

Linguagem C

Professor Alexandre Moreno

Estrutura Condicional em Linguagem C

O comando IF executa uma comparação através de uma (ou mais) condições (relacionais ou lógicas):

```
#include <stdio.h>
#include <stdlib.h>
int main(){
```

Condição ou conjunto de condições.

Exemplos:

`a==5`
`(a<7) && (a>10)`

```
if (condição){
    // conjunto de comandos
}
system("PAUSE");
return(0) ;
}
```

Comandos que serão executados caso o resultado da condição acima seja verdadeira.

Exemplo1a:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int x;
    printf("Informe um valor: ");
    scanf("%d", &x);
    if (x>0)
    {
        printf("valor eh positivo \n");
    }
    system("PAUSE");
    return(0);
}
```

Será obrigado a utilização das chaves para delimitar os comandos somente se houver mais de um comando { }

Exemplo1b:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int x;
    printf("Informe um valor: ");
    scanf("%d", &x);
    if (x>0)
        printf("valor eh positivo \n");
    system("PAUSE");
    return(0);
}
```

Exemplo2:

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

int main(){
    int x,y;
    printf("Informe um valor: ");
    scanf("%d", &x);
```

```
    if (x<0) {
```

```
        y = 10;
        printf("valor eh negativo \n");
    }
```

```
    system("PAUSE");
```

```
    return(0);
```

```
}
```

Dois comandos
(obrigatoriedade de
coloca-los entre chaves)

Exemplo3:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int x;
printf("Informe um valor: ");
scanf("%d", &x);

if (x%2==0 || x<0){
    printf("valor eh par ou negativo \n");
}
system("PAUSE");
return(0);
}
```

Não esquecer de
colocar um parênteses
para envolver as
condições

Exemplo4a:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int x;
    printf("Informe um valor: ");
    scanf("%d", &x);
    if (x%2==0)
        printf("valor eh par \n");
    else
        printf("valor eh impar \n");
    system("PAUSE");
    return(0);
}
```

Exemplo4b:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int x;
    printf("Informe um valor: ");
    scanf("%d", &x);
    if (x%2==0)
        printf("valor eh par \n");
    else
        {
            y=12;
            printf("valor eh impar \n");}
    system("PAUSE");
    return(0);
}
```

Dois comandos
(obrigatoriedade de
coloca-los entre
chaves)

Exemplo5:

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

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

```
int main(){
```

```
int x;
```

```
printf("Informe um valor: ");
```

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

```
if (x==0)
```

```
printf("valor eh zero \n");
```

```
else
```

```
if (x>0)
```

```
printf("valor eh positivo \n");
```

```
else
```

```
printf("valor eh negativo \n");
```

```
system("PAUSE");
```

```
return(0);
```

```
}
```

se então
senão se

A ideia é atribuir a uma variável um valor (caso a expressão seja verdadeira) ou outro valor caso contrário (expressão falsa).

Variável = **condição** ? expressão1 : expressão2

Se a condição for verdadeira, a variável irá receber a expressão1, caso contrário receberá a expressão2

Exemplo: Imagine que tenhamos que solicitar dois números e apresentar qual é o maior deles.

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

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

```
int main(){
```

```
int x,y,z;
```

```
printf("Informe um valor: ");
```

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

```
printf("Informe outro valor: ");
```

```
scanf("%d", &y);
```

```
if (x>y)
```

```
    z=x;
```

```
else
```

```
    z=y;
```

```
printf("O maior numero eh %d\n", z);
```

```
system("PAUSE");
```

```
return(0);
```

```
}
```

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

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

```
int main(){
```

```
int x,y,z;
```

```
printf("Informe um valor: ");
```

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

```
printf("Informe outro valor: ");
```

```
scanf("%d", &y);
```

```
z = x > y ? x : y;
```

```
printf("O maior numero eh %d\n", z);
```

```
system("PAUSE");
```

```
return(0);
```

```
}
```

Utilizado para fazer condições múltiplas através do valor de uma variável do tipo INT ou CHAR

Forma geral:

```
switch(variável) {  
    case valor_1: comando; break;  
    case valor_2: comando; break;  
    case valor_n: comando; break;  
    default: comando;  
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int x;
    printf("Informe um valor: ");
    scanf("%d", &x);
    switch(x) {
        case 1:printf("um \n"); break;
        case 2:printf("dois \n"); break;
        default:printf("outro numero qualquer \n");
    }
    system("PAUSE");
    return(0);
}
```

- Pode haver mais de um comando por case sem necessidade de chaves, basta colocar o BREAK no final;
- Default assumirá qualquer alternativa que não esteja descrita nas anteriores;

Devido ao tamanho dos programas fontes, nos próximos exemplos serão apresentados apenas os comandos em questão, sendo omitidos (se necessário) as definições iniciais (bibliotecas, variáveis, mensagens ao usuários e entrada de dados).

Laços de Repetição em Linguagem C

Laços de Repetição em Linguagem C

while (enquanto)

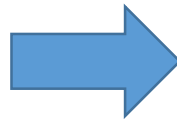
Permite executar repetidamente um conjunto de comandos de acordo com a condição (ou condições)

Forma geral:

```
while (condição) {  
    comandos;  
}
```



```
int x=1;
while (x<=10){
    printf("%d\n",x);
    x++;
}
system("PAUSE");
return(0);
}
```



```
1
2
3
4
5
6
7
8
9
10
Pressione qualquer tecla para continuar
```

Laços de Repetição em Linguagem C

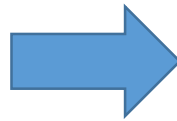
for

Permite executar repetidamente um conjunto de comandos de acordo com a condição (ou condições)

Forma geral:

```
for (inicialização; condição; incremento) {  
    comandos;  
}
```

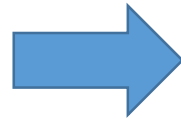
```
int x;  
    for (x=1; x<=10; x++){  
        printf("%d\n",x);  
    }  
    system("PAUSE");  
    return(0);  
}
```



```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Pressione qualquer tecla para continuar

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int x,b=1;
    for (x=1; ((x<=10) && (b<5)); x++){
        printf("%d\n",x);
        b++;
    }
    system("PAUSE");
    return(0);
}
```



1
2
3
4

Pressione qualquer tecla para continuar

Laços de Repetição em Linguagem C

do while (repita ate)

Permite executar (ao menos uma vez) repetidamente um conjunto de comandos de acordo com a condição (ou condições)

Forma geral:

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

```
int x=1;
```

```
do
```

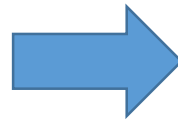
```
{  
    printf("%d\n",x);  
    x++;  
}
```

```
while (x<=10);
```

```
system("PAUSE");
```

```
return(0);
```

```
}
```



```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

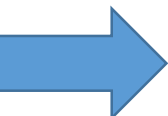
```
9
```

```
10
```

```
Pressione qualquer tecla para continuar
```



```
int x;  
do {  
    system("cls")        ;  
    printf("Escolha uma opcao \n");  
    printf("(1) Opcao 1 \n");  
    printf("(2) Opcao 2 \n");  
    printf("(3) Opcao 3 \n");  
    printf(" Escolha: ");  
    scanf("%d",&x);  
}  
while ((x<1) || (x>3));  
system("PAUSE");  
return(0);  
}
```



```
(1) Opcao 1  
(2) Opcao 2  
(3) Opcao 3  
Escolha:
```

Praticando...

- 1) Faça um programa em C que apresente todos os valores pares entre 3 e 120. Utilize o laço FOR
- 2) Faça o mesmo programa acima utilizando o WHILE
- 3) Faça o mesmo programa acima utilizando o DO WHILE
- 4) Faça um programa em C que solicite o nome, sexo e a idade de 3 pessoas. A idade deve ser validada (entre 1 e 120) e sexo (entre "M" e "F")
- 5) Apresente todos os números entre 11 e 132 em ordem inversa