

# O Scanf e o Printf da linguagem C

## Programação III

Paul Crocker e Simão Melo de Sousa

15 de Março de 1999

### 1 printf

#### 1.1 sintaxe

Em que:

**format** é uma string constante (delimitada por aspas ") ou um apontador para uma string.

**lista\_de-expressões** é uma sequência de expressões separadas por vírgulas de tipo em acordo com o conteúdo de **format**.

`printf (format, lista_de-expressões)`

c	para o tipo char (serve também para short ou int)
d	para o tipo int (serve também para short ou char)
u	para o tipo unsigned int (serve também para unsigned char ou unsigned short)
ld	para o tipo long
lu	para o tipo unsigned long
f	para o tipo double ou para o tipo float quando o formato decimal é pretendido (formato xxx.yyy)
e	para o tipo double ou para o tipo float quando o formato científico é pretendido (formato (+/-)xxx.yyye(+/-)zzz)
s	para as sequências de caracteres.

Figura 1: Opções (precedidas pelo caracter %) mais usadas do printf

Além dessas opções, é possível formatar as saídas colocando valores numéricos entre o % e a opção desejada. Por defeito, os inteiros são mostrados com o número de caracteres necessário (sem espaço nem antes nem depois), os números de vírgula flutuante são mostrados com 6 dígitos depois da vírgula. printf é uma função que tem por efeito lateral o output. O valor devolvido pela função é o número efectivo de caracteres mandados para o output.

x	espaço mínimo a atribuir ao display do número na sua integralidade.
x.y	x = espaço mínimo a atribuir ao display do número na sua integralidade, y = espaço reservado para a parte decimal.
-	esta opção suplementar permite indicar que a formatação sera feita a esquerda (os eventuais espaços serão colocado a direita e não a esquerda)
*	especifica que o valor da formatação é fornecida por um dos parâmetros da lista_de-expressões como por exemplo em printf ("%8.*f", n, x)

Figura 2: Opções de formatação para o printf

### 2 scanf

#### 2.1 sintaxe

Em que:

**format** é uma string constante (delimitada por aspas ") ou um apontador para uma string.

**lista\_de-endereços** é uma sequência de endereços de variáveis (separadas por vírgulas), cujo tipo está em acordo com o conteúdo de **format**, que irão receber os valores introduzidos.

`scanf (format, lista_de-endereços)`

c	char
d	int
u	unsigned int
hd	short int
hu	unsigned short
ld	long int
lu	unsigned long
f ou e	float (independentemente da notação escolhida, decimal ou científica)
lf ou le	double (idem)
s	sequência de caracteres.

Figura 3: Opções (precedidas pelo caracter %) mais usadas do scanf

Um aspecto importante no funcionamento do scanf é o seu "assincronismo": o scanf não es sincronizado com o canal de input. O input passa por uma zona de "buffer" que é consultada pelo scanf. Este facto impõe um comportamento particular (e eventualmente esquisito para principiante) que terá de ser tomado devidamente em conta para um bom funcionamento d

"conversas" com o programa<sup>1</sup>.

Como para o `printf`, opções suplementares são fornecidas para facilitar (formatar) a introdução dos dados. Indicar que `n` máximo `x` dígitos do input serão tomados em conta para uma determinada variável é realizado colocando o número `x` entre `%` e a opção que designa o tipo da informação recolhida.

Para indicar que a introdução sucessiva de dois input para duas variáveis especificadas em `lista.de.endereços` pode ser separada por vários espaços é preciso fornecer o carácter de "separação" `^` ao `scanf` que será introduzidos entre os dois adequados `%` da opção `format`. Por exemplo `scanf("%i %i ^ %i %i", &inteir, &caracter, &separados, &espacos)` separados por pelo menos um espaço.

---

<sup>1</sup>O buffer não está necessariamente limpo (vazio) quando é evocado o `scanf`.