



## Fondamenti di Informatica

Laurea in  
Ingegneria Civile e Ingegneria per l'ambiente e il territorio

### Linguaggio C: Le basi

Stefano Cagnoni e Monica Mordonini

## Le funzioni di input/output

<stdio.h>

## Input / Output

- Il linguaggio C consente di definire *funzioni* aggiuntive raggruppabili eventualmente in *library*.
- Esiste in particolare una libreria standard (*standard library*) di funzioni; per rendere disponibili tali funzioni, occorre inserire nelle dichiarazioni globali del programma, la direttiva di precompilazione:

```
#include <stdio.h>
```
- La libreria standard comprende, tra le altre, le seguenti funzioni di input/output

Il linguaggio C

3

## Input / Output

- ```
printf (<stringa di formato>[,<lista di espressioni>])
```
- per visualizzare dati sullo *standard output*
  - <lista di espressioni>* ::= <espressione> {<espressione>}
  - sono espressioni il cui valore deve essere visualizzato
  - <stringa di formato>*: contiene i caratteri da visualizzare ed i simboli di formato per i valori delle espressioni da visualizzare:
    - %d per la visualizzazione di un numero intero
    - %f per la visualizzazione di un numero reale
    - %c per la visualizzazione di un carattere
    - %s per la visualizzazione di una stringa

Il linguaggio C

4

## Esempi

- ```
printf ("Questa è una frase di prova\n");
      □ Questa è una frase di prova

printf ("La somma di %d e %d risulta: %d\n", m, n,
       m+n);
      □ La somma di 3 e 2 risulta: 5
          (se la variabile m ha valore 3 e la variabile n ha valore 2)

printf ("Il codice ASCII %d\nviene visualizzato come
       %c\n", 61, 61)

      □ Il codice ASCII 61
          viene visualizzato come A
```

Il linguaggio C

5

## Input / Output

- ```
scanf (<stringa di formato> , <lista indirizzi
      variabili>)
```
- consente la lettura di dati dallo *standard input*
  - <lista di indirizzi di variabili>* ::=  
 <indirizzo> ::= &<variabile>
  - <stringa di formato>* (vedi printf)
  - Esempi  
scanf ("%d %f %c", &n, &x, &c);  
Ma  
scanf ("%s", nomestringa); (una variabile  
stringa contiene l'indirizzo del suo primo carattere)

Il linguaggio C

6

## Input / Output

### Attenzione

- scandf e printf hanno sintassi simili.
- Ma scanf è di input ed ha bisogno di sapere l'indirizzo della cella in cui voglio memorizzare il dato d'ingresso
- Quindi ogni variabile (scalare) deve essere preceduta dal simbolo &
- printf è una funzione di output ed ha bisogno di sapere il valore della variabile e quindi non serve &.

Il linguaggio C

7

## Esercizi

Scrivere un programma che prenda in ingresso un codice (un numero intero) e ne visualizzi il carattere (char) cui è associato nella tabella ASCII

Scrivere un programma che prenda in ingresso un carattere e ne visualizzi il codice (numero intero) a cui è associato nella tabella ASCII

Il linguaggio C

8

| Byte     | Cod. ASCII          | Byte     | Cod. ASCII | Byte     | Cod. ASCII | Byte     | Cod. ASCII | Byte | Cod. ASCII |
|----------|---------------------|----------|------------|----------|------------|----------|------------|------|------------|
| 00000000 | 0 Null              | 00000000 | 32 Space   | 00000000 | 64 @       | 00000000 | 96 ~       |      |            |
| 00000001 | 1 Start of heading  | 00000001 | 33 !       | 00000001 | 65 A       | 00000001 | 97 a       |      |            |
| 00000002 | 2 Start of text     | 00000002 | 34 "       | 00000002 | 66 B       | 00000002 | 98 b       |      |            |
| 00000003 | 3 End of text       | 00000003 | 35 #       | 00000003 | 67 C       | 00000003 | 99 c       |      |            |
| 00000004 | 4 End of transmit   | 00000004 | 36 \$      | 00000004 | 68 D       | 00000004 | 100 d      |      |            |
| 00000005 | 5 Enquiry           | 00000005 | 37 %       | 00000005 | 69 E       | 00000005 | 101 e      |      |            |
| 00000006 | 6 Acknowledge       | 00000006 | 38 &       | 00000006 | 70 F       | 00000006 | 102 f      |      |            |
| 00000007 | 7 Audit trail       | 00000007 | 39 ,       | 00000007 | 71 G       | 00000007 | 103 g      |      |            |
| 00000008 | 8 Backspace         | 00000008 | 40 (       | 00000008 | 72 H       | 00000008 | 104 h      |      |            |
| 00000009 | 9 Horizontal tab    | 00000009 | 41 )       | 00000009 | 73 I       | 00000009 | 105 i      |      |            |
| 00000010 | 10 Line feed        | 00000010 | 42 ^       | 00000010 | 74 J       | 00000010 | 106 j      |      |            |
| 00000011 | 11 Vertical tab     | 00000011 | 43 +       | 00000011 | 75 K       | 00000011 | 107 k      |      |            |
| 00000012 | 12 Form feed        | 00000012 | 44 ,       | 00000012 | 76 L       | 00000012 | 108 l      |      |            |
| 00000013 | 13 Carriage return  | 00000013 | 45 -       | 00000013 | 77 M       | 00000013 | 109 m      |      |            |
| 00000014 | 14 Shift out        | 00000014 | 46 =       | 00000014 | 78 N       | 00000014 | 110 n      |      |            |
| 00000015 | 15 Shift in         | 00000015 | 47 /       | 00000015 | 79 O       | 00000015 | 111 o      |      |            |
| 00000016 | 16 Data link escape | 00000016 | 48 0       | 00000016 | 80 P       | 00000016 | 112 p      |      |            |
| 00000017 | 17 Device control 1 | 00000017 | 49 1       | 00000017 | 81 Q       | 00000017 | 113 q      |      |            |
| 00000018 | 18 Device control 2 | 00000018 | 50 2       | 00000018 | 82 R       | 00000018 | 114 r      |      |            |
| 00000019 | 19 Device control 3 | 00000019 | 51 3       | 00000019 | 83 S       | 00000019 | 115 s      |      |            |
| 00000020 | 20 Device control 4 | 00000020 | 52 4       | 00000020 | 84 T       | 00000020 | 116 t      |      |            |
| 00000021 | 21 Nw. acknowledge  | 00000021 | 53 5       | 00000021 | 85 U       | 00000021 | 117 u      |      |            |
| 00000022 | 22 Synchronous idle | 00000022 | 54 6       | 00000022 | 86 V       | 00000022 | 118 v      |      |            |
| 00000023 | 23 End trans. block | 00000023 | 55 7       | 00000023 | 87 W       | 00000023 | 119 w      |      |            |
| 00000024 | 24 Cancel           | 00000024 | 56 8       | 00000024 | 88 X       | 00000024 | 120 x      |      |            |
| 00000025 | 25 End of medium    | 00000025 | 57 9       | 00000025 | 89 Y       | 00000025 | 121 y      |      |            |
| 00000026 | 26 Substitution     | 00000026 | 58 :       | 00000026 | 90 Z       | 00000026 | 122 z      |      |            |
| 00000027 | 27 Escape           | 00000027 | 59 =       | 00000027 | 91 [       | 00000027 | 123 [      |      |            |
| 00000028 | 28 Record separator | 00000028 | 60 <       | 00000028 | 92 \       | 00000028 | 124 \      |      |            |
| 00000029 | 29 Group separator  | 00000029 | 61 =       | 00000029 | 93 ]       | 00000029 | 125 ]      |      |            |
| 00000030 | 30 Record Separator | 00000030 | 62 >       | 00000030 | 94 ^       | 00000030 | 126 ~      |      |            |
| 00000031 | 31 Unit separator   | 00000031 | 63 ?       | 00000031 | 95 _       | 00000031 | 127 `      |      |            |

Il linguaggio C

9

## Strutture di controllo

If, while, for, ...

## Strutture di controllo

### Selezione if

if (<condizione>) <istruzioni> [else <istruzioni>]  
Se la condizione è vera esegue il primo blocco di istruzioni, altrimenti esegue il secondo (se specificato)

<istruzioni> ::= <istruzione> | '{' <istruzione> -> <istruzione> '}'

► esempio

```
if (x > 0)
    y = x;
else
    y = -x;
printf("dato %d\n", y);
```

falso => 0

vero => non 0

operatori di confronto

maggiore >

minore <

minore o uguale >=

uguale ==

diverso !=

operatori logicci

not!

and&&

diverso !=

Il linguaggio C

10

Il linguaggio C

11

## Esempio

```
/* Dati due numeri in ingresso, individua il maggiore*/
#include <stdio.h>
int main(){
    int num1,num2;

    printf("Immetti i due numeri:");
    scanf("%d%d",&num1,&num2);
    if (num1==num2)
        printf("I due numeri sono uguali");
    else
        if(num1>num2)printf("Il primo è maggiore");
        else printf("Il secondo è maggiore");
    return 0;
}
```

**Esercizio:** Fare il diagramma di flusso di questo problema

Il linguaggio C

13

## Strutture di controllo

### ■ Ciclo for

```
for(<istr.iniziale>; <condizione>; <istr. ciclica>)
{<istruzioni>}
```

Dopo avere eseguito l'istruzione iniziale, ripete l'esecuzione del blocco di istruzioni che segue finché la condizione resta vera (se è falsa la prima volta, può anche non eseguirlo mai), concludendo ogni ciclo con l'esecuzione dell'istruzione ciclica.

→ esempio: calcolo della somma dei primi 100 numeri naturali

```
#include <stdio.h>
int main(){
    int i, somma = 0;

    for ( i=1; i<=100; i=i+1)
        somma = somma+i; /* se c'è una sola istruzione si
                            possono tralasciare le graffe */
    printf ("La somma vale: %d\n", somma);
    return 0;
}
```

Il linguaggio C

14

## Strutture di controllo

### ■ Ciclo while

```
while (<condizione>) {<istruzioni>}
Esegue il blocco di istruzioni che segue, finché la condizione è
vera (come per il for, può anche non eseguirlo mai)
→ esempio: calcolo della somma dei primi 100 numeri naturali
#include <stdio.h>
int main(){
    int i=1;
    int somma = 0;

    while(i<=100) {
        somma = somma+i;
        i=i+1;
    }
    printf ("La somma vale: %d\n", somma);
    return 0;
}
```

Il linguaggio C

15

## Strutture di controllo

### ■ Ciclo do while

```
do <istruzioni> while (<condizione>)
```

Esegue (sempre, almeno una volta) le istruzioni comprese nel blocco che segue e le ripete finché la condizione è vera

→ esempio: calcolo della somma dei primi 100 numeri naturali

```
#include <stdio.h>
int main(){
    int i=1;
    int somma = 0;

    do {
        somma = somma+i;
        i=i+1;
    } while (i<=100)
    printf ("La somma vale: %d\n", somma);
    return 0;
}
```

Il linguaggio C

16

## Esercizi

Scrivere un programma che prenda in ingresso un intero N e calcoli il doppio della somma dei primi N numeri

Scrivere un programma che chieda in ingresso un numero finché non si inserisce un numero dispari. Quando questo avviene, il programma deve uscire scrivendo quanti numeri pari erano stati inseriti in precedenza.

Il linguaggio C

17