

```

8  {
9  char sentence[ 80 ]; /* crea un vettore di caratteri */
10
11  printf( "Enter a line of text:\n" );
12
13  /* usa gets per leggere una linea di testo */
14  gets( sentence );
15
16  printf( "\nThe line printed backwards is:\n" );
17  reverse( sentence );
18
19  return 0; /* indica che il programma è terminato con successo */
20
21 } /* fine della funzione main */
22
23 /* i caratteri della stringa vengono visualizzati ricorsivamente
24    in ordine inverso */
25 void reverse( const char * const sPtr )
26 {
27     /* se viene raggiunta la fine della stringa */
28     if ( sPtr[ 0 ] == '\0' ) { /* caso base */
29         return;
30     } /* fine del ramo if */
31     else { /* se la fine della stringa non è stata raggiunta */
32         reverse( &sPtr[ 1 ] ); /* passo ricorsivo */
33
34         putchar( sPtr[ 0 ] ); /* usa putchar per visualizzare
35                                il carattere */
36     } /* fine del ramo else */
37 } /* fine della funzione reverse */

```

```

Enter a line of text;
Characters and Strings

```

```

The line printed backwards is:
sgnirtS dna sretcarahC

```

```

Enter a line of text:
able was I ere I saw elba

```

```

The line printed backwards is:
able was I ere I saw elba

```

**Figura 8.13** Usare gets e putchar

Il programma della Figura 8.14 utilizzerà le funzioni `getchar` e `puts` per leggere i caratteri dallo standard input, immagazzinarli nel vettore `sentence` e visualizzarlo come se fosse una stringa. La funzione `getchar` leggerà un carattere dallo standard input e lo restituirà

come un valore intero. La funzione `puts` riceverà come argomento una stringa (`char *`) e la visualizzerà facendola seguire da un carattere di newline.

Il programma smetterà di prendere in input i caratteri quando `getchar` avrà letto il newline immesso dall'utente alla fine della riga di testo. Un carattere nullo sarà accodato al vettore `sentence` (riga 19) in modo che possa essere trattato come una stringa. Quindi la funzione `puts` visualizzerà la stringa contenuta in `sentence`.

```

1  /* Fig. 8.14: fig08_14.c
2     Usare getchar and puts */
3  #include <stdio.h>
4
5  int main()
6  {
7     char c; /* variabile per memorizzare il carattere inserito
8              dall'utente */
9     char sentence[ 80 ] ; /* crea un vettore di caratteri */
10    int i = 0; /* inizializza il contatore i */
11
12    /* sollecita l'utente a inserire una linea di testo */
13    puts( "Enter a line of text:" );
14
15    /* usa getchar per leggere ogni singolo carattere */
16    while ( ( c = getchar() ) != '\n' ) {
17        sentence[ i++ ] = c;
18    } /* fine del comando while */
19
20    sentence[ i ] = '\0'; /* termina la stringa */
21
22    /* usa puts per visualizzare sentence */
23    puts( "\nThe line entered was:" );
24    puts( sentence );
25
26    return 0; /* indica che il programma è terminato con successo */
27 } /* fine della funzione main */

```

```

Enter a line of text:
This is a test.

```

```

The line entered was:
This is a test.

```

**Figura 8.14** Usare `getchar` e `puts`

Il programma della Figura 8.15 utilizzerà la funzione `sprintf` per riversare nel vettore di caratteri `s` i dati formattati. La funzione utilizza le stesse specifiche di conversione di `printf` (consultate il Capitolo 9, per una discussione dettagliata di tutte le potenzialità per la formattazione della stampa). Il programma riceverà in input un `int` e un `double` che dovranno essere formattati e stampati nel vettore `s`. Quest'ultimo sarà il primo argomento di `sprintf`.