pause");
    return 0;
}
```

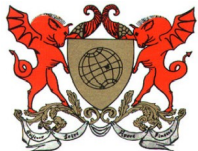


# Seleção Composta

## Exercício:

Fazer um algoritmo para calcular e imprimir a área de um círculo cujo raio o usuário digita. O cálculo e a impressão só é realizado se o raio for positivo, caso contrário uma mensagem deve ser impressa.

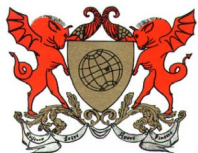
$$\text{Área} = \text{PI} * \text{raio}^2$$



# Seleção Encadeada

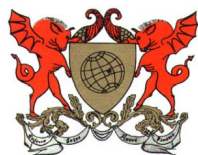
Em algumas situações é necessário termos várias soluções ligadas a respostas diferentes.

Os comandos de seleção **simples** ou **composta** não são uma solução prática, porque obrigará o programador a escrever **muitas linhas de programa**.



# Seleção Encadeada ou Aninhada

Trata-se do agrupamento de várias estruturas de seleção sendo uma dentro da outra.



# Seleção Encadeada ou Aninhada

```
if (<condição 1>
    if (<condição 2>
        Instrução a executar..
    else
        if (<condição 3>
            Instrução a executar..
        else
            if (<condição 4>
                Instrução a executar..
            else
                Instrução a executar..
    else
        if (<condição 5>
            Instrução a executar..
        else
            Instrução a executar..
```



# Estruturas Condicionais Aninhadas

```
if (<condição 1>
    if (<condição 2>
        Instrução a executar..
    else if (<condição 3>)
        Instrução a executar..
    else if (<condição 4>)
        Instrução a executar..
    else
        Instrução a executar..
else if (<condição 5>)
    Instrução a executar..
else
    Instrução a executar..
```



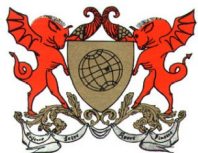
# Estruturas Condicionais Aninhadas

```
if (<condição 1>){  
    if (<condição 2>){  
        Instruções a executar..  
    }else{  
        if (<condição 3>){  
            Instruções a executar..  
        }else{  
            if (<condição 4>){  
                Instruções a executar..  
            }else{  
                Instruções a executar..  
            }  
        }  
    }  
}else{  
    if (<condição 5>){  
        Instruções a executar..  
    }else{  
        Instruções a executar..  
    }  
}
```



# Estruturas Condicionais Aninhadas

```
if (<condição 1>){  
    if (<condição 2>){  
        Instruções a executar..  
    }else{ if (<condição 3>){  
        Instruções a executar..  
    }else{ if (<condição 4>){  
        Instruções a executar..  
    }else{  
        Instruções a executar..  
    }  
}  
}else{ if (<condição 5>){  
    Instruções a executar..  
}else{  
    Instruções a executar..  
}  
}
```

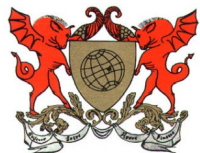




# Estruturas Condicionais Aninhadas

**Exemplo:**

```
if (num == 1)
    printf("Numero igual a 1\n");
else if (num == 2)
    printf("Numero igual a 2\n");
else if (num == 3)
    printf("Numero igual a 3\n");
else if (num == 4)
    printf("Numero igual a 4\n");
else if (num == 5)
    printf("Numero igual a 5\n");
else
    printf("Numero invalido.\n");
```



# Seleção Encadeada

## Exercício

Faça um programa que efetue o cálculo do reajuste de salário de um funcionário:

- Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500.
- Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%;
- Caso seja ainda maior que 1000 o reajuste deverá ser de 5%”

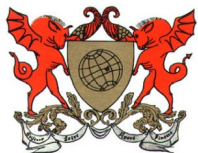


# Seleção Encadeada

## Exercício

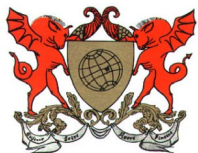
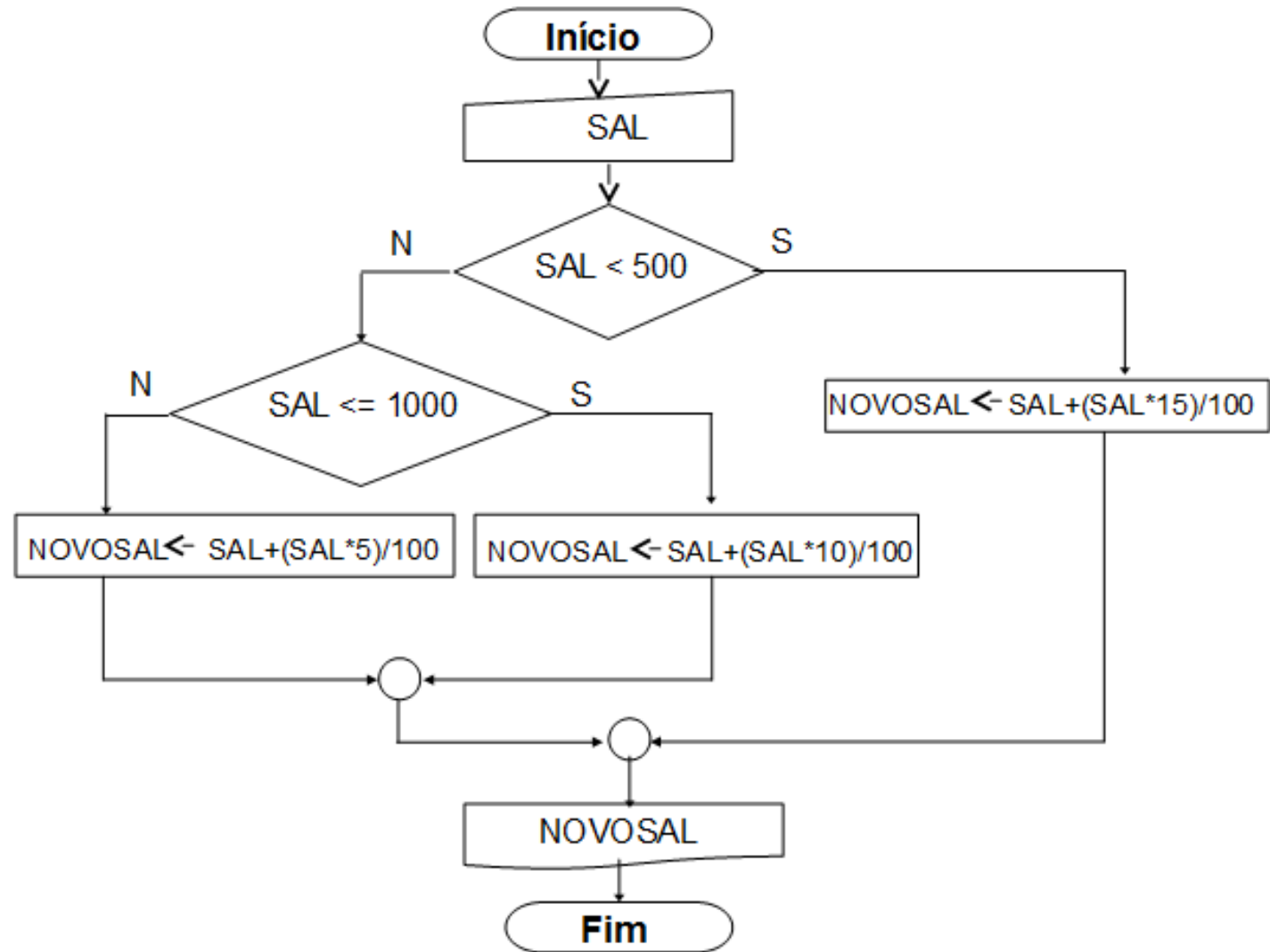
O problema estabelece três condições para calcular o reajuste do salário:

- Salário  $< 500$ , reajuste de 15%
- Salário  $\geq 500$ , mas  $\leq 1000$ , reajuste de 10%
- Salário  $> 1000$ , reajuste de 5%



# Seleção Encadeada

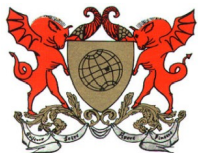
## Exercício



# Seleção Encadeada

## Exercício

```
int main(){
    float salario, novoSalario;
    scanf("%f", &salario);
    if (salario < 500.0) {
        novoSalario = salario+(salario*15)/100;
    }else{
        if (salario <=1000.0) {
            novoSalario = salario+(salario*10)/100;
        }else{
            novoSalario = salario+(salario*5)/100 ;
        }
    }
    printf("%f\n", novoSalario);
}
```



# Estrutura Condicional de Múltipla Escolha

## Comando **switch**

útil quando o programa necessita escolher uma entre várias alternativas.

formato mais limpo e claro comparado ao if-else-if.

dependendo do valor de uma variável, uma operação diferente é executada.



# Estrutura Condicional de Múltipla Escolha

**Forma Geral:**

```
switch (expressão (int ou char))
{
    case valor1:
        instruções;
        break;
    case valor2:
        instruções;
        break;
    ...
    case valorN:
        instruções;
        break;
    default:
        instruções;
}
```



# Estrutura Condicional de Múltipla Escolha

```
switch (expressão (int ou char))
```

```
{  
    case valor1:  
        instruções;  
        break;  
    case valor2:  
        instruções;  
        break;  
    ...  
    case valorN:  
        instruções;  
        break;  
    default:  
        instruções;  
}
```

O comando **switch** avalia a expressão entre parênteses, após a palavra switch, e compara seu valor com os rótulos dos cases.

A expressão entre parêntese deve ser de valor **inteiro ou char**.





# Estrutura Condicional de Múltipla Escolha

```
switch (expressão (int ou char))  
{  
    case valor1:  
        instruções;  
        break;  
    case valor2:  
        instruções;  
        break;  
    ...  
    case valorN:  
        instruções;  
        break;  
    default:  
        instruções;  
}
```

Cada **case** pode ser seguido por uma constante do tipo inteiro ou char.

Os rótulos dos cases devem ser todos diferentes

A linha do case deve sempre terminar com dois pontos (:)



# Estrutura Condicional de Múltipla Escolha

```
switch (expressão (int ou char))
```

```
{
```

```
case valor1:
```

```
    instruções;
```

```
    break;
```

```
case valor2:
```

```
    instruções;
```

```
    break;
```

```
...
```

```
case valorN:
```

```
    instruções;
```

```
    break;
```

```
default:
```

```
    instruções;
```

```
}
```

Pode haver uma ou mais instruções seguindo cada case. Estas instruções não precisam estar entre chaves.

No entanto, o corpo de um **switch** deve estar envolto por chaves.



# Estrutura Condicional de Múltipla Escolha

```
switch (expressão (int ou char))  
{  
    case valor1:  
        instruções;  
        break;  
    case valor2:  
        instruções;  
        break;  
    ...  
    case valorN:  
        instruções;  
        break;  
    default:  
        instruções;  
}
```

O comando **break** causa uma saída imediata do switch.

Se não existir um comando break seguindo as instruções de um case, o programa segue executando todas as instruções dos cases abaixo.



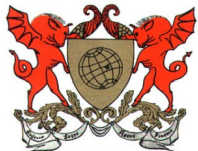
# Estrutura Condicional de Múltipla Escolha

```
switch (expressão (int ou char))  
{  
    case valor1:  
        instruções;  
        break;  
    case valor2:  
        instruções;  
        break;  
    ...  
    case valorN:  
        instruções;  
        break;  
    default:  
        instruções;  
}
```

Se um rótulo do **case** tiver o **valor igual** ao da expressão, a execução começa nele.

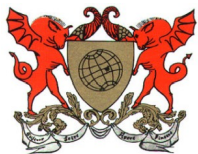
Se **nenhum case** for satisfeito e existir um **default**, a execução começara nele.

A instrução default é **opcional**.



# Estrutura Condicional de Múltipla Escolha

```
switch(num) {  
    case 1:  
        printf("Numero igual a 1\n");  
        break;  
    case 2:  
        printf("Numero igual a 2\n");  
        break;  
    case 3:  
        printf("Numero igual a 3\n");  
        break;  
    case 4:  
        printf("Numero igual a 4\n");  
        break;  
    case 5:  
        printf("Numero igual a 5\n");  
        break;  
    default:  
        printf("Numero invalido\n");  
}
```



# Estrutura Condicional de Múltipla Escolha

```
switch(num + 0){  
    case 1:  
        printf("Numero igual a 1\n");  
        break;  
    case 2:  
        printf("Numero igual a 2\n");  
        break;  
    case 3:  
        printf("Numero igual a 3\n");  
        break;  
    case 4:  
        printf("Numero igual a 4\n");  
        break;  
    case 5:  
        printf("Numero igual a 5\n");  
        break;  
    default:  
        printf("Numero invalido\n");  
}
```



# Estrutura Condicional de Múltipla Escolha

```
switch(letra) {  
    case 'A':  
        printf("Letra igual a A\n");  
        break;  
    case 'B':  
        printf("Letra igual a B\n");  
        break;  
    case 'C':  
        printf("Letra igual a C\n");  
        break;  
    case 'D':  
        printf("Letra igual a D\n");  
        break;  
    case 'E':  
        printf("Letra igual a E\n");  
        break;  
    default:  
        printf("Letra invalida\n");  
}
```

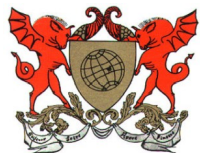


# Exemplos

Faça um programa que leia do teclado os valores reais  $a$ ,  $b$  e  $c$  de uma equação do 2º grau  $ax^2 + bx + c = 0$  e depois calcule e escreva na tela as **raízes** da equação.

Sujeito a algumas condições:

- se  $a = 0$ , escreva que não é equação do 2º grau.
- $\Delta < 0$ , a equação não possui raízes reais.
- $\Delta = 0$ , a equação possui uma única raiz real.
- $\Delta > 0$ , a equação possui duas raízes reais.



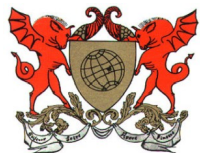


# Revisão

O comando **if** oferece um meio de escolha entre executar ou não um bloco de instruções.

O comando **if-else** oferece um meio de executar um ou outro bloco de instruções.

O **operador condicional ternário** opera sobre três expressões. A primeira é avaliada e, se verdadeira, a expressão toda assume o valor da segunda expressão; caso contrário, assume o valor da terceira expressão.



# Revisão

O comando **switch** oferece um meio de escolher uma entre muitas opções.

A expressão do comando **switch** deve ter um valor **inteiro ou char**. E as expressões de cada **caso** devem ter um valor inteiro ou char constante e devem ser todas diferentes;

No **switch** o controle do programa pula para o caso em o valor for igual ou, se não houver nenhum caso igual, para o **default**. O caso default é opcional;

O comando **break** provoca uma saída imediata da construção switch.



# Exercícios

1) Determine se um determinado ano lido é bissexto. Sendo que um ano é bissexto se for divisível por 400 ou for divisível por 4 e não for divisível por 100. Por exemplo: 1988, 1992, 1996.

2) Faça um programa que receba a altura e o sexo de uma pessoa e calcule e mostre seu peso ideal, utilizando as seguintes fórmulas (onde h corresponde à altura):

Homens:  $(72.7 * h) - 58$

Mulheres:  $(62,1 * h) - 44,7$

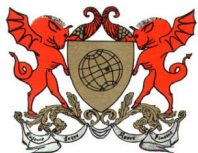


# Exercícios

3) O Índice de Massa Corporal (IMC) é um método aproximado para se medir o grau de obesidade de uma pessoa. É calculado como sendo a massa da pessoa (em kg) dividido pelo quadrado da altura da mesma (em metros).

Faça um programa que peça ao usuário seus dados de peso e altura, calcule seu IMC e exiba em tela o resultado, junto com a classificação de acordo com a tabela a seguir.

IMC	Classificação
< 16	Magreza grave
16 a < 17	Magreza moderada
17 a < 18,5	Magreza leve
18,5 a < 25	Saudável
25 a < 30	Sobrepeso
30 a < 35	Obesidade Grau I
35 a < 40	Obesidade Grau II (severa)
≥ 40	Obesidade Grau III (mórbida)



# Exercícios

4) Bafômetro (ou etilômetro): aparelho que permite determinar a concentração de bebida alcoólica em uma pessoa, analisando o ar exalado dos pulmões.

Taxa de alcoolemia: quantidade de álcool existente no sangue de um indivíduo, em determinado momento, expressa em gramas de álcool por litro de sangue (g/l). Ex.: uma alcoolemia de 0,5 g/l é o mesmo que dizer que existem 0,5 g de álcool por litro de sangue.



# Exercícios

Alcoolemia (valor aproximado):

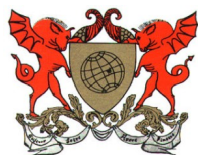
Taxa de alcoolemia =  $\frac{\text{Álcool consumido (g)}}{(\text{Peso corporal (kg)} \times \text{coeficiente})}$

- 0,7 nos homens em jejum
- 0,6 nas mulheres em jejum
- 1,1 durante as refeições (ambos os sexos)



Limite (Lei Seca): 0,05 g/l

(em outras palavras, se for dirigir não beba ou se beber não dirija)



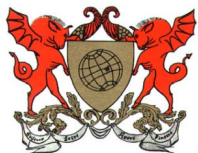
# Exercícios

Quadro 1: Unidades de álcool em cada dose de bebida

Bebida	Volume	Teor Alcoólico	Quantidade de Álcool (Volume x Teor Alcoólico)	Gramas de Álcool (Volume de Álcool x 0,8*)	Dose 1D = 14g
Vinho Tinto	150ml	12%	18ml	14,4g	1
1 lata de Cerveja	350ml	5%	17,5ml	14g	1
Destilado	40ml	40%	16ml	12,8g	1

(\*) A quantidade de álcool em gramas é obtida a partir da multiplicação do volume de álcool contido na bebida pela densidade do álcool ( $d=0,8$ ).

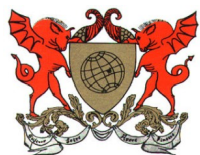
Desenvolva um programa que permita calcular e exibir na tela a taxa de alcoolemia do sangue de pessoas de acordo com o **número de copos de cerveja** ingeridos por uma pessoa.



# Exercícios

Etanol no sangue (gramas/litro)	Estágio	Sintomas
0,1 a 0,5	Sobriedade	Nenhuma influência aparente.
0,3 a 1,2	Euforia	Perda de eficiência, diminuição da atenção, julgamento e controle
0,9 a 2,5	Excitação	Instabilidade das emoções, incoordenação muscular. Menor inibição. Perda do julgamento crítico
1,8 a 3,0	Confusão	Vertigens, desequilíbrio, dificuldade na fala e distúrbios da sensação.
2,7 a 4,0	Estupor	Apatia e inércia geral. Vômitos, incontinência urinária e fezes.
3,5 a 5,0	Coma	Inconsciência, anestesia. Morte
Acima de 5	Morte	Parada respiratória

Poderíamos até fazer uma classificação quanto à quantidade de álcool no sangue.





# Exercícios

5) O número 3025 tem uma propriedade interessante:

$$30 + 25 = 55, \text{ e } 55^2 = 3025.$$

Faça um programa para ler um número de 4 algarismos e dizer se ele possui essa propriedade ou não.

Qual o número? **3025**

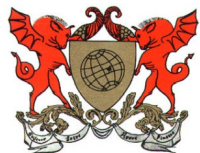
3025 possui a propriedade.

Qual o número? **3125**

3125 não possui a propriedade.

Qual o número? **9801**

9801 possui a propriedade.



# Exercícios

6) Ler as notas de um aluno nas 3 provas do semestre. Ler também o número de aulas dadas e o número de faltas do aluno. Calcular sua nota final e sua porcentagem de falta. Dizer se ele foi aprovado ou não. O aluno é aprovado se sua nota total for 60 ou mais, e se não tiver faltado mais que 25% das aulas. Veja alguns exemplos.

Notas: 20 30 30  
Numero de aulas: 30  
Numero de faltas: 4  
Nota final: 80  
Faltas: 13.3333%  
Você foi aprovado.

Notas: 20 20 25  
Numero de aulas: 60  
Numero de faltas: 17  
Nota final: 65  
Faltas: 28.3333%  
Você foi reprovado.



# Exercícios

7) Usando **switch**, Escreva um programa que, dada a idade de um nadador, classifique-o em uma das seguintes categorias:

Categoria	Idade
Infantil A	5 a 7
Infantil B	8 a 10
Juvenil A	11 a 13
Juvenil B	14 a 17
Sênior	maiores de 18 anos

