

SELEÇÃO COM O COMANDO **switch** - (caso)

Este tipo de estrutura é uma generalização da construção Se, onde somente uma condição era avaliada e dois caminhos podiam ser seguidos. Na estrutura de decisão do tipo Caso pode haver uma ou mais condições a serem testadas e um comando diferente associado a cada uma destas.

A estrutura para instrução **caso...fim_caso...senão** é representado por:

A sintaxe da construção de Caso é:

Linguagem C
<pre>switch(Expressão) { case constante1 : Sequência de instruções 1 case constante2 : Sequência de instruções 2 ... case constante n : Sequência de instruções n default : Sequência de comando x }</pre>

Aí, a **expressão** argumento do comando deve resultar num valor do tipo **int** ou num valor do tipo **char** e, opcionalmente, a ultima instrução de cada uma das sequências Sequência de instruções i é **break**.

A semântica deste comando é bem simples: a Expressão é avaliada e as sequências de instruções situadas entre o valor da expressão apresentado nos cases e um comando break ou o delimitador do comando são executadas. Se o valor da Expressão for diferente de todas as opções dadas pelas constantes associadas aos cases, a sequência de instruções vinculada ao **default** será executada.

OBS * O último comando da Sequência de Comandos é o comando **break**;
Caso contrário serão executados todos os comandos que vierem a seguir



```

#include <stdio.h>
main()
{
    int n;
    printf("Digite um número inteiro entre 1 e 5 \n");
    scanf("%d", &n);
    switch (n)
    {
        case 1 : printf("Valor de n: %d \n", n); break;
        case 2 : printf("Valor do dobro de %d: %d \n", n, 2*n); break;
        case 3 : printf("Valor do triplo de %d: %d \n", n, 3*n); break;
        case 4 : printf("Valor do quadruplo de %d: %d \n", n, 4*n); break;
        default : printf("Valor digitado: %d \n", n);
    }
}

```

/* Programa que determina o numero de dias de um mês dado */

```

#include <stdio.h>
main()
{ int Mes, Ano, NumDias;
  printf("Digite o mes");
  scanf("%d", &Mes);
  if ((Mes == 4 ) || (Mes == 6) || (Mes == 9) || (Mes == 11))
      NumDias = 30;
  else
      if (Mes == 2)
      {
          printf("Digite o ano");
          //: um ano é bissexto se ele for divisível por 400 ou se ele for
          divisível por 4 e não por 100.

          scanf("%d", &Ano);
          if ((Ano % 400==0) || (Ano%4 ==0 && Ano%100 !=0))
              NumDias = 29;
          else
              NumDias = 28;
      }
      else
          NumDias = 31;

  printf("O mes %d tem %d dias", Mes, NumDias);
}

```

Funções de biblioteca

asin(x)	double	Arco cujo valor do seno e o argumento x
atan(x)	double	Arco cujo valor da tangente e o argumento x
cos(x)	double	Co-seno do argumento x
log(x)	double	Logaritmo natural do argumento x
log10(x)	double	Logaritmo decimal do argumento x
pow(x, y)	double double	Argumento x elevado ao argumento y
pow10(x)	int	10 elevado ao argumento x
rand(x)	double	Um número aleatório entre 0 e x-1
sin(x)	double	Seno do argumento x
sqrt(x)	double	Raiz quadrada do argumento x
Sqr(x)	int	Quadrado de x
tan(x)	double	Tangente do argumento x
tolower(x)	char	Converte o caracter x para minúsculo
toupper(x)	char	Converte o caracter x para maiúsculo

Exercício

Escrever um programa que leia um valor.

Se este valor for 1 escrever o próprio número, se o valor for 2 escrever o dobro, se o valor for 3 escrever o tripulo e se o valor for 4 escrever o quadruplo. Se não for nenhum escreva: "valor digitado = *valor*"

Escrever usando o comando if....else....

Exemplo: Usando o comando switch

O programa para determinar o numero de dias de um mes poderia utilizar o comando *switch*:

Linguagem C
<pre>#include <stdio.h> main() { int Mes, Ano, NumDias; printf("Digite o mes \n"); scanf("%d", &Mes); switch (Mes) { case 2 : printf("Digite o ano"); scanf("%d", &Ano); if (Ano % 400==0) (Ano%4 ==0 && Ano%100 !=0) NumDias = 29; else NumDias = 28; break; case 4 : case 6 : case 9 : case 11 : NumDias = 30; break; default : NumDias = 31; } printf.....</pre>

Os códigos especiais

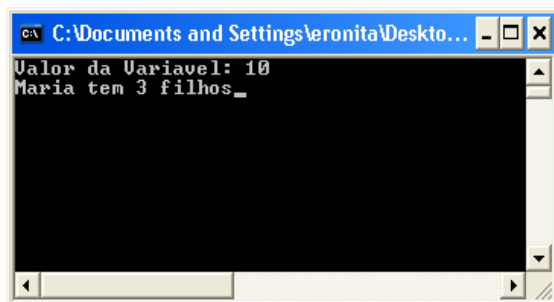
Códigos especiais da função printf()

Código	Ação
\n	leva o cursor para a próxima linha
\t	executa uma tabulação
\b	executa um retrocesso
\f	leva o cursor para a próxima página
\"	exibe o caractere "
\\	exibe o caractere \
\%	exibe o caractere %

Impressão dos Números Inteiros (INT)

```
// Impressão de Variáveis Inteiras
#include <stdio.h>
#include <conio.h>    // necessário para a função e getch main ()
{
    int Contador;
    int NroDeFilhos;

    Contador = 10;
    printf("Valor da Variavel: %d\n", Contador);
// No momento da execução sinal %d vai
// ser substituído pelo valor da
// variável Contador
    NroDeFilhos = 3;
    printf("Maria tem %d filhos", NroDeFilhos);
// o inteiro pode ficar no meio da string
    getch();
// espera que o usuário pressione uma tecla
}
```



Impressão dos Números Reais (FLOAT)

O padrão utilizado pela maioria dos compiladores C é exibir os números de ponto flutuante com seis

casas decimais. EX: (7.400000).

O número de casas decimais com as quais os números de ponto flutuante serão exibidos pode ser alterado pelo programa. Para isso deve-se acrescentar .n ao código de formatação da saída, sendo n o número de casas decimais pretendido.

Para formatar de maneira diferente usar-se, junto com o *%f* uma especificação de quantas casas decimais se deseja que o número tenha. Especifica-se também o número total de caracteres do número a ser impresso.

Por exemplo:

%6.3f especifica que se quer imprimir um **float** com **3 casas decimais** e com um **tamanho total** de **6** caracteres no total.

```
#include <stdio.h>
#include <conio.h>
main()
{
    float NotaDaP1, NotaDaP2;
    float Media;

    NotaDaP1 = 6.6; // Atribuição do Valores das médias
    NotaDaP2 = 8.2;

    Media = (NotaDaP1 + NotaDaP2) / 2.0;

    printf("Média Final : %6.3f", Media);
    // No momento da execução sinal %6.3f vai ser substituído
    // pelo valor da variável Media
    // Média Final : 7.400
    getch(); // espera que o usuário pressione uma tecla
}
```

Regras para impressão de um número real

- o número de casas decimais é sempre respeitado. Se for preciso, zeros serão acrescentados à direita do número
- o **tamanho total** significa o número de caracteres do número incluindo o ponto decimal e um eventual sinal de menos (-), se for o caso;
- Se a soma do número de caracteres da **parte inteira**, mais o **ponto decimal**, mais a **parte fracionária**, mais um **eventual sinal de menos** *ainda for menor* do que o tamanho total especificado no formato, então, espaços em branco serão acrescentados à esquerda da parte real do número.
- Se a soma do número de caracteres da **parte inteira**, mais o **ponto decimal**, mais a **parte fracionária**, mais um **eventual sinal de menos** for maior do que o tamanho total especificado no formato, então, apenas o número de casas decimais é respeitado

Por exemplo:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float Numero;
    Numero = -2.5;

    printf("1234567890\n");
    printf("%7.0f\n", Numero);
    printf("%7.3f\n", Numero);
    printf("%8.3f\n", Numero);
    printf("%9.3f\n", Numero);
    printf("\n");
    printf("%8.4f\n", Numero);
    printf("%8.1f\n", Numero);
    printf("%6.12f\n", Numero);

    getch();
}

// Resultados
1234567890
    -2      printf("%7.0f\n", Numero);
-2.500    printf("%7.3f\n", Numero);
-2.500    printf("%8.3f\n", Numero);
-2.500    printf("%9.3f\n", Numero);
          printf("\n");
-2.5000   printf("%8.4f\n", Numero);
    -2.5   printf("%8.1f\n", Numero);

-2.5000000000000    printf("%6.12f\n", Numero);
```

Alinhamento de números à DIREITA

Nos exemplos anteriores os números ficavam sempre alinhados a partir da esquerda. Experimente colocar um sinal de menos logo depois do sinal de % e veja o que acontece.

```
printf("%-7.3f\n", Numero);
```