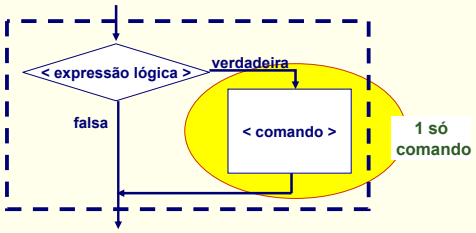
Comando condicional simples

C

```
if ( expressão lógica )
   comando ;
```

Linguagem algorítmica

```
se < expressão lógica >
   < comando >
```



Comando: simples / composto

Comando simples

comando;

1 só comando

Comando composto

```
{
  < comandos, separados por ponto e vírgula >
}
```

1 comando (bloco)

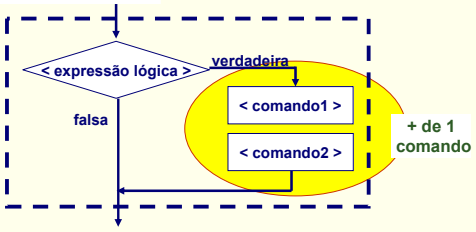
Comando condicional simples

C

```
if ( expressão lógica )
  {
     comando1;
     comandon;
  }
```

Linguagem algorítmica

```
se < expressão lógica >
   < comandoS >
```



Ex1: Fazer um programa que leia um número natural e informe se o número lido é par ou ímpar

Algoritmo **Par ou Ímpar**
{ Verifica se um valor lido do teclado é par ou ímpar }
entrada: o valor a ser testado
saídas: Mensagem de "par" ou "ímpar"
1. início
2. *ler* o valor
3. verificar se valor é par
4. se (ehpar)
4.1 escrever 'par'
5 se (não (ehpar))
5.1 escrever 'ímpar'
6. fim

Comando de seleção dupla!!

Comandos de seleção condicional

✓ Condicional Simples

```
if (condição)
    comando;
```

→ Seleção Dupla

```
if (condição)
   comando;
else comando;
```

• Seleção múltipla

```
switch (variável)
{
  expressão;
}
```

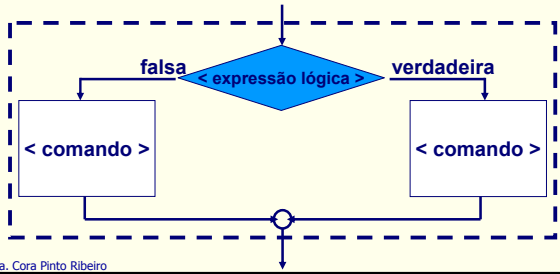

Comando de seleção dupla

C

```
if (expressão lógica)
   comando;
else
   comando;
```

Linguagem algorítmica

```
se < expressão lógica >
   < comando >
senão
   < comando >
```



1. início
2. *ler* o valor
3. verificar se valor é par
4 se (ehpar)
4.1 escrever 'par'
5 se (não (ehpar))
5.1 escrever 'ímpar'
6. fim

Algoritmo Par ou Ímpar

{ Verifica se um valor lido do teclado é par ou ímpar }
entrada: o valor a ser testado
saídas: Mensagem de "par" ou "ímpar"

1. início
2. *ler* o valor
3. verificar se valor é par ← resto da divisão inteira = 0
4. se (ehpar)
4.1 escrever 'par'
4.2 senão
4.2.1 escrever 'ímpar'
6. fim

Algoritmo Par ou Ímpar

{ Verifica se um valor lido do teclado é par ou ímpar }
entrada: o valor a ser testado
saídas: Mensagem de "par" ou "ímpar"

1. início
2. *ler* o valor
3. se (valor % 2 = 0)
3.1 ehpar ← 1 (true é diferente de zero)
3.2 senão
3.2.1 ehparr ← 0 (false é igual a zero)
4 se (ehpar)
4.1 escrever 'par' (executa se par é verdadeiro)
4.2 senão
escrever 'ímpar' (executa se par é falso)
5. fim


```c
// Determina se valor inteiro é par ou impar, analisando resto da divisão por 2
#include <stdio.h>
#include <stdlib.h>

int main( )
{
  int valor, ehpar;
  printf("digite o valor a ser testado\n");
  scanf("%d",&valor);
  if ((valor % 2) == 0) // analisa resto da divisão inteira
        ehpar = 1;      // verdadeiro
  else
        ehpar = 0;      // falso
  if (ehpar)            // equivale a (ehpar == 1) (<>0 é tr
        printf("PAR!\n");
  else
        printf("IMPAR!\n");  // ehpar == 0  -  false
  system("pause");
  return 0;
}
```

mesmo teste 2 vezes!


```c
// Determina se valor inteiro é par ou impar, analisando resto da divisão por 2
#include <stdio.h>
#include <stdlib.h>

int main( )
{
  int valor ;
  printf("digite o valor a ser testado\n");
  scanf("%d",&valor);

  // integra em um só "if":
  if ((valor % 2) == 0) // analisa resto da divisão inteira
        printf("PAR!\n");      // se true
  else
        printf("IMPAR!\n");    // se  false

  system("pause");
  return 0;
}
```


Condições em C:

o operador condicional "?"

Operador "?"

Sintaxe:

```
condição?expressão1:expressão2;
```

- Único operador ternário de C, ou seja, necessita de três argumentos.
- Uma condição é avaliada:
  . se a condição for verdadeira, o resultado da expressão1 é o valor resultado de toda a expressão;
  . se a condição for falsa, o resultado da expressão2 é o valor resultado de toda a expressão.
- Equivale a:

```
if (condição)
   expressão1;
else
   expressão2 ;
```

Operador "?"

Sintaxe:

condição?expressão1:expressão2;

Exemplo: Se salario for superior a 1000, terá um reajuste de 5% (conforme a expressão1), se salário for igual ou inferior a 1000, terá um reajuste de 7% (conforme a expressão2).

```
salario > 1000? salario = salario *1.05: salario = salario * 1.07;
```

Equivale à expressão: (escolher esta!!!)

```
if (salario > 1000)
    salario = salario *1.05;
else
    salario = salario * 1.07;
```


Condições em C:

~~o operador condicional "?"~~

usar IF!!!!

Ex1: Fazer um programa que leia um número natural e informe se o número lido é par ou ímpar

```
// Determina se valor inteiro é par ou impar, analisando resto da divisão por 2
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int valor, ehpar;
  printf("digite o valor a ser testado\n");
  scanf("%d",&valor);
  if (valor % 2 == 0) // analisa resto da divisão inteira
      printf("PAR!\n");        // se true
  else
      printf("IMPAR!\n");      // se false
  system("pause");
  return 0;
}
```

Consistir valor informado!

1. início
2. *ler* o valor
3. **se valor > 0**
3.1 se (valor mod 2 = 0)
3.1.1 escrever 'par'
3.1.2 senão
3.1.2.1 escrever 'ímpar'
4. fim


```
// Determina se valor inteiro é par ou impar, analisando resto da divisão por 2
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int valor, ehpar;
  printf("digite o valor a ser testado\n");
  scanf("%d",&valor);
  if (valor >=0)
     if ((valor % 2) == 0) // analisa resto da divisão inteira
        printf("PAR!\n");        // se true
     else
        printf("IMPAR!\n");      // se false
  system("pause");
  return 0;
}
```

1 único comando
if-else
Não é composto!

'If' aninhados

```
if ( expressão lógica )
   comando ;
```

```
if (expressão lógica)
   comando;
else
   comando;
```

comando:
- **comando simples**
- **comando composto (entre chaves!)**

Cuidado:
Determinar a qual else o if está ligado!

Solução:
Uso de chaves e ALINHAMENTO!


```
// Determina se valor inteiro é par ou impar, analisando resto da divisão por 2
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int valor;
  printf("digite o valor a ser testado\n");
  scanf("%d",&valor);
  if (valor >=0)
  { //serve para delimitar if - else - aqui é opcional (1 comando simples)
     if ((valor % 2) == 0) // analisa resto da divisão inteira
        printf("PAR!\n");        // se true
     else
        printf("IMPAR!\n");      // se false
  } //serve para delimitar if - else

  system("pause");
  return 0;
}
```

Ex2: Processar uma venda de livros em uma livraria.
Obter código do tipo de livro vendido (A, B, C) e número de unidades.
Preços: Tipo A: R$ 10,00
Tipo B: R$ 20,00
Tipo C: R$ 30,00
Calcular e informar preço a pagar. Caso tenham sido vendidas mais de 10 livros, emitir uma mensagem.

```
Algoritmo UmaVenda
{processa uma venda e avisa caso tenham sido vendidas maisde 10 unidades }
entradas: codigo {do tipo do livro}
          nro_livros {vendidos}
saídas:  apagar
         mensagem {caso tenham sido vendidas mais de 10 unidades}
1. início
2. ler codigo, nro_livros
3. se codigo = ´A´
3.1   apagar ← nro_livros * 10;
4. se codigo = ´B´
4.1   apagar ← nro_livros *  20;
5. se codigo = ´C´
5.1   apagar ← nro_livros *  30;
6. informar apagar
7. se nro_livros > 10
7.1   informar "mais de 10 livros vendidos"
8. fim
```

Redundância!!!! Encadear testes !!!!

'If' encadeados

```
If (condição)
  {
     < comando >
     if  (condição)
        < comando >
     else < comando>
  }
```


Ex2: Processar uma venda de livros em uma livraria.
Obter código do tipo de livro vendido (A, B, C) e número de unidades.
Preços: Tipo A: R$ 10,00
Tipo B: R$ 20,00

```
Algoritmo UmaVenda
{processa uma venda e avisa caso tenham sido vendidas maisde 10 unidades }
entradas: codigo {do tipo do livro}
          nro_livros {vendidos}
saídas:  apagar
         mensagem {caso tenham sido vendidas mais de 10 unidades}
1. início
2. ler codigo, nro_livros
3.  se codigo = ´A´
3.1     apagar ← nro_livros * 10
3.2 senão
3.2.1   se codigo = ´B´
3.2.1.1       apagar ← nro_livros *  20
3.2.1.2  senão
3.2.1.2.1   se codigo = ´C'
3.2.1.2.1.1         apagar ← nro_livros *  30
3.2.1.2.1.2 senão "Código inexistente!"
4. informar apagar
5. se nro_livros > 10
5.1   informar "mais de 10 livros vendidos"
6. fim
```

Consistência !!


```
/* processa uma venda e avisa quando mais de 10 unidades
foram vendidas */
#include<stdio.h>
#include<stdlib.h>

int main()
{
char codigo;
int nro_livros;
float apagar;
printf("digite o codigo do produto e número de unidades:\n");
scanf("%c %d",&codigo, &nro_livros); //sem brancos entre formatos!
if (codigo == 'a' || codigo == 'A')
   apagar = nro_livros*10;
else
   if (codigo == 'b' || codigo == 'B')
      apagar = nro_livros*20;
   else
      if (codigo == 'c' || codigo == 'C')
         apagar = nro_livros*30;
      else printf(" Codigo informado não existe");
printf("o valor a pagar e' R$ %4.2f\n",apagar);
if (nro_livros > 10)
   printf("Foram vendidas mais de 10 unidades do tipo %c",codigo);
system("pause");
return (0);
}
```

Se código inválido, não existe valor a ser pago!


```
// substitui trecho assinalado por:
if (codigo == 'a'|| codigo == 'A')
   apagar = nro_livros*10;
else
   if (codigo == 'b' || codigo == 'B')
      apagar = nro_livros*20;
   else
      if (codigo == 'c' || codigo == 'C')
         apagar = nro_livros*30;
      else
        {  //início de comando composto
           printf("Codigo informado nao existe\n \n");
           apagar = 0;    // confirma ausência de pagamento!
        } // fim de comando composto
if (apagar > 0)  // comandos executados apenas se ocorreu venda!
  {  //início de comando composto
     printf("O valor a pagar e' R$ %4.2f\n",apagar);
     if (nro_livros > 10)
        printf("Foram vendidas mais de 10 unidades do tipo
%c\n",codigo);
  } // fim de comando composto
```


Fazer: otimizar localização de um ponto no plano

Dados um par de valores x e y, que representam as coordenadas de um ponto no plano, determinar a localização do ponto: se em um quadrante, em um dos eixos ou na origem.

- leitura dos valores de x e y.
- determinação, pela avaliação de condições, de onde o ponto se encontra: se em um quadrante, em um eixo ou na origem.
- escrita da mensagem, onde é indicada a localização do ponto – apenas 1 mensagem por par de valores.

Em C:

```
/* Obtem coordenadas e informa localização, utilizando if: */
#include <stdio.h>
#include <stdlib.h>
int main  ( )
{
   float x , y;
   printf("\n Coordenadas: x = ");
   scanf("%f", &x);
   printf("  e y = ");
   scanf("%f", &y);
   if ( x == 0  &&  y == 0 )
      printf("\n Ponto na origem");
   if ( x > 0  &&  y > 0 )
      printf("\n Ponto no quadrante 1");
   if ( x < 0  &&  y > 0 )
      printf("\n Ponto no quadrante 2");
   if ( x < 0  &&  y < 0 )
      printf("\n Ponto no quadrante 3");
   if ( x > 0  &&  y < 0 )
      printf("\n Ponto no quadrante 4");
   if ( x == 0  &&  y  != 0 )
      printf("\n Ponto no eixo dos y");
   if ( x != 0  &&  y  == 0 )
      printf("\n Ponto no eixo dos x");
   printf("\n");
   system("pause");
   return 0;
}
```

Redundância!!!

Refazer, usando encadeamento!

Em C:

```
/* Obtem coordenadas e informa localização, utilizando if_else */
#include <stdio.h>
#include <stdlib.h>
int main  ( )
{
  float x , y;
  printf("\n Coordenadas: x = ");
  scanf("%f", &x);
  printf("  e y = ");
  scanf("%f", &y);



            Fazer para a próxima aula!!



  printf("\n");
  system("pause");
  return 0;
}
```



```
/* Obtem coordenadas e informa localização, utilizando if-else SOLUÇÃO 1 */
#include <stdio.h>
#include <stdlib.h>
int main  ( )
{
 float x , y;
 printf("\n Coordenadas: x = ");
 scanf("%f", &x);
 printf("  e y = ");
 scanf("%f", &y);
 if ( x == 0  &&  y == 0 )
    printf("\n Ponto na origem");
 else if ( x > 0  &&  y > 0 )
         printf("\n Ponto no quadrante 1");
      else if ( x < 0  &&  y > 0 )
              printf("\n Ponto no quadrante 2");
           else if ( x < 0  &&  y < 0 )
                   printf("\n Ponto no quadrante 3");
                else if ( x > 0  &&  y < 0 )
                        printf("\n Ponto no quadrante 4");
                     else if ( x == 0  &&  y  != 0 )
                             printf("\n Ponto no eixo dos y");
                          else // sobrou x != 0  &&  y  == 0 , nem precisa testar!
                              printf("\n Ponto no eixo dos x");
 printf("\n");
 system("pause");
 return 0;
}
```



```
/* Obtem coordenadas e informa localização, utilizando if_else SOLUÇÃO 2 */
#include <stdio.h>
#include <stdlib.h>
int main  ( )
{
  float x , y;
  printf("\n Coordenadas: x = ");                scanf("%f", &x);
  printf("  e y = ");                            scanf("%f", &y);
  // identifica origem e eixos
  if ( x == 0 )   // x = 0: possibilidade de origem ou eixo y
     if (y == 0 )  // y também = 0: localizou origem
        printf("\n Ponto na origem");
     else          // só pode ser eixo y
        printf("\n Ponto no eixo dos y");
  else      // x certamente é diferente de zero: não testa mais!
     if ( y  == 0 ) // localizou eixo y
        printf("\n Ponto no eixo dos x");
     else              // só sobraram os quadrantes!!!
        if ( x > 0 )     // quadrantes 1 ou 4, dependendo de y
           if ( y > 0 )  // quadrante 1
              printf("\n Ponto no quadrante 1");
           else          // quadrante 4, sem precisar mais testes
              printf("\n Ponto no quadrante 4");
        else //  x é < 0: sobraram quadrantes 2 e 3, dependendo de y
           if ( y > 0 )
              printf("\n Ponto no quadrante 2");
           else // sobrou x < 0 e y < 0: não precisa testar
              printf("\n Ponto no quadrante 3");
  printf("\n");
  system("pause");
  return 0;
}
```