

## CONDICIONAIS

Comandos utilizados para testar diferentes situações e executar instruções definidas pelo programador. Os comandos mais utilizados são: IF; IF-ELSE; IF - ELSE IF – ELSE e o SWITCH-CASE.

### IF

Neste caso, verifica o valor a ser testado e caso seja verdadeiro executa a instrução logo abaixo, caso seja falso não faz absolutamente nada. Exemplo:

```
If(x>3)          //se x maior que 3
```

```
Execute a instrução.... //então
```

Não é necessário a utilização de chaves

### IF - ELSE

Este outro modelo verifica o valor a ser testado e caso seja verdadeiro executa a instrução logo abaixo, caso seja falso executa a outra instrução. Exemplo:

```
If(x>3)          //se x maior que 3
```

```
Execute instrução 1 //então
```

```
else //senão
```

```
Execute instrução2
```

O uso de chaves é opcional, sendo necessário em situações com IF encadeado.

### IF - ELSE IF - ELSE

Esta forma permite vários testes diferentes, ou seja, verifica o valor a ser testado e caso seja verdadeiro executa a instrução logo abaixo, caso seja falso testa a próxima condição, sendo que a última deverá utilizar obrigatoriamente a cláusula else. Exemplo:

```
If(x>3){          //se x maior que 3
```

```
Execute instrução 1 //então
```

```
}else if(x==2){ //senão se
```

```
Execute instrução2
```

```
}else if(x!=0){ //senão se
```

```
Execute instrução3
```

```
}else{ // senão
```

```
Execute instrução4
```

```
}
```

Necessário uso das chaves

## SWITCH-CASE

Este comando é parecido com o IF ELSE IF, sendo mais prático para algumas situações como por exemplo construção de menus de opção.

Exemplo:

```
Int x = 2;
```

```
switch(x){    //chaveia a variável

    case 1:

        execute instrução 1    //caso x seja 1, executa instrução 1

        break;

    case 2:

        execute instrução 2    //caso x seja 2, executa instrução 2

        break;

    case 3:

        execute instrução 3    //caso x seja 3, executa instrução 3

        break;

    default:

        execute a instrução padrão    //caso x seja qualquer outro valor, executa instrução padrão

        break;

}
```

O uso do break é necessário, pois ao cair em alguma situação deve encerrar a instrução.

## LAÇOS DE REPETIÇÃO

São usados para repetir instruções em loops definidos pelo programador.

Os comandos mais utilizados são: FOR, WHILE e DO-WHILE.

### FOR

O FOR possui uma sintaxe bem enxuta e característica, estima-se que é usado em 90% dos casos que necessitam de laços, devido sua praticidade.

Exemplo:

Imprimir no console os números de 0 a 100

```
for(int i=0; i<=100; i++){           //para i iniciando em 0, incremente 1 até i igual a 100
    printf("%d\n", i);
}
```

O primeiro parâmetro é o valor inicial do laço, o parâmetro do meio é o intervalo e o último parâmetro é o incremento para fazer o laço girar.

O i é uma variável local, dentro do for e tem vida útil apenas dentro do laço, ou seja, não existe fora dele.

## WHILE

Este comando é um pouco mais complicado de se utilizar, mas pode ser útil em casos que demandem mais flexibilidade no laço. Neste caso, o valor inicial vem fora da instrução, ou seja é declarado em uma variável antes do laço, tornando possível manter o último valor para a variável. A variável continua existindo após o término do laço.

Exemplo:

```
Int i=0;
```

```
While(i<=100){           //enquanto i menor igual a 100
    printf("%d\n", i);     //imprima i
    i++;                  //incremente 1 para fazer o laço girar, caso contrário o laço entrará em
                           loop infinito, pois não existirá nada que altere valor de i.
}
```

A versatilidade do while permite a utilização do incremento ou decremento em diferentes pontos da aplicação.

## DO-WHILE

É uma variação do while, onde o que se pede é que se execute a instrução enquanto a condição de teste for verdadeira.

Exemplo.

```
char ch = getch();
```

```
do{    //faça
```

```
printf("Acesso negado, tente novamente");
```

```
}while(ch!='a') //enquanto ch for diferente de a.
```

O exemplo acima poderia ser aplicado para o teste de senha.

### Uso do continue e do break.

Estes comandos são bastante utilizados para ajudar no controle dos laços de repetição.

Exemplo:

Exibir números de 0 a 100, não imprimir de 50 a 75 e terminar antes caso i igual a 100 ou maior que 95.

```
Int i=0;
```

```
While(i<=100){           //enquanto i menor igual a 100
    i++;                 //incremente 1 para fazer o laço girar, caso contrário o laço entrará em
                        //loop infinito, pois não existirá nada que altere valor de i.
    If(i==100 || i>95)    // se i igual a 100 ou i maior que 95 termine o laço
        break;
    If(i>50 && i<75)
        continue;        //se i maior que 50 e i menor que 75 continue deste ponto e não
                        //execute a instrução abaixo, no caso o printf.
    printf("%d\n", i);    //imprima i
}
```

Resultado esperado:

1

2

.

.

.

50

75

76

.

.

.

94

95

Fim do laço...