

# Programação Procedimental - Aulas Práticas

Universidade Federal de Uberlândia - 2013

## 1 Tipos de Dados e Variáveis

Para que um programa possa definir seu conjunto de instruções, é necessário que espaços de memória sejam reservados, pois serão utilizados para o armazenamento de valores durante a execução do programa. Para tanto, são definidas **variáveis**, que são associadas aos espaços de memória e armazenam tais valores.

Uma declaração de variável em linguagem C consiste no nome de um tipo, seguido do nome da variável (ou de uma lista de nomes de variáveis), seguido de ponto-e-vírgula.

```
tipo_da_variavel nome_da_variavel;  
tipo_da_variavel lista_de_nomes;
```

O nome de uma variável deve conter um ou mais caracteres, sendo o primeiro sempre uma letra, e não pode ser igual às palavras reservadas da linguagem C. Dependendo do local em que uma variável é declarada podemos classificá-la de **local** ou **global**:

- Variável local: É declarada dentro de uma função e será utilizada apenas dentro do bloco em que foi declarada. Essa variável somente existe quando o bloco de instruções em que foi declarada estiver ativo (sendo executado).
- Variável global: É declarada a parte, ficando fora de todas as funções do programa e pode ser utilizada em qualquer parte do programa. Essa variável existe durante todo o tempo de vida do programa.

Existem cinco tipos de dados básicos são eles: caracter, inteiro, ponto flutuante, ponto flutuante de precisão dupla e nulo (char, int, float, double, void), mostrados na tabela abaixo. Os dados da tabela dados podem variar conforme o tipo de processador e a implementação do compilador C utilizado.

| Nome do tipo | Tamanho em Bytes | Intervalo de Valores                 |
|--------------|------------------|--------------------------------------|
| char         | 1                | -128 a 127 (código ASCII)            |
| int          | 2                | -32768 a 32767                       |
| float        | 4                | $3.4 * 10^{-38}$ a $3.4 * 10^{38}$   |
| double       | 8                | $1.7 * 10^{-308}$ a $1.7 * 10^{308}$ |
| void         | 0                | nenhum valor                         |

Como exemplo de declaração de variáveis, temos:

```
int num;  
float a, b, c;  
char letra;
```

Podem-se aplicar modificadores sobre os tipos básicos. Um modificador altera o significado de um tipo:

- Long: indica um tipo de dados maior para *int* e *double*.

- Short: indica valores inteiros curtos *short int*.
- Signed: permite o uso de sinal (pode ser usado para caracteres).
- Unsigned: não permite o uso de sinais, assim, valores negativos não são válidos.

Após feita a declaração de tipos para variáveis, podem-se atribuir valores para as mesmas utilizando-se o símbolo = e a sintaxe `nome_variavel = expressao`. A expressão armazenada pode ser uma constante ou uma formula matemática:

```
#include <stdio.h>
int main() {
    int idade;
    float peso, altura;
    float imc;

    idade = 28;
    peso = 67;
    altura = 1.79;
    imc = peso / (altura * altura);

    return 0;
}
```

Para o trecho de programa acima, utilizou-se o cálculo do índice de massa corporal (IMC), desenvolvido pela Organização Mundial de Saúde, através da fórmula  $\text{peso} / (\text{altura} \times \text{altura})$ . Para que os valores da idade, peso e altura sejam informados pelo usuário, e o valor calculado seja impresso na tela, podemos modificar o código:

```
#include<stdio.h>
int main()
{
    int idade;
    float peso, altura;
    float imc;

    printf("\nEntre com a idade :");
    scanf("%d",&idade);
    printf("\nEntre com o peso :");
    scanf("%f",&peso);
    printf("\nEntre com a altura :");
    scanf("%f",&altura);

    imc = peso / (altura * altura);

    printf("\nO Indice de Massa Corporal eh %f",imc);
    return 0;
}
```

As funções `printf` e `scanf` para saídas e entradas formatadas estão detalhadas na próxima aula, e mais informações sobre variáveis e tipos podem ser encontradas na 2a. vídeo-aula em <http://programacaodescomplicada.wordpress.com/indice/linguagem-c/>.

## 1.1 Exercícios Propostos

1. Seja o código abaixo. Execute o código e verifique se está correto. Caso não esteja, identifique os erros (veja as mensagens geradas pelo compilador) e faça as correções necessárias.

```
#include <stdio.h>
int Main(void){
int a=1; b=2; c=3;
printf("Os números são: %d, %d e %d\n", a, b, c, d)
}
```

2. Modifique o programa anterior sobre cálculo de massa corporal para que a saída de dados informe além da massa corporal calculada, uma análise sobre o resultado. Essa análise deve se basear nos dados abaixo para o IMC:

| Resultado          | Valor IMC      |
|--------------------|----------------|
| Abaixo do peso     | Abaixo de 18.4 |
| Peso normal        | 18.5 - 24.9    |
| Sobrepeso          | 25.0 - 29.9    |
| Obesidade Grau I   | 30.0 - 34.9    |
| Obesidade Grau II  | 35.0 - 39.9    |
| Obesidade Grau III | Acima de 40    |

## Referências

- [1] Salvetti, D. D., Barbosa M. B. *Algoritmos*, Makron Books, 1993.