

Declaração de variáveis e tipos de dados em C

A declaração de variáveis em C segue o padrão **< tipo_de_dados identificador; >**. Os tipos dados primitivos são quatro, sendo:

Palavra-chave	Tipo (descrição)
char	Caracter
int	inteiro
float	Ponto flutuante (real) de precisão simples
double	Ponto flutuante (real) de precisão dupla
void	vazio (sem valor)

Com exceção de void, os outros tipos de dados primitivos possuem modificadores que alteram o tamanho do tipo de dado ou sua forma de representação. Os modificadores fazem com que seja possível adequar o tipo de dados às necessidades de armazenamento em determinados casos. São eles: signed, unsigned, long e short. Para ver as formas de utilização consultar documentação mais abrangente.

Exemplos de declaração de variáveis

```
int a, b, c;  
float salario, _peso, altura23;  
char tipo, nome[50];
```

Segue as mesmas regras dos identificadores em algoritmos. Não pode conter caracteres especiais (Ex: \$%#@), não podem conter espaços e nem começar com números.

Estrutura mínima de um programa em C

```
// Diretivas de pré-processamento  
#include <stdio.h>    // Inclusão de bibliotecas auxiliares  
#include <stdlib.h>  
  
// Função principal  
int main(){           // Função principal que é acionada pelo S.O  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
}
```



Todo programa em C é composto no mínimo de duas partes que são as diretivas de pré-processamento e a função principal (main) que é o ponto de partida acionado pelo sistema operacional quando da execução.

Primeiro Programa em C

```
#include <stdio.h>    // Biblioteca de entrada e saída de dados

int main(){
    printf("Hello world\n");
    return 0;
}
```

Entrada e saída de dados

Comando leia: **scanf**

O comando **scanf** é composto por dois parâmetros, sendo o primeiro o tipo de dados e o segundo a variável para qual o valor será lido.

Sintaxe:

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

onde:

- “%d”: indica o formato (tipo) do dado que será recebido pelo fluxo de entrada. Neste caso (utilizando “%d”) será dado entrada de um valor do tipo double. Para maiores detalhes consultar outras opções para formatação de fluxo de entrada e saída de dados;
- &: indica que será utilizado passagem de parâmetro por referência (endereço do identificador na memória)
- variável: identificador da variável.

Comando escreva: **printf**

O comando **printf** é composto por dois parâmetros, sendo o primeiro o tipo de dados e o segundo a variável que será escrita.

Sintaxe:

```
printf("%d", variavel);
```

onde:

- “%d”: indica o formato (tipo) do dado que será escrito pelo fluxo de saída. Neste caso (utilizando “%d”) será escrito um valor do tipo double. Para maiores detalhes consultar outras opções para formatação de fluxo de entrada e saída de dados;
- variável: identificador da variável.

Formatação de fluxo de entrada e saída de dados



Tipo de Dados	Formatação
char	"%C"
int	"%i"
float	"%f" ou "%.2f" para formatar casas decimais
char[]	"%s"

Operadores lógicos, relacionais e aritméticos

Operador	Tipo
&&	E lógico
	Ou lógico
!	Não
==	Igual
!=	Diferente
>=	Maior ou igual a
<=	Menor ou igual a
/	Divisão
%	Módulo - Resto da divisão inteira
//	Div - Parte inteira da divisão inteira
pow(a,b)	Potência de a^b . É necessário incluir a biblioteca math.h
sqrt(a)	Raiz quadrada de a. É necessário incluir a biblioteca math.h

Estrutura condicional se - senão (if - else)



```
if ( <condição> ) {  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
} else {  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
}
```

Estrutura condicional caso (switch - case)

```
switch (variável) {  
    case constante1 :  
        comando 1;  
        comando n;  
        break;  
    case constante2 :  
        comando 1;  
        comando n;  
        break;  
    default :  
        comando 1;  
        comando n;  
}
```

Estrutura de repetição enquanto (while)

```
while (<condição>) {  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
}
```

Estrutura de repetição repita (do - while)



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CATARINENSE
Campus Videira

Mini Apostila de Programação em C

Prof. Dr. Manassés Ribeiro

```
do {  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
} while ( <condição> );
```

Estrutura de repetição para-faça (for)

```
for (valor_inicial; <condição>; passo) {  
    comando 1;  
    comando 2;  
    ...  
    comando n;  
}
```

Exemplo:

```
for (int cont=0; cont < 10; cont++) {  
    printf("%i\n", cont);  
}
```

Exemplo de programa em C - Mínimo divisor comum (MDC.c)

O programa abaixo encontra o mínimo divisor comum de dois números inteiros informados pelo usuário.



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CATARINENSE
Campus Videira

Mini Apostila de Programação em C

Prof. Dr. Manassés Ribeiro

```
#include <stdio.h>

int main(){
    int n1, n2, aux, cont=1, mdc=1;
    printf("Digite n1: ");
    scanf ("%i", &n1);
    printf("Digite n2: ");
    scanf ("%i", &n2);

    if (n1 > n2){
        aux = n1;
        n1 = n2;
        n2 = aux;
    }

    while (cont <= n1){
        if (n1 % cont == 0 && n2 % cont == 0)
            mdc = cont;
        cont++;
    }

    printf("MDC de %d e %d é: %d\n", n1, n2, mdc);
    return 0;
}
```