

Curso Profissional: Programador/a de Informática
PSD – 10.º ano: UFCD 0809 - Programação em C/C++ - fundamentos

Ficha de Trabalho 3

Ano letivo 21/22

IDENTIFICADORES, TIPOS DE DADOS BÁSICOS, VARIÁVEIS, CONSTANTES, OPERADORES numéricos e de atribuição

Identificadores – São os nomes atribuídos às entidades (como variáveis, constantes e funções) dum programa.

Regras na escolha de identificadores:

- Todas as que constam na ficha de trabalho n.º6, da UFCD 0804, da disciplina de CIAAS;
- Devem utilizar-se nomes até 32 caracteres (dependendo do compilador);
- Devemos usar minúsculas (normalmente a declaração de constantes é feita em maiúsculas).

Tipos de dados básicos:

Tipo	Descrição
void	Ausência de valor (0 bytes). É usado associado a funções que não devolvem qualquer valor.
char	N.ºs inteiros ou caracteres (1 byte). Pode conter apenas 1 único carácter, e o n.º de representações possíveis são 256.
int	N.º inteiros (2 bytes). Valor min. -32768, valor Max. 32767.
float	N.ºs reais (4 bytes). Para armazenar valores numéricos com parte fracionária.
double	N.ºs reais de dupla precisão (8 bytes). Reserva, para armazenar um valor, o dobro do n.º de bytes do tipo float.

Existem quatro modificadores de tipo int:

Tipo	Descrição
signed	Indica que o valor terá sinal positivo ou negativo.
unsigned	Indica que o valor não terá sinal, ou seja, é sempre positivo.
short	Indica um inteiro pequeno (2 bytes).
long	Indica um inteiro longo (4 bytes).

Variável – entidade que pode assumir diferentes valores ao longo da execução dum programa.

SINTAXE: **tipo var1 , var2, ..., varn ;**

Exemplo:

```
int i;  
char letra, letter;
```

- ❖ A declaração de variáveis tem que ser sempre realizada antes da sua utilização e antes de qualquer instrução.

```
main()
{
    declaração de variáveis
    instrucaoK;
    instrucaoX;
    ...
}
```

- ❖ As variáveis são sempre armazenadas em memória e são uma forma simples de referenciar posições de memória.
O tipo que lhes está associado indica o n.º de bytes que irão ser utilizados, para guardar o valor armazenado nessa variável.

Constante – Entidade que mantém o seu valor inalterado dentro do programa.

Devem ser definidas (de preferência) a seguir às diretivas do pré-processor - **# include**

SINTAXE: **const** tipo nome = valor ;

Exemplo: const char letra = 'a' ;

Diferença entre const e # define

- **const** faz parte das palavras reservadas da linguagem de programação C ;
- **# define** é uma diretiva do pré-processor em que o símbolo que se lhe segue vai ser substituído pelo valor que aparece a seguir a esse mesmo símbolo;
Exemplo: # define MAX 100
- Uma constante definida com a palavra **const** fica com o tipo que lhe foi atribuído na definição, quando definida com **# define**, o tipo da constante é aquele que está associado à componente valor;
- As constantes definidas com **# define**, normalmente são colocadas em maiúsculas.

Atribuição

SINTAXE: variável = expressão ;

Exemplo: num = 2*n ;

- ❖ Quando são escritas várias atribuições sucessivas, estas são realizadas (em C) da direita para a esquerda.

Exemplo: a = b = 5 ; equivale a a=5; b=5;

Operadores

Existem 5 operadores aritméticos normais (por normais entende-se que são operadores cuja função é igual noutras linguagens de programação).

Os operadores de atribuição, tal como o próprio nome indica, atribuem um valor a um operando.

Operadores aritméticos normais:

Operador	Nome	Formato	Descrição
+	Soma	op1 + op2	Calcula a soma dos operandos.
-	Subtração	op1 - op2	O segundo operando é subtraído ao primeiro operando.
*	Multiplicação	op1 * op2	Calcula a multiplicação dos operandos.
/	Divisão	op1 / op2	Calcula o quociente da divisão do 1º operando pelo 2º operando (<u>se os operandos forem ambos inteiros calcula a divisão inteira</u>)
%	Resto	op1 % op2	Calcula o resto da divisão do 1º operando pelo 2º operando.

Operadores de atribuição:

Operador	Nome	Formato	Descrição
=	Atribuição	op1 = op2	Atribui um valor a uma variável.
+=	Atribuição composta da soma	op1 += op2	O valor de op1 + op2 é calculado e atribuído a op1. Equivalente a op1 = op1 + op2.
-=	Atribuição composta da subtração	op1 -= op2	O valor de op1 - op2 é calculado e atribuído a op1. Equivalente a op1 = op1 - op2.
*=	Atribuição composta da multiplicação	op1 *= op2	O valor de op1 * op2 é calculado e atribuído a op1. Equivalente a op1 = op1 * op2.
/=	Atribuição composta da divisão	op1 /= op2	O valor de op1 / op2 é calculado e atribuído a op1. Equivalente a op1 = op1 / op2.
%=	Atribuição composta do resto	op1 %= op2	O valor de op1 % op2 é calculado e atribuído a op1. Equivalente a op1 = op1 % op2.

Exemplo:

```
Se distancia = 4, velocidade = 2
distancia = velocidade resultado: 2
distancia += velocidade resultado: 6
distancia -= velocidade resultado: 2
distancia *= velocidade resultado: 8
distancia /= velocidade resultado: 2
distancia %= velocidade resultado: 0
```

Precedências/prioridades (aumentam no sentido da seta):

*, /, %

+, -

=, +=, -=, *=, /=, %=



EXERCÍCIOS:

Resolve as seguintes expressões:

1- $x = -3 + 4 * 5 - 6$

2- $y = 3 + 4 \% 6 / 5$

3- $z = (7 + 6) \% 5 / 2$

4- para $x=1$; $y=2$; $z=3$

a. $z += -2 + 3$

b. $y *= x - 3 * z$

c. $x -= z \% y$

Valores de vírgula-flutuante

A representação em vírgula flutuante não é mais do que a notação científica utilizada nas máquinas de calcular e, contrariamente à representação em vírgula fixa, é feita através de um par de inteiros que representam respetivamente a *mantissa* **M** e o *expoente* **E**, de forma que para uma determinada *base* **b**, o seu valor é:

$$F = M \times b^E$$

Os tipos de dados float e double são tipos de dados de ponto flutuante.

Qualquer número X é dado pela expressão geral: $X = \pm M \times b^{\pm E}$, onde b é a base do sistema de numeração considerado (em decimal $b=10$, em binário $b=2$) □

Por exemplo, para $22,625_{(10)} \equiv 10110,101_{(2)}$, viria $<0,625$ em binário dá $\equiv 101_{(2)}$ **

$$22,625 \equiv 2,2625 \times 10^1$$

$$10110,101 \equiv 1,0110101 \times 10^{100}$$

(Note que $10_{(2)} \equiv 2_{(10)}$ e $100_{(2)} \equiv 4_{(10)}$)

Em binário fica: 00 01000000110101₂

A representação por vírgula flutuante utiliza-se quando se pretende aumentar a quantidade de números codificáveis à custa da perda de precisão (nº de dígitos mantidos na parte fracionária do número)

A tabela abaixo compara o intervalo, precisão, maior inteiro e tamanho:

Tipo	Intervalo	Precisão (Dígitos)	Maior Inteiro Exato	Tamanho
Float	$1.5 \times 10^{-45} / 3.4 \times 10^{38}$	7	2^{24}	4 bytes
Double	$5.0 \times 10^{-324} / 1.7 \times 10^{308}$	15-16	2^{53}	8 bytes
Decimal	$1.0 \times 10^{-28} / 7.9 \times 10^{28}$	28-29	2^{113}	16 bytes

Obs.: Maior inteiro exato significa o maior valor inteiro que pode ser representado sem perda de precisão. Estes valores devem ser mantidos em mente quando se converte entre tipos inteiros e ponto flutuante.

Para qualquer cálculo que envolva dinheiro ou finanças, o tipo Decimal deve ser sempre utilizado. Só este tipo tem a precisão adequada para evitar os erros críticos de arredondamento.

