



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



Desenvolvimento de Software I

Revisão de Lógica com C

Prof: Diego Corrêa

Revisão - Lógica

- Variável?
- Operadores lógicos?
- Operadores Aritméticos?
- Operadores Relacionais?

Definição de variáveis

- Regras:
 - Máximo 30 caracteres para o nome da variável;
 - Todo nome deve ser composto apenas por letras, números e sublinhado
 - Deve começar com uma letra
 - Não pode ser igual a nenhuma palavra reservada, ou seja, palavras que já possuam um significado próprio para a linguagem como por exemplo, true, var ou if

Atividade

1. Na lista seguinte, assinale com **V** os nomes de variáveis válidos e com **I** os inválidos.

- | | | |
|---------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> abc | <input type="checkbox"/> 3abc | <input type="checkbox"/> a |
| <input type="checkbox"/> 123a | <input type="checkbox"/> a?B | <input type="checkbox"/> acdl |
| <input type="checkbox"/> _ | <input type="checkbox"/> Aa | <input type="checkbox"/> 1 |
| <input type="checkbox"/> Al23 | <input type="checkbox"/> _1 | <input type="checkbox"/> AO123 |
| <input type="checkbox"/> al23 | <input type="checkbox"/> _al23 | <input type="checkbox"/> b3l2 |
| <input type="checkbox"/> AB CDE | <input type="checkbox"/> etc... | <input type="checkbox"/> guarda-chuva |

Exemplo 1

- Principal maneira de armazenar informação em uma variável.

```
1  #include <stdio.h>
2
3  int main(){
4      float preco_unidade, preco_total;
5      int quantidade;
6
7      preco_unidade = 5.0;
8      quantidade = 10;
9      preco_total = preco_unidade * quantidade;
10
11     return 0;
12 }
```

Entrada/Saída de dados

- A entrada de dados foi criada para receber dados de um dispositivo, geralmente o teclado.

Ex:

leia x

```
scanf("%d", &x)
```

- A saída de dados é o meio pelo qual, informações contidas na memória do computador são colocadas no dispositivo de saída.

Ex:

Escreva x

```
printf("%d", x)
```


Exemplo 2

```

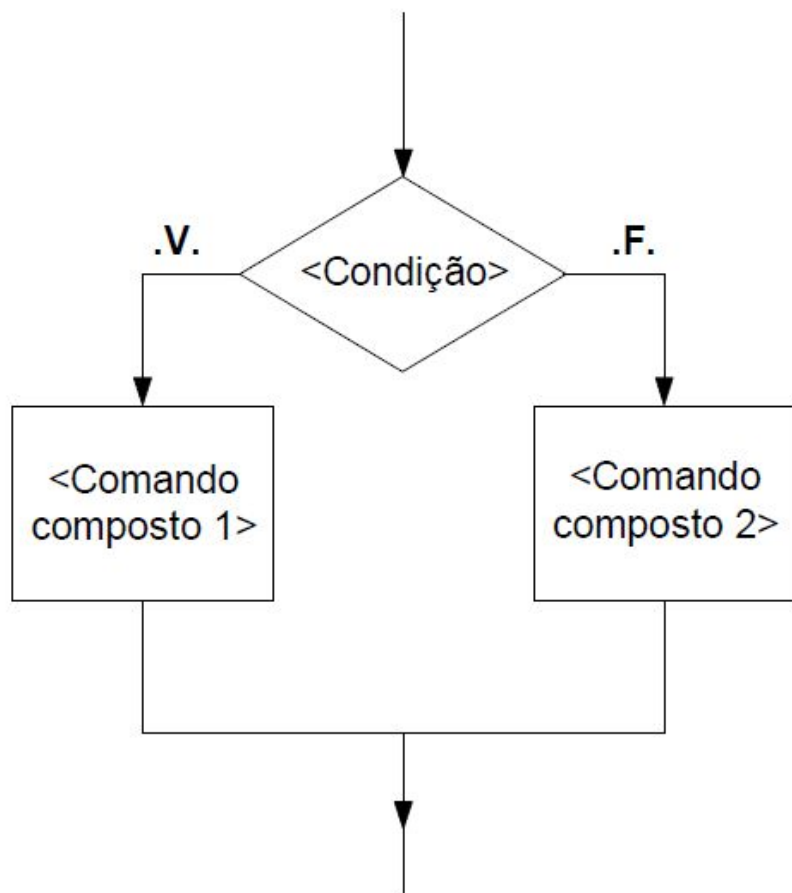
1  #include <stdio.h>
2
3  int main(){
4      float preco_unidade, preco_total;
5      int quantidade;
6
7      printf("Digite o preço da unidade: ");
8      scanf("%f", &preco_unidade);
9      printf("Digite a quantidade: ");
10     scanf("%d", &quantidade);
11
12     preco_total = preco_unidade * quantidade;
13     printf("O preço total é: %f\n", preco_total);
14
15     return 0;
16 }
```

Estrutura de decisão

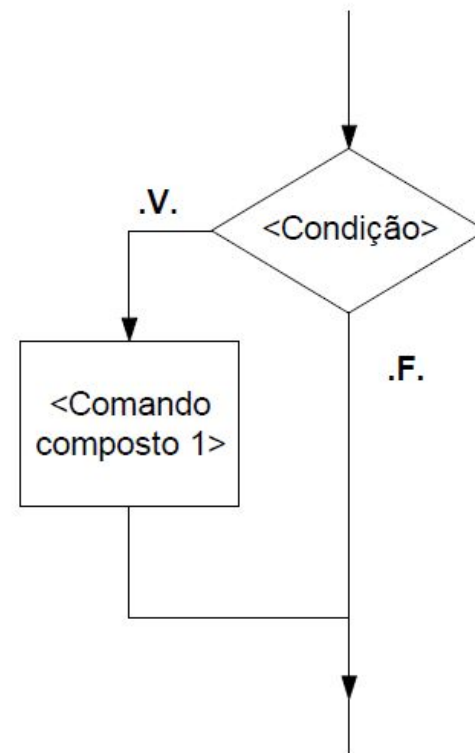
- Estrutura tipo Se:
 - Uma única condição (expressão lógica) é avaliada.

```
Se (<condição>) {
    <Comando_composto_1>;
} Senão {
    <Comando_composto_2>;
}
```

Estrutura “Se”

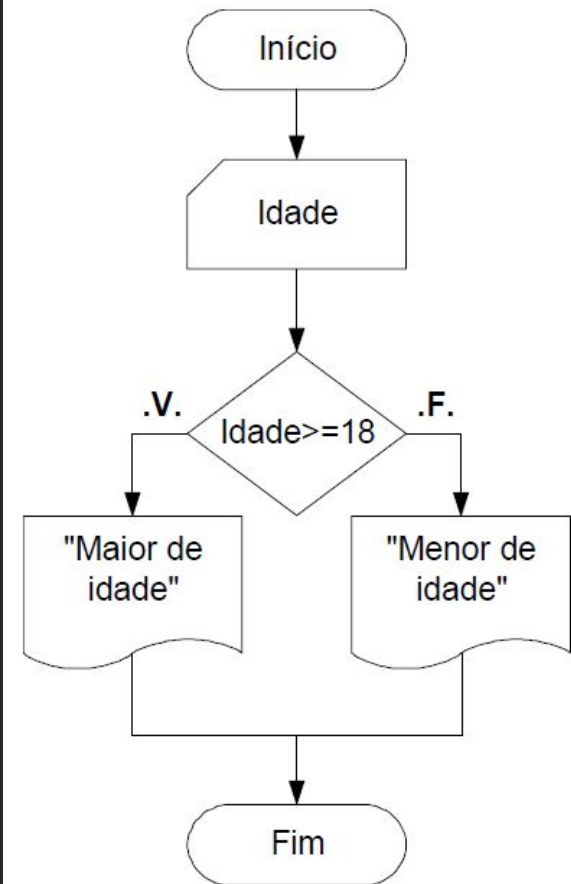


```
Se (<condição>)  
{  
    <Comando_composto_1>;  
}
```



Exemplo 3

```
1  #include <stdio.h>
2
3  int main(){
4      int idade, maior_de_idade = 18;
5
6      printf("Qual é a idade? ");
7      scanf("%d", &idade);
8
9      if (idade >= maior_de_idade){
10         printf("Maior de idade\n");
11     } else {
12         printf("Menor de idade\n");
13     }
14
15     return 0;
16 }
```



Exemplo 4

- Crie um algoritmo para verificar, dentre 2 números qual o maior deles.

Exemplo 4 - Solução

```

1  #include <stdio.h>
2
3  int main(){
4      int primeiro_numero, segundo_numero;
5
6      printf("Escreve o primeiro numero: ");
7      scanf("%d", &primeiro_numero);
8
9      printf("Escreve o segundo numero: ");
10     scanf("%d", &segundo_numero);
11
12     if (primeiro_numero > segundo_numero) {
13         printf("O primeiro número é maior que o segundo: %d > %d\n", primeiro_numero, segundo_numero);
14     } else {
15         printf("O segundo número é maior que o primeiro: %d < %d\n", primeiro_numero, segundo_numero);
16     }
17
18     return 0;
19 }

```

Funciona!

Mas possui uma falha!

Exemplo 5 - Solução

```

1  #include <stdio.h>
2
3  int main(){
4      int primeiro_numero, segundo_numero;
5
6      printf("Escreve o primeiro numero: ");
7      scanf("%d", &primeiro_numero);
8
9      printf("Escreve o segundo numero: ");
10     scanf("%d", &segundo_numero);
11
12     if (primeiro_numero > segundo_numero) {
13         printf("O primeiro número é maior que o segundo: %d > %d\n", primeiro_numero, segundo_numero);
14     } else if (primeiro_numero < segundo_numero){
15         printf("O segundo número é maior que o primeiro: %d < %d\n", primeiro_numero, segundo_numero);
16     } else {
17         printf("Os números são iguais: %d == %d\n", primeiro_numero, segundo_numero);
18     }
19
20     return 0;
21 }

```

Aninhamento

Se aninhando

- Verificam vários casos inserindo umas estruturas **Se/Senão** em outras.

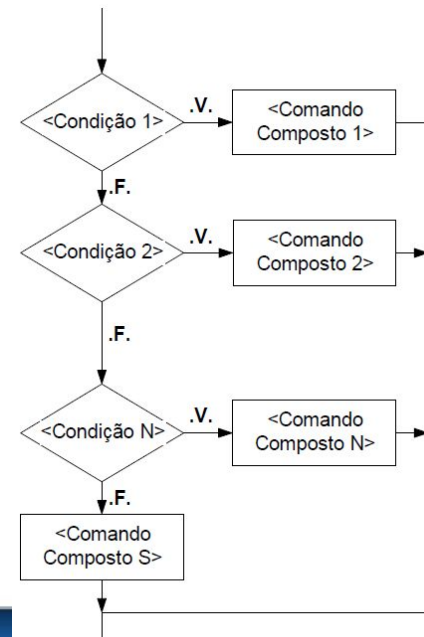
```

1  #include <stdio.h>
2
3  int main(){
4      float graus;
5
6      printf("Qual a temperatura: ");
7      scanf("%f", &graus);
8
9      if (graus >= 50) {
10         printf("Quente demais!\n");
11     } else if (graus >= 40 && graus < 50){
12         printf("Preciso de uma praia!\n");
13     } else if (graus >= 30 && graus < 40){
14         printf("Liga o ventilador, me leva pro ar condicionado!\n");
15     } else if (graus >= 20 && graus < 30){
16         printf("Ambiente agradável, mas as vezes faz calor!\n");
17     } else if (graus >= 10 && graus < 20){
18         printf("Bom mas as vezes faz frio\n");
19     } else if (graus >= 0 && graus < 10){
20         printf("Tá frio, me da cobertor!\n");
21     } else {
22         printf("Picolé humano!\n");
23     }
24
25     return 0;
26 }
```

Decisão do Tipo Escolha

- Este tipo de estrutura é uma generalização da estrutura **Se**, onde somente uma condição era avaliada e dois caminhos podiam ser seguidos.

```
Escolha (<variável>) {
    Caso <Valor_1> : <Comando_composto_1>; Pare;
    Caso <Valor_2> : <Comando_composto_2>; Pare;
    .....
    Caso <Valor_N> : <Comando_composto_N>; Pare;
    Senão <Comando_composto_S>;
}
```



Decisão do Tipo Escolha

```
1  #include <stdio.h>
2
3  int main(){
4      float graus;
5
6      printf("Qual a temperatura: ");
7      scanf("%f", &graus);
8      switch(int (graus/10)) {
9          case 0:
10             printf("Tá frio, me da cobertor!\n");
11             break;
12          case 1:
13             printf("Bom mas as vezes faz frio\n");
14             break;
15          case 2:
16             printf("Ambiente agradável, mas as vezes faz calor!\n");
17             break;
18          case 3:
19             printf("Liga o ventilador, me leva pro ar condicionado!\n");
20             break;
21          case 4:
22             printf("Preciso de uma praia!\n");
23             break;
24          case 5:
25             printf("Quente demais!\n");
26             break;
27          case 6 ... 10:
28             printf("Extinção!\n");
29             break;
30          default:
31             printf("Picolé humano!\n");
32      }
33
34      return 0;
35 }
```

Exemplo

- Escrever um algoritmo que solicite ao usuário três números inteiros. Coloque-os em ordem decrescente.

Laços de repetição

- Repetição com condição no início;
- Repetição com condição no fim;
- Repetição com variável de controle;

Repetição com condição no início

Enquanto *<expressão lógica>* faça
 <sequência de comandos>
fimenquanto

Atenção: Como o laço *enquanto...faça* testa sua condição de parada antes de executar sua sequência de comandos, esta sequência poderá ser executada zero ou mais vezes.

Repetição com condição no início

Ex:

```
1  #include <stdio.h>
2
3  int main() {
4      int contador = 10;
5
6      while(contador > 0){
7          printf("%d\n", contador);
8          contador = contador - 1;
9      }
10
11     return 0;
12 }
```

Repetição com condição no fim

Repita

<sequencia de comandos>

Ate *<expressão lógica>*

Atenção: Como o laço repita...ate testa sua condição de parada depois de executar sua seqüência de comandos, esta seqüência poderá ser executada uma ou mais vezes.

Repetição com condição no fim

- Ex:

```
1  #include <stdio.h>
2
3  int main() {
4      int contador = 10;
5
6      do {
7          printf("%d\n", contador);
8          contador = contador - 1;
9      } while(contador > 0);
10
11     return 0;
12 }
```

Repetição com variável

Para X de 1 ate 10 faça
 <lista de comandos>
fimpara

Atenção: Se, logo no início da primeira repetição, for maior que (ou menor, quando for negativo), o laço não será executado nenhuma vez

Repetição com variável

- Ex:

```
1  #include <stdio.h>
2
3  int main() {
4      int contador;
5
6      for(contador = 0; contador < 10; contador = contador + 1){
7          printf("%d\n", contador);
8      }
9
10     return 0;
11 }
```


Revisão Geral

Crie uma calculadora onde o usuário digitará o “primeiro número”, depois o “segundo número” e então um dos quatro sinais: ‘+’, ‘-’, ‘*’, ‘/’.

Cada operação deve ser descrita em sua própria função e ser chamada de acordo com a escolha do usuário.