

Programação em C

Cledja Rolim

Informações

- Variável
 - Posição de memória cujo conteúdo pode ser modificado durante a execução de um programa
 - Identificador
 - Tipo de dados

Nome ou identificador

- É uma sequência de letras, dígitos e caractere para sublinhamento escolhida pelo programador e será utilizado no programa para se fazer referência àquela variável
- Regras para definir identificadores?
- Exemplos:
 - Válidos: *A, a, nota, Nota, n1, n2, nota_1, a32*
 - Inválidos: *5b, e 12, x-y, prova 2n, nota (2), case*

Palavras reservadas em C

Tabela 2 Palavras reservadas da linguagem C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	Signed	void
default	goto	sizeof	volatile
do	if	static	while

Tipo de dados

- Tipo de dado associado a uma variável é o conjunto dos valores que podem ser nela armazenados.

Tabela 3 Tipos de dados da Linguagem C

Denominação	Número de Bytes	Conjunto de valores
char	1	caracteres codificados no código ASCII
int	2	números inteiros de -32768 a 32767
long ou long int	4	números inteiros de -65536 a 65535
float	4	números reais de $-3,4 \times 10^{38}$ a $-3,4 \times 10^{-38}$ e $3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
double	8	números reais de $-1,7 \times 10^{308}$ a $-1,7 \times 10^{-308}$ e $1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
void	0	conjunto vazio

Declaração de variáveis

- *Tipo_de_dado Lista_de_identificadores;*

int a; / declara uma variável do tipo int */*

int b; / declara uma variável do tipo int */*

float c; / declara uma variável do tipo float */*

int d, e; / declara duas variáveis do tipo int */*

Declaração de variáveis

`int a = 5, b = 10; /* declara e inicializa duas variáveis do tipo int */`

`float c = 5.3; /* declara e inicializa uma variável do tipo float */`

Variáveis só armazenam valores do mesmo tipo com que foram declaradas

`int a; /* declara uma variável do tipo int */`

`a = 4.3; /* a armazenará o valor 4 */`

Declaração de variáveis

```
int a, b, c; /* declara e inicializa duas variáveis do tipo int */
```

```
a = 2;
```

```
c = a + b; /* ERRO: b contém "lixo" */
```


Expressões aritméticas

Tabela 4 Operadores aritméticos

Operador	Operação
+	adição
-	subtração
*	multiplicação
/	divisão

Exemplo

int a

double b, c;

a = 3; /* a recebe o valor 3 */

b = a / 2.0; /* b recebe o valor 1.5 */

c = 3.2 * b; /* c recebe o resultado da operação */

Operador módulo: %

- Aplicado apenas em números inteiros
- Retorna o resto da divisão de dois números
- Exemplo:

```
int a = 4, b = 2, c;
```

```
c = a % b;
```

Operadores de incremento e decremento

- Incrementa ou decrementa de uma unidade o valor de uma variável
- Pode ser antes ou depois da variável ser utilizada
 - `n++` incrementa `n` de uma unidade, depois de ser usado
 - `++n` incrementa `n` de uma unidade, antes de ser usado

Exemplo

`n = 5;`

`x = n++;` `/* x recebe 5; n é incrementada para 6 */`

`x = ++n;` `/* n é incrementada para 6; x recebe 6 */`

`a = 3;`

`b = a++ * 2;` `/* b termina com o valor 6 e a com o valor 4 */`

Operadores relacionais

- $<$, $<=$, $==$, $>=$, $>$, $!=$
- O resultado será 0 ou 1

```
int a, b;
```

```
int c = 23;
```

```
int d;
```

```
d = c + 1;
```

```
a = c < 20;
```

```
b = d > c;
```

Operadores lógicos

- $\&\&$, $\|$, $!$
- A avaliação é da esquerda para a direita
- A avaliação para quando o resultado pode ser conhecido

Tabela 5 Avaliação de expressões lógicas

R_1	R_2	$(R_1)\&\&(R_2)$	$(R_1)\ (R_2)$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Tabela 6 Operador unário !

R_1	$!R_1$
1	0
0	1

Exemplo

```
int a, b;
```

```
int c = 23;
```

```
int d = c + 4;
```

```
a = (c < 20) || (d > c); /* retorna 1 */
```

```
/* as duas sub-expressões são avaliadas */
```

```
b = (c < 20) && (d > c); /* retorna 0 */
```

```
/* apenas a primeira sub-expressão é avaliada *
```

Exercício

- Defina as variáveis a, b e c para obter todas as possíveis respostas. Suponha que:

$$a = 3$$

$$b = a / 2$$

$$c = b + 3.1$$

- Qual é o valor de c?

$$\%00C = 4.6$$

$$\%00C = 4.1$$

$$\%00C = 4$$

Saída de dados

- Função “printf”: possibilita a saída de valores segundo um determinado formato
- *printf(Mensagem, Lista de argumentos);*

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Estou aprendendo a programar em C");
```

```
}
```

Saída de dados

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int x = 2;
```

```
    float a, b;
```

```
    a = 2.1;
```

```
    b = 3.2;
```

```
    printf("Valores das variáveis e expressão: %f , %f e  
    %d, %f", a, b , x, (a + b + c)/3);
```

```
}
```

Especificação de formato

- %c especifica um char
- %d especifica um int
- %u especifica um unsigned int
- %f especifica um double (ou float)
- %e especifica um double (ou float) no formato científico
- %g especifica um double (ou float) no formato mais apropriado
- (%f ou %e)
- %s especifica uma cadeia de caracteres

Entrada de dados

- Função “scanf”:
 - captura valores fornecidos via teclado
- Exemplo:

```
int n;
```

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

*valor inteiro digitado pelo usuário é armazenado na
variável n*

Exemplo

```
#include <stdio.h>
main()
{
    int dia, mes, ano;
    float nota;
    char sexo;
    printf("Digite o dia, o mês e o ano: ");
    scanf("%d/%d/%d", &dia, &mes, &ano);
    printf("Digite a nota da prova mensal: ");
    scanf("%f", &nota);
    printf("Digite o sexo: (M ou F)");
    scanf("%c",&sexo);
    . . .
}
```


Exemplo 2

```
/* Programa que determina a média de três números dados */  
#include <stdio.h>  
main()  
{  
    float a, b, c, media;  
    printf("Digite três números");  
    scanf("%f %f %f", &a, &b, &c);  
    media = (a + b + c)/3;  
    printf("A media dos números %f , %f e %f é igual a %f ", a, b, c,  
    media);  
}
```