

Computação 1

Estruturas de Seleção

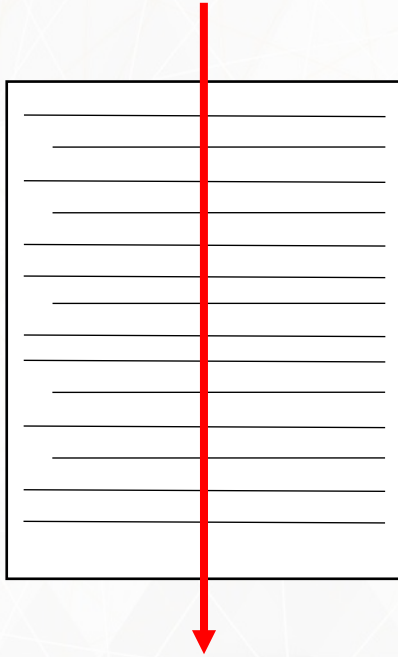
Prof. Luiz Fernando Carvalho

luizfcarvalho@utfpr.edu.br

Execução de Instruções

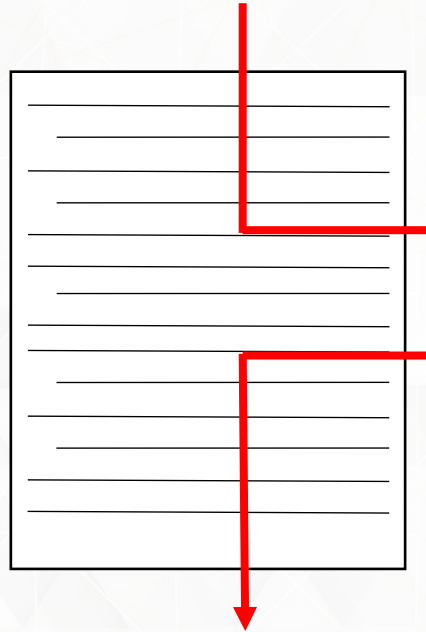
- Programa de computador
 - Conjunto de instruções organizadas de forma a produzir a solução de um determinado problema.
- Fluxo de execução
 - Começa na primeira linha e avança sequencialmente
 - de cima para baixo;
 - Em muitas circunstâncias é necessário executar instruções em uma ordem diferente;
 - Necessidade de decisão entre fluxos alternativos de execução ou da repetição de determinadas instruções
 - Pode haver bifurcações, repetição de código e tomada de decisão

Execução de Instruções



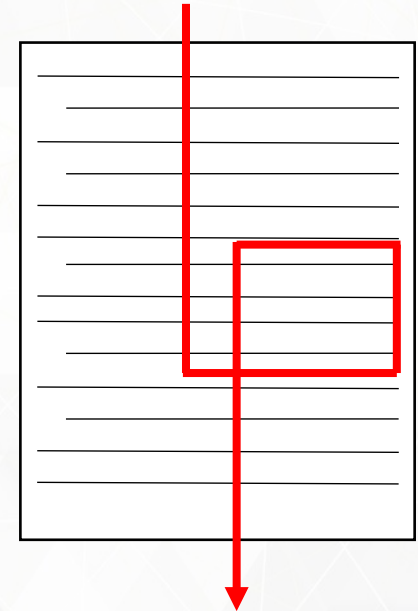
Fluxo de execução
Sequencial

Comandos são executados um após o outro



Fluxo de execução
com desvio

Comandos são executados dependendo do valor de uma condição



Fluxo de execução
repetitivo

Comandos são executados de forma repetida

Estrutura de Controle

- As estruturas de controle dividem-se em:
 - Estruturas de decisão/seleção (condicionais);
 - Estruturas de repetição (loops de repetição).
- As estruturas de controle estão vinculadas às condições que determinam se instruções serão ou não executadas;
- Uma condição de controle está relacionada aos operadores relacionais e lógicos;

Estrutura de Seleção

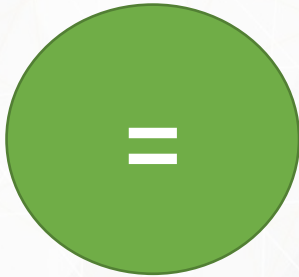
- Uma estrutura de decisão permite decidir se um conjunto de instruções **será** ou **não** executado de acordo com determinadas condições;
- A decisão é feita com base no resultado de um teste lógico que determina a condição;
- Estruturas de decisão em C/C++:
 - `if`;
 - `if-else`;
 - `switch`;

Operadores Relacionais

- Os operadores relacionais comparam dois valores e retornam um valor booleano
 - Verdadeiro (true);
 - Falso (false);

Operador	Descrição	X	Y	Lógico	Resultado
==	Igual a	2	3	X == Y	Falso
!=	Diferente de	2	3	X != Y	Verdadeiro
>	Maior que	2	3	X > Y	Falso
>=	Maior ou Igual	2	3	X >= Y	Falso
<	Menor que	2	3	X < Y	Verdadeiro
<=	Menor ou igual	2	3	X <= Y	Verdadeiro

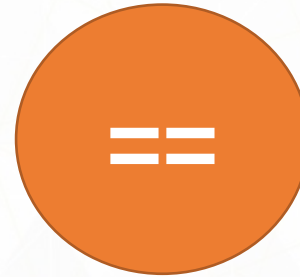
Operadores Relacionais



Operador de Atribuição

`x = 5`

x recebe o valor 5



Operador Relacional

`x == 5`

O valor armazenado
em x é igual a 5?

É um erro de programação utilizar “=” quando se deve utilizar “==” (e vice-versa)

Declaração IF

Estrutura condicional simples

```
//comandos
```

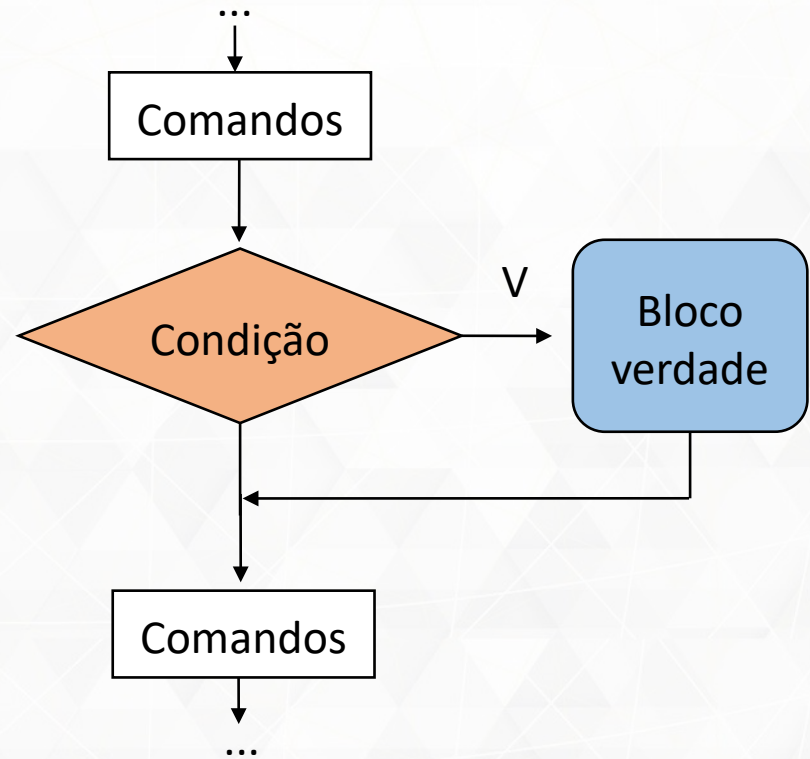
```
if(condição)
```

```
{
```

```
    //bloco verdade
```

```
}
```

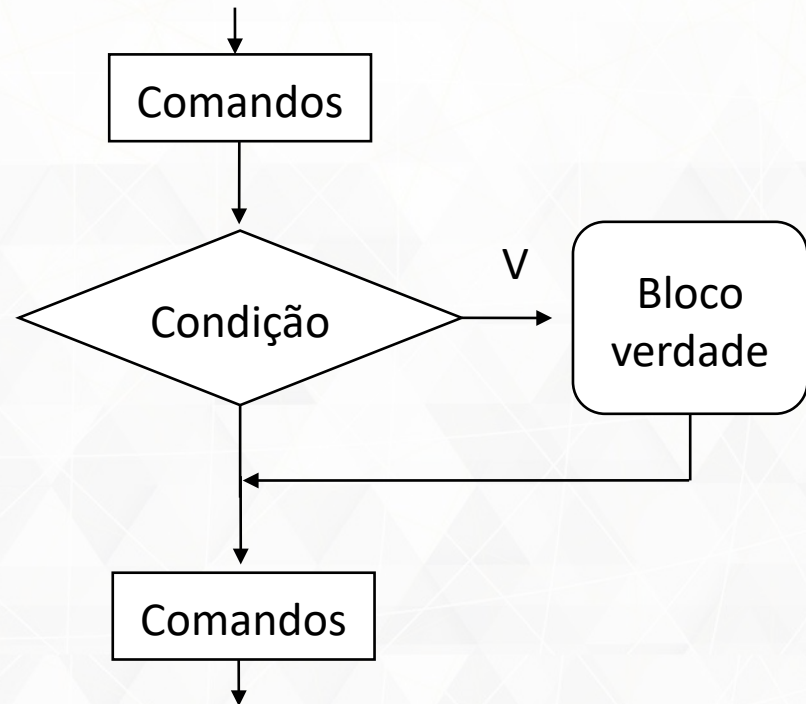
```
//comandos
```



Declaração IF

Estrutura condicional Simples

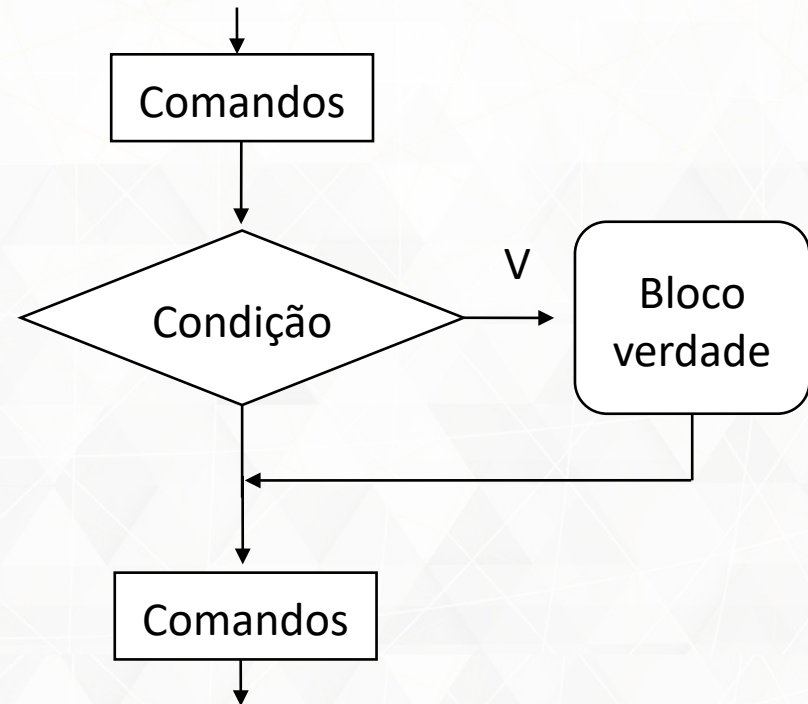
```
int main(){  
    float nota = 7.6;  
    if(nota >= 7.0)  
        printf("APROVADO!");  
    ...  
    ...  
    return 0;  
}
```



Declaração IF

Estrutura condicional Simples

```
int main(){  
  
    float nota1 = 7.6;  
    float nota2 = 5.0;  
    float media;  
  
    if(nota1 == 10.0)  
        printf("PARABÉNS!");  
  
    media = (nota1 + nota2)/2;  
    printf("Sua media e': %f", media);  
    ...  
    ...  
  
    return 0;  
}
```



Declaração IF-ELSE

```
//comandos
```

```
if(condição)
```

```
{
```

```
    //bloco verdade
```

```
}
```

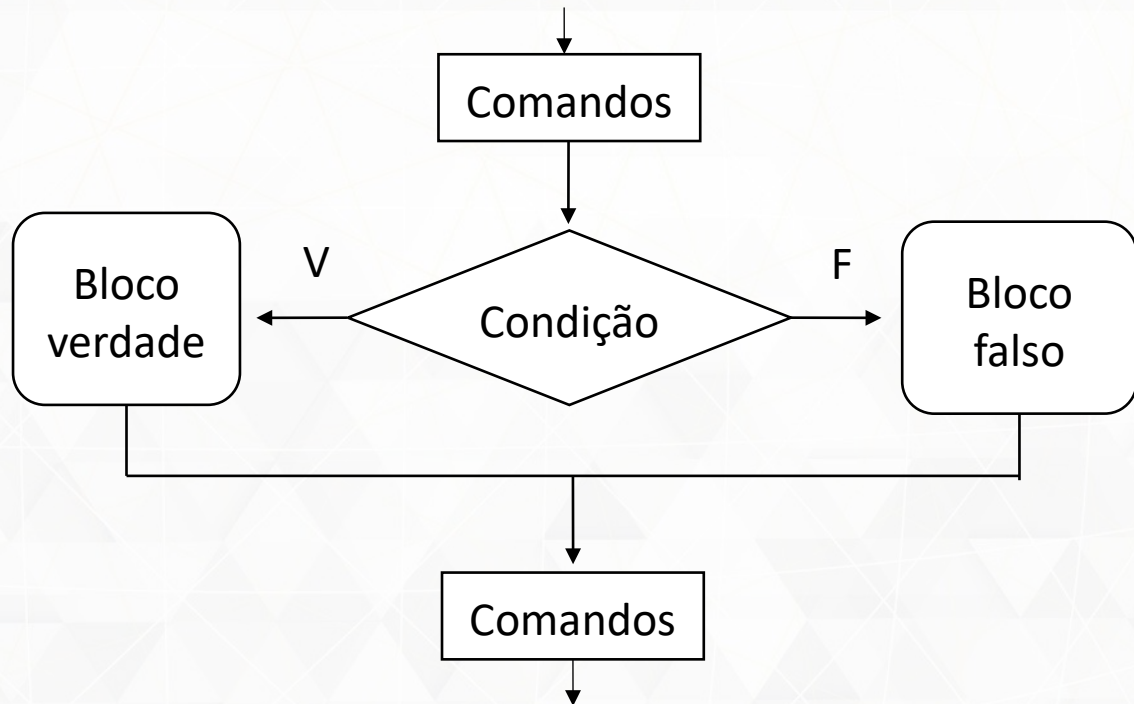
```
else
```

```
{
```

```
    //bloco falso
```

```
}
```

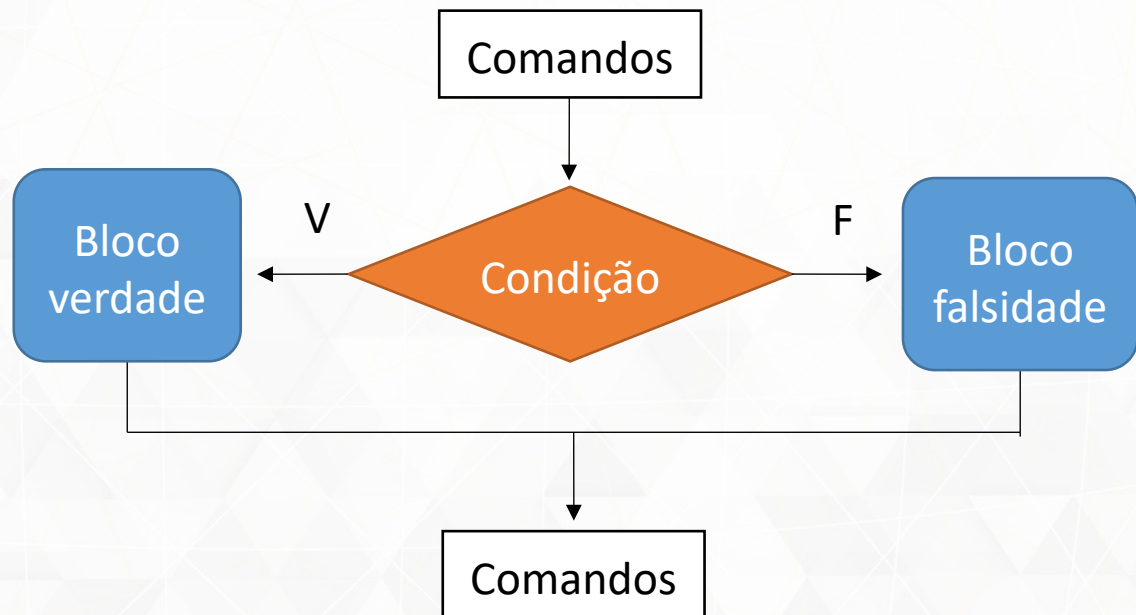
```
//comandos
```



A declaração *else* é opcional: pode-se utilizá-la para determinar um conjunto de comandos que serão executados caso a condição testada seja falsa

Declaração IF-ELSE

```
int a = 8;  
  
if(a == 5)  
    printf("a vale 5");  
else  
    printf("a não vale 5");  
  
a = 3;
```



Somente um dos conjuntos de comandos (*if* ou *else*) será executado, nunca os dois!

Blocos de Instrução

- Com a adição de estruturas de controle segue um novo conceito: bloco de instruções;
- **Bloco:** Conjunto de instruções agrupadas e delimitadas por chaves `{ }`
- Se o bloco for de apenas uma instrução, pode-se omitir as `{ }`
- Exemplos:

```
if(x > 100)
    printf("x maior que 100");
```

```
if(x > 100)
{
    printf("x maior que 100\n");
    printf("x vale: %d", x);
}
else
    printf("x não maior que 100");
```

Blocos de instruções – Erros comuns I

```
int main(){  
    float saldo = 150.0;  
    float saque = 200.0;  
  
    if(saldo - saque >= 0)  
        saldo = saldo - saque;  
        printf("Saque realizado com sucesso. Saldo atual = %f", saldo);  
    else  
        printf("Impossivel realizar o saque. Saldo insuficiente");  
  
    return 0;  
}
```

Onde está o erro?

O compilador acusará que existe um ELSE sem um IF, porque?

Blocos de instruções – Erros comuns II

```
int main(){  
    float saldo = 150.0;  
    float saque = 50.0;  
  
    if(saldo - saque >= 0){  
        saldo = saldo - saque;  
        printf("Saque realizado com sucesso. Saldo atual = %f", saldo);  
    }  
    else  
        printf("Impossivel realizar o saque.");  
    printf("Informe um valor menor ou igual a %f", saldo);  
  
    return 0;  
}
```

Onde está o erro?

Estrutura de seleção aninhadas

- É possível também aninhar comandos **if**, ou seja, fazer uma declaração **if** dentro de outra declaração **if** anterior.
- **Exemplo:** examinar se um número é positivo. Em caso afirmativo, verificar se o mesmo é divisível por 2.

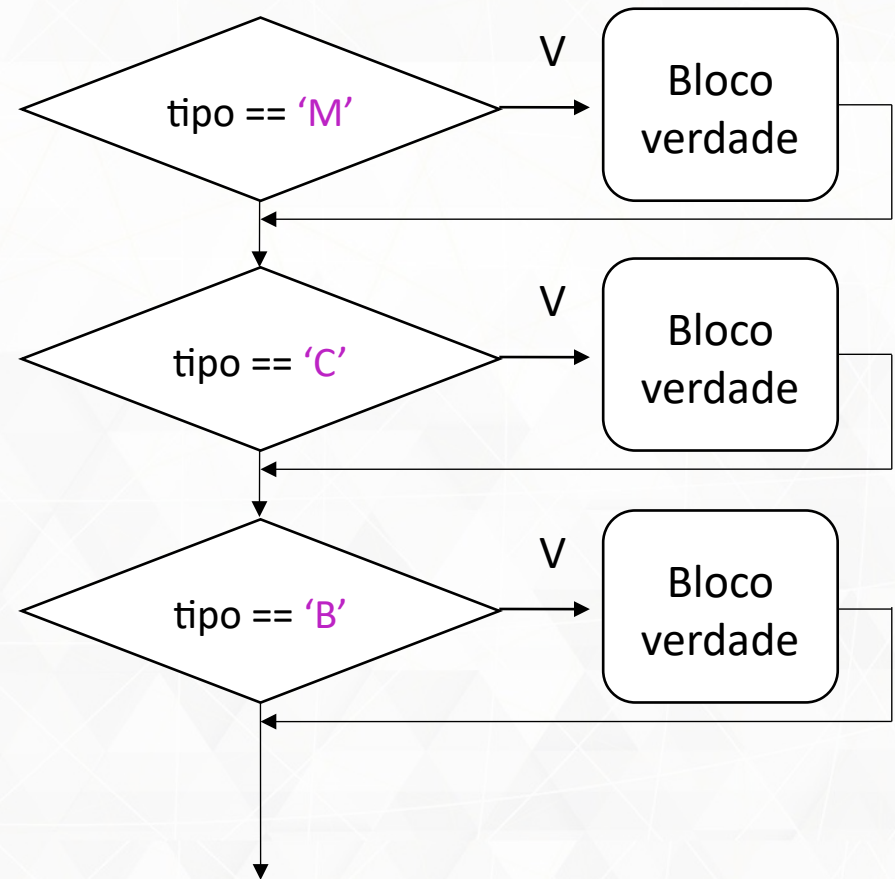
←
indentação

```
if(num >= 0) {  
    if(num % 2 == 0) //resto da divisão por 2  
        printf("O numero é par e positivo");  
    else  
        printf("O numero é impar e positivo");  
}  
else  
    printf("O numero é negativo");
```

Um **if** aninhado é um comando **if** que está dentro de um outro comando **if**.

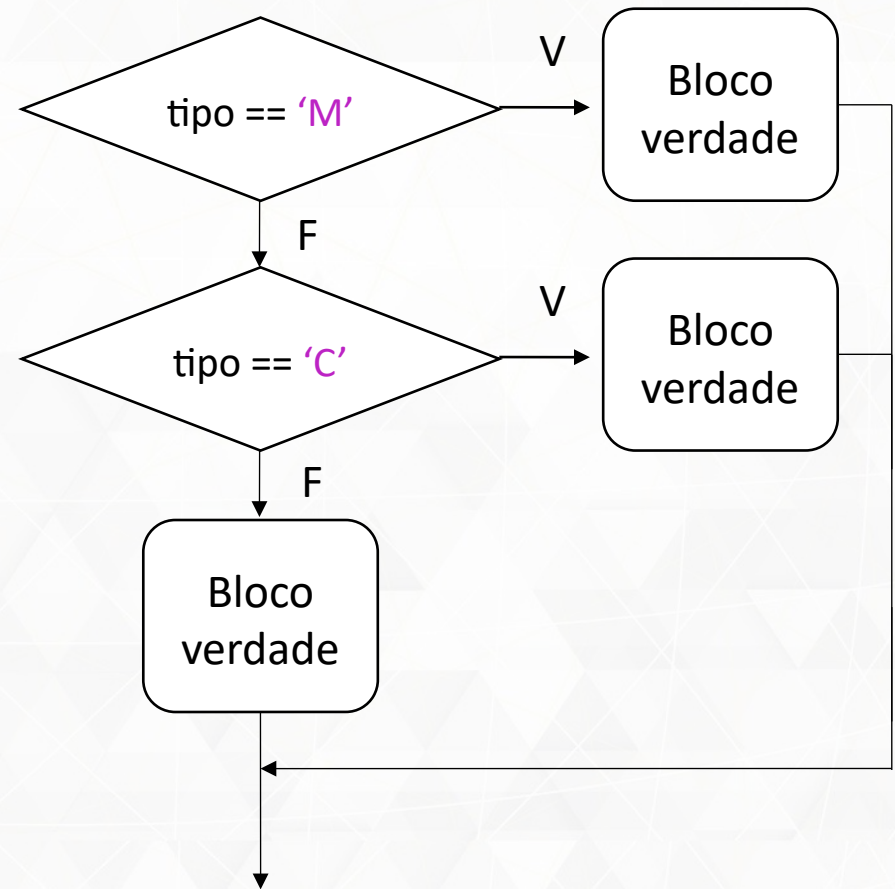
Estrutura de seleção composta

```
int main(){  
    char tipo;  
    scanf("%c", &tipo);  
  
    if(tipo == 'M')  
        printf("Mucarela");  
    if(tipo == 'C')  
        printf("Calabresa");  
    if(tipo == 'B')  
        printf("Bacon");  
  
    return 0;  
}
```



Estrutura de seleção composta

```
int main(){  
    char tipo;  
    scanf("%c", &tipo);  
  
    if(tipo == 'M')  
        printf("Mucarela");  
    else if(tipo == 'C')  
        printf("Calabresa");  
    else  
        printf("Bacon");  
  
    return 0;  
}
```



Declaração IF-ELSE-IF

```
int num = 10;

if(num < 100)
    printf("Menor que 100");
if(num < 1000)
    printf("Menor que 1000");
if(num < 10000)
    printf("Menor que 10000");
if(num >= 10000)
    printf("Maior ou igual a 10000");
```

```
int num = 10;

if(num < 100)
    printf("Menor que 100");
else if(num < 1000)
    printf("Menor que 1000");
else if(num < 10000)
    printf("Menor que 10000");
else
    printf("Maior ou igual a 10000");
```

```
int num = 10;

if(num < 100)
    printf("Menor que 100");
else
    if(num < 1000)
        printf("Menor que 1000");
    else
        if(num < 10000)
            printf("Menor que 10000");
        else
            printf("Maior ou igual a 10000");
```

- As condições são avaliadas de cima para baixo;
- Assim que uma condição verdadeira é encontrada, o comando associado a ela é executado;
- O restante das condições não são executadas;
- Se nenhuma das condições for verdadeira, o último **else** é executado.

Operadores Lógicos

- Trabalham com valores booleanos e seu resultado também é booleano (**verdadeiro** ou **falso**);
- Eles são usados somente em expressões lógicas;
- Usados para combinar condições simples, criando condições complexas

Operador	Significado	Equivalente
!	Negação	Não
&&	Conjunção	E
	Disjunção	OU

Operadores Lógicos

- Sejam A e B duas variáveis booleanas que assumem valores:
 - Verdadeiro (V)
 - Falso (F)

A	B	!B	A && B	A B
V	V	F	V	V
V	F	V	F	V
F	V	F	F	V
F	F	V	F	F

Operadores lógicos

- Se quisermos saber se um número num é positivo e par

```
if(num >= 0){  
    if(num % 2 == 0)  
        printf("Numero par e positivo");  
}
```

Combinando as duas condições em apenas uma...

```
if(num >= 0 && num % 2 == 0)  
    printf("Numero par e positivo");
```

Exercícios I

- 1) Faça um programa que leia três números quaisquer e imprima o maior deles;
- 2) Faça um programa que receba duas notas de um aluno. Calcule e mostre a média aritmética das notas e uma mensagem conforme a tabela a seguir

Média	Mensagem
[0, 4)	Reprovado
[4, 6)	Exame
[6, 10]	Aprovado

Exercícios II

3) Desenvolver a lógica para um programa que efetue o cálculo do reajuste de salário de um funcionário

- Salario \leq 500, reajuste de 15%
- Salario $>$ 500, mas Salario \leq 1000, reajuste será de 10%
- Salario $>$ 1000, reajuste será de 5%

Ao final do programa, mostrar o valor do salário somado ao reajuste.

4) Faça um programa que recebe a idade de um nadador e classifique-o numa das seguintes categorias:

- Adulto (idade ≥ 18);
- Juvenil (idade ≥ 14 e idade < 18);
- Infantil (idade ≥ 9 e idade < 14);
- Mirim (Idade < 9).

Declaração SWITCH

- Construções **if-else** facilitam a escrita de programas que devem escolher uma entre duas alternativas.
- Algumas vezes, entretanto, o programa necessita escolher uma entre várias alternativas.
- Embora **if-else** possa ser alinhado para executar vários testes condicionais, elas podem tornar o código muito complexo.
- Uma solução a ser considerada é utilizar uma construção para teste multi-condicional chamado **switch**
- Porém, só opera com tipos de variável **char** ou **int**

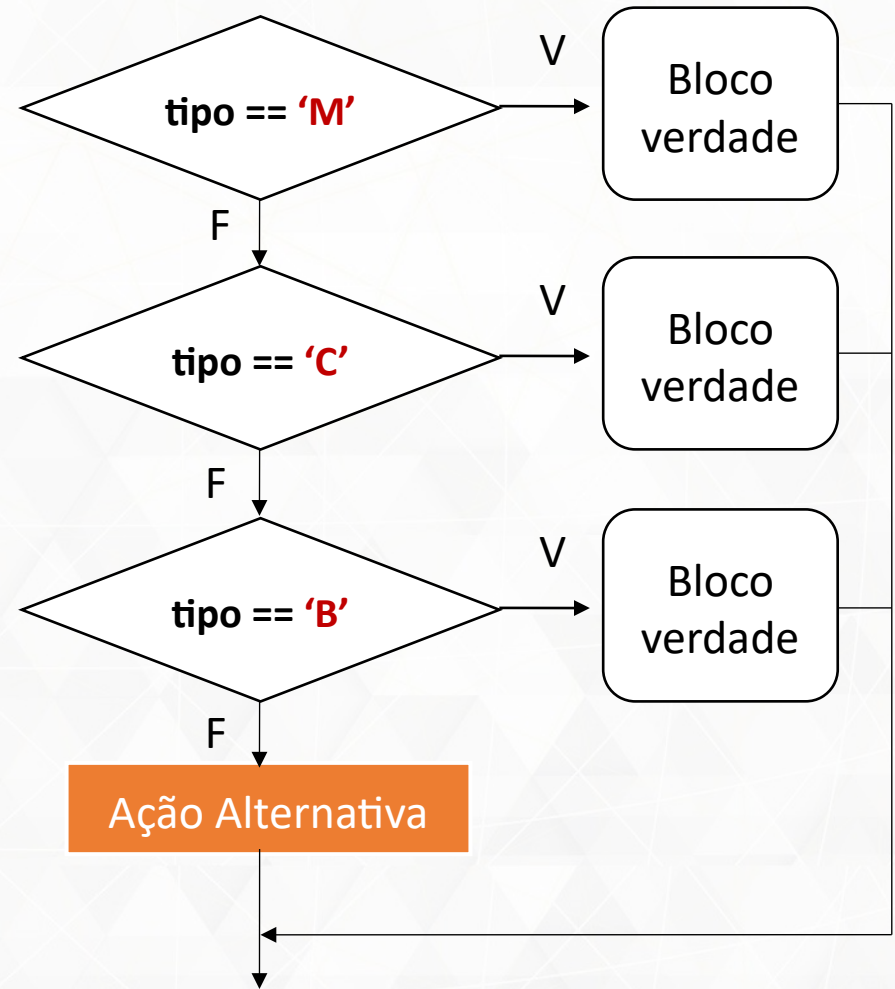
Declaração SWITCH

```
switch(variável do tipo char ou tipo int){  
    case <valor possível da variável>:  
        ...comandos...  
        break;  
    case <valor possível da variável>:  
        ...comandos...  
        break;  
    case <valor possível da variável>:  
        ...comandos...  
        break;  
    default:  
        ...comandos se nenhum case for atendido...  
        break;  
}
```

- **break** indica o término das instruções de um **case**
- **default** é utilizado se nenhum dos **cases** for atendido
- comando **switch** só aceita variáveis do tipo **int** e **char**.

Declaração SWITCH

```
1  int main()  
2  {  
3      char tipo;  
4      scanf("%c", &tipo);  
5  
6      switch(tipo)  
7      {  
8          case 'M':  
9              printf("Mucarela");  
10             break;  
11          case 'C':  
12              printf("Calabresa");  
13              break;  
14          case 'B':  
15              printf("Bacon");  
16              break;  
17          default:  
18              printf("opcao invalida");  
19              break;  
20      }  
21      return 0;  
22  }
```



Declaração SWITCH

```
1  int main()  
2  {  
3      int opcao;  
4      printf("Digite uma opção (1, 2 ou 3)\n");  
5      scanf("%i", &opcao);  
6  
7      switch(opcao)  
8      {  
9          case 1:  
10             printf("Escolheu a opção 1");  
11             break;  
12          case 2:  
13             printf("Escolheu a opção 2");  
14             break;  
15          case 3:  
16             printf("Escolheu a opção 3");  
17             break;  
18          default:  
19             printf("opção invalida");  
20             break;  
21      }  
22      return 0;  
23  }
```

Exercícios III

5) Crie um programa onde:

- O usuário deve fornecer um valor;
- O programa deve responder com o nome do dia da semana correspondente.
- O programa não deve aceitar valores fora da faixa convencional, e deve apresentar uma mensagem de erro.

1 → Domingo

2 → Segunda-feira

3 → Terça-feira

4 → Quarta-feira

5 → Quinta-feira

6 → Sexta-feira

7 → Sábado

Exercícios IV

6) Crie um programa que inicialmente receba dois números inteiros. Depois disso, mostre um menu com as seguintes opções:

1. Adição
2. Subtração
3. Multiplicação
4. Divisão

Peça para o usuário informar um valor de 1 a 4. Baseado nesse valor, calcule e mostre o resultado da opção escolhida pelo usuário, levando em consideração os dois números inteiros recebidos.