



Disciplina:

Programação Computacional*

Prof. Fernando Rodrigues
e-mail: fernandorodrigues@sobral.ufc.br



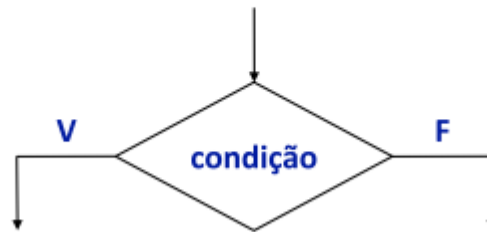
Aula 09: Programação em C

- ❖ Operadores relacionais
- ❖ Estruturas de seleção
- ❖ Estruturas de repetição
- ❖ Desvio de fluxo

Estruturas de seleção

► Execução condicional

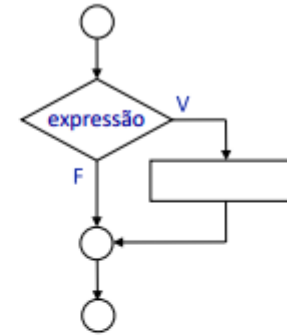
Etapa em um programa onde uma condição é avaliada e, dependendo do resultado, diferentes ações podem ser realizadas.



Estruturas de seleção

► Condição simples

Uma ou mais ações serão executadas se, e apenas se, uma determinada expressão for verdadeira.



Números pares

Leia um número inteiro e, se o mesmo for divisível por 2, exiba a mensagem “O número digitado é par.”

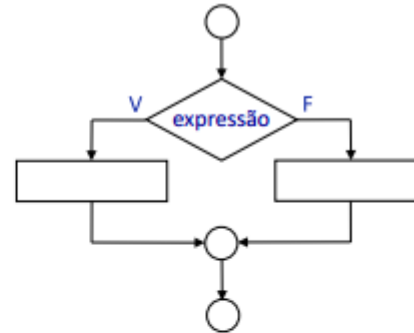
∞

Leia os valores de A e B e, se B for diferente de zero, calcule a divisão de A por B.

Estruturas de seleção

► Condição composta

Se uma determinada expressão for verdadeira, **algumas** ações serão executadas. Se a expressão for falsa, **outras** ações serão executadas.



Números pares e ímpares

Leia um número inteiro e, se o mesmo for divisível por 2, exiba a mensagem “O número informado é par.” Caso contrário, exiba a mensagem “O número informado é ímpar”.

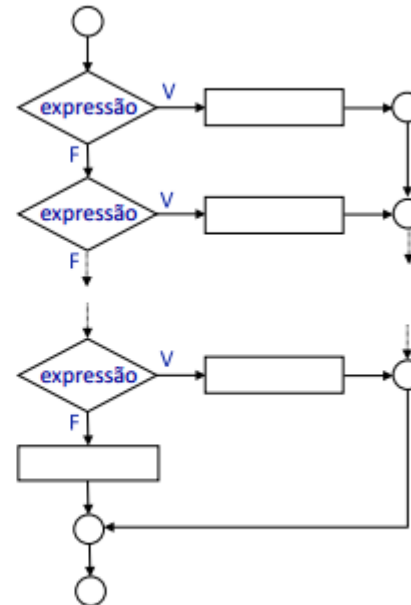
∞ (versão *Tropa de Elite*)

Leia os valores de A e B e, se B for diferente de zero, calcule a divisão de A por B. Caso contrário, exiba a mensagem “Pede prá sair!”

Estruturas de seleção

► Condição múltipla

A condição avaliada permite mais de 2 opções.



Expressões condicionais

- Estabelecem uma comparação entre os dados (constantes ou variáveis) de um programa.
- O resultado dessa comparação é interpretado pela linguagem como um **valor lógico booleano** (verdadeiro ou falso).
- As expressões condicionais podem envolver operadores aritméticos, relacionais e lógicos.

Operadores relacionais

Estabelecem relações entre os dados de um programa.

Operador relacional	Significado	Precedência
==	igual	baixa
!=	diferente	baixa
<	menor	alta
<=	menor ou igual	alta
>	maior	alta
>=	maior ou igual	alta

Operadores relacionais tem precedência mais baixa que os operadores aritméticos.

a é menor do que b

a < b

a é igual a zero

a == 0

a é maior do que a soma de b e c

a > b + c

a é diferente de zero

a != 0

Verdadeiro ou falso em C

Expressão relacional

Expressão envolvendo dados e/ou operadores aritméticos, e operadores relacionais.

A avaliação de uma expressão relacional produz um valor booleano (verdadeiro ou falso), que é convertido para um valor numérico:

- 0, se a expressão for avaliada como falsa.
- 1, se a expressão for avaliada como verdadeira.

```
5 == 5.0
```

1, pois não há diferença entre variáveis de tipos numéricos.

```
5 == '5'
```

0, pois o valor decimal do caractere '5' é 53.

Verdadeiro ou falso em C

Expressão lógica

Expressão envolvendo dados e/ou operadores aritméticos e/ou operadores relacionais e/ou operadores lógicos.

A avaliação de uma expressão lógica produz um valor booleano (verdadeiro ou falso), que é convertido para um valor numérico:

- 0, se a expressão for avaliada como falsa.
- 1, se a expressão for avaliada como verdadeira.

Generalizando...

- Qualquer valor ou expressão diferente de zero será considerada como verdadeira.
- Qualquer valor ou expressão igual à zero será considerada como falsa.

Estrutura condicional(seleção) simples

if ()

Estrutura utilizada para avaliar uma condição simples.

A sintaxe da estrutura `if ()` é:

```
if (expressão)
{
    // comandos a serem executados se a expressão for verdadeira.
}
```

Atenção

Não se deve usar “;” após o `if ()`.

Bloco de código

As instruções que deverão ser executadas, no caso da **expressão** da estrutura **if ()** ser avaliada como verdadeira, deverão ser escritas dentro de um **bloco** delimitado pelos símbolos { e }.

Dica importante

Para facilitar a visualização, os comandos dentro de um bloco também devem ser recuados.

A omissão dos símbolos { e } só será permitida se o bloco de instruções se constituir de apenas 1 comando.

No caso da **expressão** da estrutura **if ()** ser considerada falsa, todo o bloco de comandos será ignorado e o programa prossegue a partir do primeiro comando após o bloco.

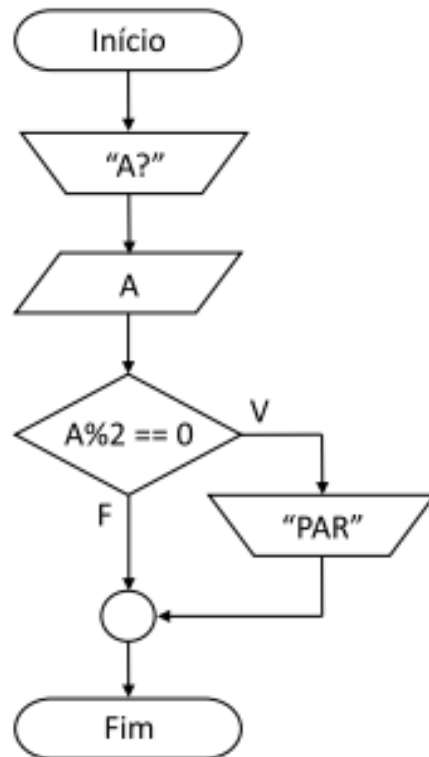
Considere o seguinte trecho de código:

```
aprovado = 0;
if (media >= 7.0)
{
    aprovado = 1;
    puts("PARABÉNS!!!");
}

if (aprovado == 0)
    puts("Você foi reprovado!");
```

E se os delimitadores de bloco do primeiro **if ()** forem removidos?

Exemplo: Ler um número inteiro e informar se o mesmo é par.



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

main()
{
    int a;

    printf("Digite um número inteiro: ");
    scanf("%d", &a);
    if (a % 2 == 0)
    {
        printf("O numero digitado é par.\n");
    }
    system("PAUSE");
}
```

Exemplo: Ler 3 valores numéricos distintos e exibir o maior deles.

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

int main()
{
    float a, b, c;
    float maior;

    printf("Digite 3 números: ");
    scanf("%f %f %f", &a, &b, &c);
    if ( a > b && a > c ) { maior = a; }
    if ( b > a && b > c ) { maior = b; }
    if ( c > a && c > b ) { maior = c; }
    printf("O maior numero digitado é %f\n", maior);
    system("PAUSE");
    return(0);
}
```

Exemplo: Verificar se um número real x pertence ao intervalo $[-1, 1]$.

Versão 1: testa se está dentro.

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

int main()
{
    float x;

    printf("Digite um número: ");
    scanf("%f", &x);
    if (x >= -1 && x <= 1)
    {
        printf("Pertence.\n");
    }
    system("PAUSE");
    return(0);
}
```

Estrutura condicional composta (if()-else)

if()-else

Estrutura utilizada para avaliar uma condição composta.

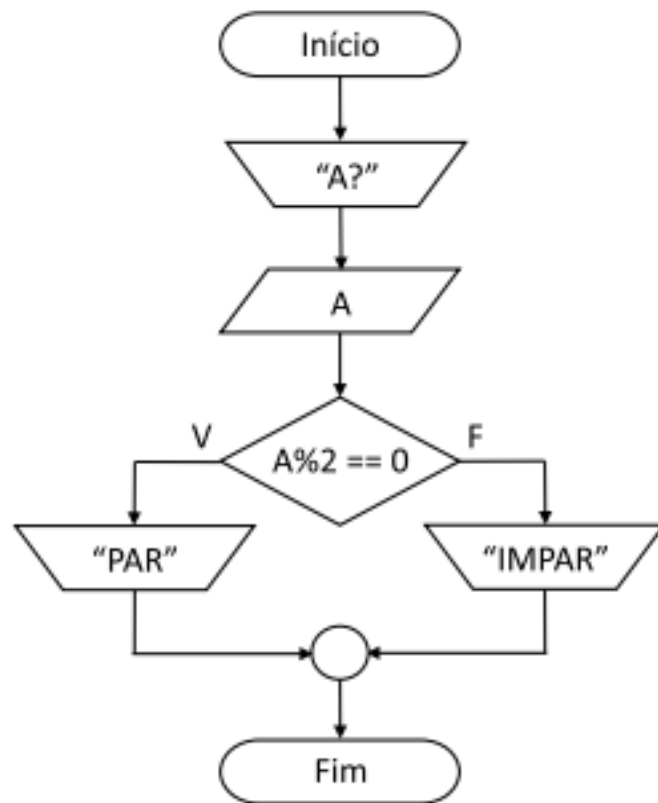
A sintaxe da estrutura **if ()-else** é:

```
if (expressão)
{
    // comandos a serem executados se a expressão for verdadeira.
}
else
{
    // comandos a serem executados se a expressão for falsa.
}
```

Atenção

Não se deve usar ";" após o **if ()**, nem após o **else**.

Exemplo: Ler um número inteiro e informar se o mesmo é par ou ímpar.



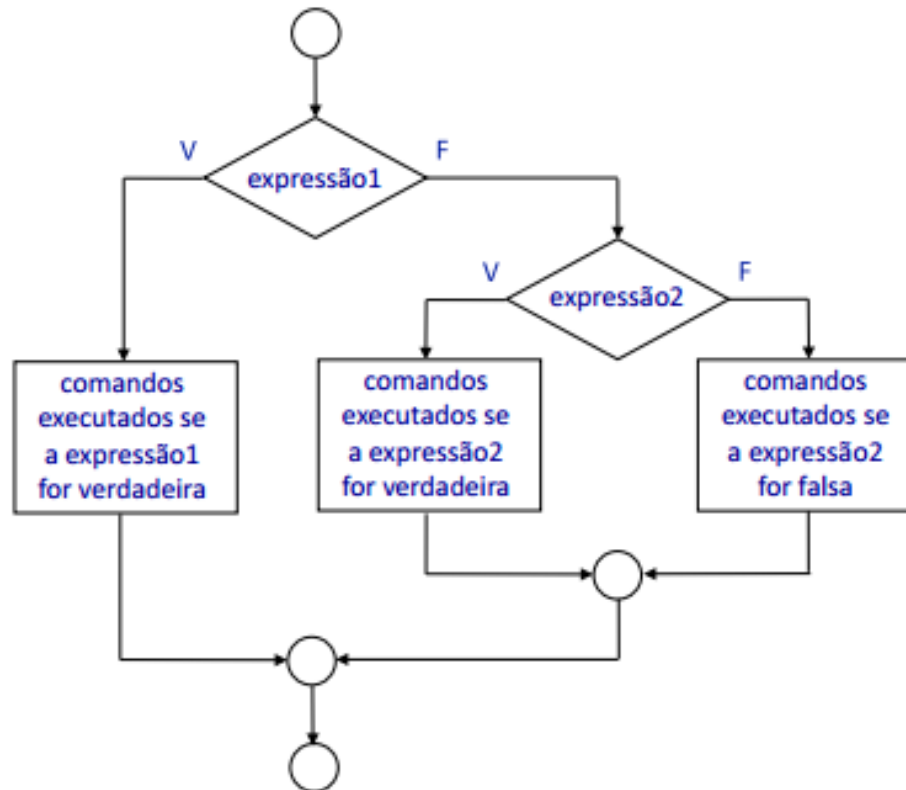
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a;

    printf("Digite um número inteiro: ");
    scanf("%d", &a);
    if (a % 2 == 0)
    {
        printf("O numero digitado é par.\n");
    }
    else
    {
        printf("O numero digitado é impar.\n");
    }
    system("PAUSE");
    return(0);
}
```

Aninhamento

Técnica de seqüenciar os testes de modo a melhorar o desempenho do programa.



Aninhamento

Se a média M for maior ou igual a 7.0, não há necessidade de realizar outros testes.

```
if ( M >= 7.0 )
    printf("Aprovado.\n");
else
    if ( M >= 5.0 && M < 7.0 )
        printf("Realizar Prova 5.\n");
    else
        printf("Reprovado.\n");
```

Atenção

Uma cláusula **else** sempre pertence ao **if** imediatamente anterior. Caso isso não for desejável, deve-se utilizar os delimitadores de bloco { e }.

A estrutura condicional múltipla switch()

O aninhamento de várias estruturas compostas **if()**-**else** (ou **else-if**) permite o tratamento de expressões condicionais que permitem mais de duas alternativas.

Se a expressão resultar em um valor do tipo **int** ou **char**, é possível utilizar a estrutura condicional múltipla **switch()**.

```
switch(expressão)
{
    case constante1:
        //comandos1
        break;

    case constante2:
        //comandos2
        break;

    ...

    case constanteN:
        //comandosN
        break;

    default:
        //comandos
}
```

A estrutura condicional múltipla switch()

1. A expressão é avaliada.
2. O fluxo de execução é desviado para o rótulo **case** cujo valor equivale ao da expressão. Serão executados todos os comandos até que seja encontrado o comando **break**.
3. A execução do comando **break** desvia o fluxo de execução para o primeiro comando **após** a estrutura **switch()**.
4. Se nenhum **case** corresponde ao valor da expressão, então o fluxo de execução será desviado para o rótulo **default** (opcional).

Cuidado!

A omissão do comando **break** causará a execução dos **case's** subsequentes.



A estrutura condicional múltipla switch() - Exemplo:

```
#include <stdio.h>

main()
{
    float num1, num2, result;
    char oper;
    int valido = 1;
    printf("Digite a operacao a ser executada de forma infixada: (operando real_1 sinal da operacao operando real_2)\n");
    scanf("%f%c%f",&num1,&oper,&num2);
    fflush(stdin);
    switch(oper){
        case '+':
            result = num1+num2;
            break;
        case '-':
            result = num1-num2;
            break;
        case '*':
            result = num1*num2;
            break;
        case '/':
            result = num1/num2;
            break;
        default:
            printf("Operador não definido!\n");
            valido = 0;
    }
    if(valido){
        printf("O resultado de %.2f %c %.2f eh %f\n",num1,oper,num2,result);
    }
}
```

```
Digite a operacao a ser executada de forma infixada: (operando real_1 sinal da operacao operando real_2)
2.5*4
O resultado de 2.50 * 4.00 eh 10.000000
```