



INTRODUÇÃO A LINGUAGEM C

Profª Ms. Erica Lopes
ericaolsilva@gmail.com

COMANDOS CONDICIONAIS

Na linguagem C o comando condicional mais simples é o **if – else**.

```
if (condição)
    comando1;
else
    comando2;
```

Comparando com algoritmo temos:

```
se condição entao
    comando1
Senao
    comando2
fimse
```



```
if (condição)
    comando1;
else
    comando2;
```

COMANDOS CONDICIONAIS

Ex:

```
#include<stdio.h>
main()
{
    int a,b;
    printf("digite dois números:");
    scanf("%d%d",&a,&b);
    if (b!=0)
        printf("%d\n",a/b);
    else
        printf("divisão por zero\n");
}
```

COMANDOS CONDICIONAIS

No exemplo anterior o comando if tinha apenas uma única instrução a ser executada. Para que possam ser executadas várias instruções é necessária usar a representação de instrução composta :

- Uma chave aberta ({)
- Uma sequência de instruções (cada uma terminada por ponto e vírgula ;)
- Uma chave fechada (})

COMANDOS CONDICIONAIS

Existe ainda uma estrutura condicional para casos de seleção múltipla: o **switch - case**.

```
switch(variável)
{
    case valor1:
        comandos 1;
        break;
    case valor2:
        comandos 2;
        break;
    ...
    default:
        comandos;
}
```

COMANDOS CONDICIONAIS

A variável é testada sucessivamente comparando com os valores de cada caso. Ao encontrar uma coincidência, o comando ou o bloco de comandos é executado.

Se nenhuma coincidência for encontrada o comando default será executado. O default é opcional.

A sequência de comandos é executada até que o comando break seja encontrado. Se não houver o comando break, todos os comandos abaixo serão realizados em sequência, mesmo que estejam declarados em outro bloco de comandos.

COMANDOS CONDICIONAIS

Comparando com algoritmo temos:

Teste(variável)

```
{  
  caso valor1  
    comandos 1  
  caso valor2  
    comandos 2  
  fimescolha  
  comandos  
}
```



switch (variável)

```
{  
  case valor1:  
    comandos 1;  
    break;  
  case valor2:  
    comandos 2;  
    break;  
  default:  
    comandos;  
}
```

COMANDOS CONDICIONAIS

```
#include<stdio.h>
main()
{
    int x;
    printf("1. incluir\n");
    printf("2. alterar\n");
    printf("3. excluir\n");
    printf("Digite sua opção:");
    scanf("%d",&x);
```

```
switch(x)
{
    case 1:
        printf("escolheu inclusão\n");
        break;
    case 2:
        printf("escolheu alteração\n");
        break;
    case 3:
        printf("escolheu exclusão\n");
        break;
    default:
        printf("opção inválida\n");
}
}
```


COMANDOS DE REPETIÇÃO

O comando mais simples de repetição é o **for** que define um valor inicial a uma variável e vai modificando-o automaticamente a cada execução:

```
for(inicialização; condição; incremento) comando;
```

- ✓ Inicialização: atribuição de um valor à variável de controle;
- ✓ Condição: teste que verifica quando o comando de repetição será encerrado;
- ✓ Incremento: modificação que será realizada no valor da variável a cada execução

COMANDOS DE REPETIÇÃO

Comparando com algoritmo temos:

```
para variável de valor-inicial ate valor-fim faça  
  comando  
fimpara
```



```
for(inicialização; condição; incremento)  
  comando;
```

COMANDOS DE REPETIÇÃO

Ex: Imprime valores de 1 a 100 um em cada linha.

```
#include<stdio.h>
main()
{
    int x;
    for(x=1;x<100;x++)
        printf("%d\n",x);
}
```

Obs: para mais de um comando não esquecer de usar a instrução composta com { e }

COMANDOS DE REPETIÇÃO

Outros exemplos:

- ✓ Para mudar o valor da variável de 2 em 2

```
for(x=1;x<100;x+=2)
    printf("%d\n",x);
```

- ✓ Para usar mais de uma variável de controle no mesmo comando for

```
for (x=0,y=0;x+y<100;++x,++y)
    printf("%d ",x+y);
```

COMANDOS DE REPETIÇÃO

Outro comando de repetição é o **while** que testa uma condição e executa um comando caso a condição seja verdadeira. Ao chegar no fim do comando um novo teste será executado para se determinar se a repetição continua ou se termina (caso o resultado do teste seja falso):

```
while(condição)
    comando;
```

- ✓ A condição é testada antes de se lançar a execução do comando.

COMANDOS DE REPETIÇÃO

Comparando com algoritmo temos:

```
enquanto (condição) faça  
    comando  
fimpara
```



```
while(condição)  
    comando;
```

COMANDOS DE REPETIÇÃO

Ex: Lê caracteres do teclado até que a letra a seja digitada.

```
#include<stdio.h>
main()
{
    char ch;
    while(ch != 'a')
        scanf("%c",&ch);
}
```

Obs: para mais de um comando não esquecer de usar a instrução composta com { e }

COMANDOS DE REPETIÇÃO

O último comando de repetição é o **do-while** que executa um comando e depois testa uma condição para se determinar se a repetição continua ou se termina (caso o resultado do teste seja falso):

```
do
{
    comando;
} while(condição);
```

- ✓ Ele realiza sempre pelo menos uma execução do comando já que o teste é só no fim.

COMANDOS DE REPETIÇÃO

Comparando com algoritmo temos:

```
repita  
    comando  
ate condição
```



```
do  
{  
    comando;  
} while(condição);
```

COMANDOS DE REPETIÇÃO

Ex: Lê caracteres do teclado até que a letra a seja digitada.

```
#include<stdio.h>
main()
{
    char ch;
    do
    {
        scanf("%c",&ch);
    } while(ch != 'a');
}
```