

11.8 Immissione ed emissione su stringa

Esistono altre due funzioni, `sscanf` e `sprintf`, che eseguono le medesime conversioni di `scanf` e `printf`, ma operando su stringa invece che su standard input e standard output. Il prototype di queste due funzioni è:

```
int sprintf(char *s, char *format, arg1, arg2, ...)
int sscanf(char *s, char *format, punt1, punt2, ...)
```

La funzione `sprintf` è comunemente usata per scrivere in un determinato formato variabili C dentro una stringa. La stringa in cui scrivere gli argomenti `arg1, arg2, ...`, secondo il formato specificato dalla stringa `format`, è il primo argomento della funzione, `s`. Per ciò che riguarda la specifica di formato valgono le medesime convenzioni di `printf`. Tipicamente `sprintf` è usata per preparare delle stringhe che saranno poi usate da altri programmi di visualizzazione più semplici che, per esempio, non utilizzano alcun meccanismo di formattazione dei dati ■. A titolo illustrativo si consideri il programma del Listato 11.2, nel quale il contenuto delle variabili `i` e `d` è copiato nella stringa `bufusc` che viene successivamente visualizzata da `printf`.

```
#include <stdio.h>

int    i = 80;
double d = 3.14546;
main()
{
    int  numusc;
    char bufusc[81];

    numusc = sprintf(bufusc, "Il valore di i = %d e \
il valore di d = %g\n", i, d);

    printf("sprintf() ha scritto %d caratteri e il \
buffer contiene:\n%s", numusc, bufusc);
}
```

Listato 11.2 Copia su stringa

La funzione `sscanf` ha lo scopo di leggere dei caratteri da una stringa, `s`, che possiamo chiamare buffer di caratteri, convertirli e memorizzarli nelle locazioni puntate da `punt1, punt2, ...` secondo il formato specificato dalla stringa `format`. Per ciò che riguarda la specifica di formato valgono le medesime convenzioni di `scanf`. Tipicamente `sscanf` è usata per la conversione di caratteri in valori. Generalmente si leggono le stringhe da decodificare, per esempio, con la funzione `gets`, e poi si estraggono dei valori dalla stringa con `sscanf`.

La funzione `sscanf` restituisce il numero di campi che sono stati letti, convertiti e assegnati a variabili con successo. Se la stringa dovesse finire prima della fine dell'operazione di lettura, `sscanf` ritornerebbe il valore costante EOF, definito in `stdio.h`. A titolo di esempio si consideri il programma del Listato 11.3, che accetta in ingresso assegnazioni di variabili secondo il formato `"nome = <valore>"`. La funzione `gets` brutalmente legge tutta la stringa corrispondente all'assegnazione e la funzione `sscanf` la scompone nei due elementi `nome` e `valore` del `nome`.

```
#include <stdio.h>

main()
{
    double valore;
    char buf[31], nome[31];

    printf("Inserire una variabile nel formato\
\n\"nome = <valore>\":");
    gets(buf);

    /* Con sscanf() si separano il nome dal valore */

    sscanf(buf, " %[^=] = %lf", nome, &valore);

    printf("La variabile %s vale %f\n", nome, valore);
}
```

```
}
```

Listato 11.3 Esempio di uso `sscanf`