

Programação de Computadores I

Variáveis, Tipos de Dados e Operadores.

Variáveis

Variável em linguagem de programação é um espaço nomeado na memória do computador usado para armazenar, recuperar e manipular dados (como números, textos ou objetos) que podem ser alterados durante a execução de um programa. Funciona como uma "caixa" ou "etiqueta" que guarda uma informação específica, sendo fundamental para tornar o programa dinâmico e interativo

Variáveis

- Características principais de uma variável:
 - Nome (Identificador): Um nome único utilizado para acessar o valor armazenado.
 - Valor: O dado atual contido na variável, que pode mudar (ser reatribuído) ao longo do tempo.
 - Tipo: Define que tipo de dado a variável pode guardar (ex: inteiro, texto/string, booleano), dependendo da linguagem

Variáveis e Tipos de Dados

- Os tipos de variáveis em C definem o tamanho e a interpretação dos dados na memória.
- Os tipos básicos incluem:
 - int (inteiros),
 - float (ponto flutuante de precisão simples),
 - double (ponto flutuante de precisão dupla) e
 - char (caracteres ASCII).
- Podem ser modificados por short, long, signed e unsigned para alterar o alcance

Tipos de Dados

A tabela abaixo traz o tamanho e os valores que podem representar cada tipo com seus modificadores.

Palavra Chave	Tipo	Tamanho (Bytes)	Intervalo
char	Caracter	1	-128 a 127
unsigned char	Caractere sem sinal	1	0 a 255
int	Inteiro	2	-32.768 a 32.767
unsigned int	Inteiro sem sinal	2	0 a 65.535
short int	Inteiro curto	2	-32.768 a 32.767
long int	Inteiro longo	4	-2.147.483.648 a 2.147.483.647
unsigned long int	Inteiro longo sem sinal	4	0 a 4.294.967.295
float	Ponto flutuante precisão simples	4	3.4E-38 a 3.4E+38
double	Ponto flutuante precisão dupla	8	1.7E-308 a 1.7E+308

Variáveis e Tipos de Dados

Para uma variável existir em um programa é necessário fazer sua declaração, ou seja, dizer ao programa qual é, dando um nome, e que tipo de informação ela armazenará durante todo o programa.

- Obs.: O valor da variável muda durante a execução.

Ao declarar uma variável o computador reserva um espaço, iniciando em uma determinada posição (endereço) da memória principal para armazenar os valores atribuídos à variável.

- Obs.: Não é garantido que esse espaço reservado esteja “vazio”, normalmente existem valores que não sabemos quais são ali, o que chamamos de “lixo”.

Declaração de Variáveis

A declaração de variáveis em C segue o padrão:

```
TIPO LISTA_VARIAVEIS;
```

Onde :

TIPO : é um modificador (opcional) seguido de um tipo básico;

LISTA_VARIAVEIS: é uma lista de nomes de variáveis separados por vírgula.

Por exemplo:

```
sort int numero_alunos, numero_provas;
```

```
float nota1, nota2, nota3, nota4, _media;
```

Nomeando Variáveis

Para dar nome a uma variável em C é preciso observar as seguintes regras:

- Deve iniciar com uma letra ou sublinhado (_);
- Pode conter somente letras, dígitos(0-9) e sublinhado;
- Não podem conter caracteres especiais (@,!,\$,#...) e nem espaço
- Não podem ser palavras reservadas (int, char, if,);
- Geralmente são considerados os primeiros 31 caracteres.

Nomeando Variáveis

As boas práticas de programação recomenda:

- Utilizar nomes significativos e claros ; e
- Usar *cameCase* (ex.: totalVenda) ou *snake_case* (ex.: total_venda).

Outra boa prática recomendada é inicializar as variáveis antes de usá-las; ou seja, atribuir-lhes um valor inicial para garantir um valor conhecido para a variável. Por exemplo:

```
float totalVendas;  
totalVendas=0.0;  
int numero_itens=0;
```

Operadores

Uma das maneiras de manipular variáveis é através de operações sobre uma ou duas variáveis por vez aplicando operadores.

Os operadores em C são símbolos e estão divididos em categorias de acordo com o tipo de dado dos resultados que essas operações retornam.

- Estão divididos em operadores: aritméticos, relacionais, lógicos, de atribuição, de incremento, bit-a-bit(bitwise) e especiais.

Operadores Aritméticos

Os operadores aritméticos são aplicados sobre variáveis numéricas, int, float e double, e retornam um valor numérico. São operadores de:

- adição(+),
- subtração(-),
- divisão(/),
- multiplicação(*) e
- resto(%).

Operadores Relacionais

- Os operadores relacionais fazem comparações, são aplicados sobre duas variáveis com dados do mesmo tipo básico e retornam dois valores, verdadeiro ou falso. São os operadores:
 - Igual a (==);
 - Não igual (!=);
 - Maior que (>);
 - Maior igual a (>=);
 - Menor que (<);
 - Menor igual a (<=).

Operadores Lógicos

Os operadores lógicos são aplicados sobre uma ou duas variáveis ou ainda sobre expressões relacionais, também retornam verdadeiro ou falso. São operadores lógicos:

- E (&&) : aplicado sobre dois operandos, retorna verdadeiro quando os dois operandos são verdadeiros;
- OU (||) : aplicado sobre dois operandos, retorna verdadeiro sempre que pelo menos um dos operandos seja verdadeiro; e
- NÃO (!) : aplicado sobre um operando retorna verdadeiro quando o operando for falso, e retorna falso sempre que o operando for verdadeiro.

Operadores bit-a-bit

Os operadores bit-a-bit (bitwise) são aplicados sobre as representações binárias dos valores.

- & (E bit-a-bit) : compara bit a bit os valores à esquerda e à direita , resultando 1 quando os bits comparados são iguais a 1. Por exemplo:

$$\begin{array}{l} 5 \ \& \ 3 = 1 \text{ porque} \\ 5 =^2 00000101 \\ 3 =^2 00000011 \\ \hline 00000001 =^{10} 1 \end{array}$$

Operadores bit-a-bit

- | (OU bit-a-bit) : compara bit a bit os valores à esquerda e à direita , resultando 1 quando pelo menos um dos bits comparados é igual a 1. Por exemplo:

$$\begin{array}{l} 5 | 3 = 7 \text{ porque} \quad 5 =^2 00000101 \\ \quad \quad \quad \quad \quad 3 =^2 00000011 \\ \quad \quad \quad \quad \quad 00000111 =^{10} 7 \end{array}$$

- ^ (XOR – Exclive OR) : compara bit a bit os valores à esquerda e à direita , resultando 1 quando os bits comparados forem diferentes. Por exemplo:

$$\begin{array}{l} 5 \wedge 3 = 6 \text{ porque} \quad 5 =^2 00000101 \\ \quad \quad \quad \quad \quad 3 =^2 00000011 \\ \quad \quad \quad \quad \quad 00000110 =^{10} 6 \end{array}$$

Operadores bit-a-bit

- \sim (not bit-a-bit): operador unitário, inverte o valor dos bits. Por exemplo:

$$\sim 5 = 250 \text{ porque } 5 =^2 00000101$$

$$11111010 =^{10} 250$$

- \gg (shift para direita): operador unitário, desloca para a direita os bits do valor à esquerda, o número de vezes indicado pelo valor à direita. Exemplo:

$$- \quad 5 \gg 1 = 2 \text{ porque } 5 =^2 0000101$$

$$0000010 =^{10} 2$$

Operadores bit-a-bit

- \ll (shift para esquerda): operador unitário, desloca para a esquerda os bits do valor à esquerda, o número de vezes indicado pelo valor à direita. Exemplo:

$5 \ll 1 = 10$ porque $5 =^2 00000101$

$00001010 =^{10} 10$

Operadores de Atribuição

Os operadores de atribuição são usados para definir ou atualizar o valor de uma variável, onde o valor à esquerda deve ser uma variável e o valor à direita pode ser qualquer expressão válida. São simples ou compostos.

- = : atribui o valor à direita para a variável à esquerda. (A = 10)
- += : soma o valor à direita ao valor da variável à esquerda e atribui o resultado à variável à esquerda.(A += 10)
- -= : subtrai o valor à direita ao valor da variável à esquerda e atribui o resultado à variável à esquerda.(A -= 10)

Operadores de Atribuição

- `*=` : multiplica o valor à direita ao valor da variável à esquerda e atribui o resultado à variável da esquerda.(`A *= 10`)
- `/=` : divide o valor à direita ao valor da variável à esquerda e atribui o resultado à variável da esquerda.(`A /= 10`)
- `%=` : calcula o resto da divisão do valor da variável à esquerda pelo valor da direita e atribui o valor à variável da esquerda.(`A%=2`)
- `&=` : faz o E bit-a-bit entre o valor da variável à esquerda e o valor à direita e atribui à variável da esquerda.

Operadores de Atribuição

- $|=$: faz o OU bit-a-bit entre o valor da variável à esquerda e o valor à direita e atribui à variável da esquerda.
- $\wedge=$: faz o XOR bit-a-bit entre o valor da variável à esquerda e o valor à direita e atribui à variável da esquerda.
- $\ll=$: faz o deslocamento à esquerda e atribui à variável da esquerda.
- $\gg=$: faz o deslocamento à direita e atribui à variável da esquerda.