Valores e Tipos

Programação I **2022/23**

Salvador Abreu
spa@uevora.pt
Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

Tipos

Conversão de tipos

Tipos

Tipos base

int

Inteiro que pode ser positivo ou negativo

float

Valor numérico com parte decimal

A melhor aproximação aos números reais

char

representa inteiros "pequenos" (1 byte)

Literais

Os tipos numéricos permitem exprimir uma constante, designada um «**literal**», i.e.

- um texto que
- representa um valor
- imutável
- desse tipo.

Por exemplo

```
(int) 123
(float) 123.456
(char) 'A'
```

Caracteres

Tipo char: literais tem caracteres especiais

da forma "**\???**" em que "****" é o backslash e ??? é interpretado de forma particular; alguns exemplos:

`\n`	mudança de linha
`\t`	tabulação (salta para a coluna próximo múltiplo de 8)
`\'`	uma plica
`\\`	o "backslash" em si

Literal de cadeia de caracteres (string)

Indicada entre "aspas"

Não é um tipo base no C (veremos isso depois)

Representação

int (char, short, int, long)

Complemento para 2 (positivo ou negativo)

Ocupa respetivamente 1, 2, 4 ou 8 bytes

float (float, double)

Vírgula flutuante (IEEE 754 - ver https://en.wikipedia.org/wiki/IEEE_754)

Ocupa respetivamente 4 ou 8 bytes

char

Inteiro que representa o código dum carácter: ASCII

Ocupa 1 byte (8 bits)

Código ASCII

ASCII (American Standard Code for Information Interchange)

7 bits

Extended ASCII

8 bits

Unicode

UTF-8

UTF-16

UTF-32

USASCII code chart

Bits $b_7 b_6 b_5$	000	001	010	011	100	101	110	111
b_4 b_3 b_2 b_1 Row ↓ Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Valores booleanos

Uma expressão com operadores relacionais origina um valor dito "booleano" (toma valores "verdade" ou "falso")

x == y → **verdade** se x for igual a y

Não confundir com a afetação

x != y → **verdade** se x for diferente de y

x < y → **verdade** se x for menor que y

x > y → **verdade** se x for maior que y

x <= y → **verdade** se x for menor ou igual a y

x >= y → **verdade** se x for maior ou igual a y

Em C não existe um tipo para valores booleanos

São usados valores numéricos

0 : considerado **falso**

≠ 0 : considerado **verdade**

Operadores lógicos

Operadores lógicos

&& (and)

|| (or)

! (not)

Tabela de verdade

a	b	a && b	a || b	!a
F	F	F	F	T
F	T	F	T	T
T	F	F	T	F
T	T	T	T	F

Avaliação de expressões lógicas

A avaliação é mínima (*short-circuit*)

Começa por avaliar a expressão da esquerda; avalia as seguintes **se necessário**

Comportamento

`a || b`

Se **`a == verdade`** então **`verdade`**, senão **`b`**

`a && b`

Se **`a == falso`** então **`falso`**, senão **`b`**

`! a`

Se **`a == falso`** então **`verdade`**, senão **`falso`**

Conversão de tipos

Conversão de tipos

Conversão implícita

Se a expressão envolver 2 tipos compatíveis a expressão fica com o tipo mais geral. p/ex

char → int

int → float

Nota: não dá para "descer" automaticamente na hierarquia de tipos (p/ex float -> short)

Explícita (type cast)

(tipo) valor

Type Cast

(float) expr_num

Converte expr para um valor real

(int) expr_num

maior inteiro inferior ou igual a expr (parte inteira do número)

(char) expr_int

Caráter correspondente a expr_int (código ASCII)

(int) expr_char

Código ASCII correspondente a expr_char

Exercício 1

Calcule o perímetro, a área e o volume

de uma circunferência, círculo e esfera (respetivamente) cujo raio é especificado pelo utilizador (na linha de comando).

o valor deverá ser impresso com 2 casas decimais

A saber

pi: 3.14159265

perimetro: 2 * pi * r

area: pi * r^2

volume: 4* pi * r^3 / 3

Perimetro, area e volume

```
MAIN() {
   float raio, perimetro, area, volume;
   float pi = 3.14159265;

   raio = atof (ARG1);

   perimetro = 2 * pi * raio;
   area = pi * raio * raio;
   volume = 4 * pi * raio * raio * raio / 3;

   printf( “perimetro = %.2f\n”, perimetro );
   printf( “area = %.2f\n”, area );
   printf( “volume = %.2f\n”, volume );

   return 0;
}
```

É possivel declarar várias variáveis na mesma instrução

É possivel declarar e atribuir valor na mesma instrução

Exercício 2

Escreva um programa que aceita um número real e o escreve de volta, com 3 casas decimais

ex: 1234.56789 → resultado: 1234.568

ex: 1234 → resultado: 1234.000

Número com 3 casas decimais

```
MAIN() {
    float x, y;
    int n;
    x = atof (ARG1);

    y = x *1000;
    n = (int) y;
    y = n / 1000.0;
    printf("y = %.3f", y);
    return 0;
}
```

n é convertido (implicitamente) para float antes da divisão