

Esercizi

- 14.2 Scrivete un programma che calcoli il prodotto di una serie di interi passati alla funzione **product**, usando un elenco variabile di argomenti. Verificate la vostra funzione con diverse chiamate, ognuna con un numero differente di argomenti.
- 14.3 Scrivete un programma che visualizzi gli argomenti immessi sulla riga di comando di un programma.
- 14.4 Scrivete un programma che ordini un vettore di interi in modo ascendente o discendente. Il programma dovrà usare gli argomenti della riga di comando per passare **-a**, per un ordinamento ascendente, o **-d** per uno discendente. (Nota: in UNIX questo è il formato standard per il passaggio delle opzioni a un programma).
- 14.5 Scrivete un programma che inserisca uno spazio tra ogni carattere di un file. Il programma dovrà prima scrivere il contenuto del file da modificare in un archivio temporaneo, inserendo gli spazi tra i caratteri, e quindi dovrà ricopiare il file temporaneo su quello originale. Questa operazione dovrà sostituire i contenuti del file originale.
- 14.6 Consultate il manuale del vostro sistema per determinare quali segnali siano supportati dalla relativa libreria di gestione (**signal.h**). Scrivete un programma che contenga alcune gestioni dei segnali standard **SIGABRT** e **SIGINT**. Il programma dovrà controllare l'intercettazione di questi segnali richiamando la funzione **abort**, per generarne uno di tipo **SIGABRT**, e premendo **<ctrl> c**, per generarne uno di tipo **SIGINT**.
- 14.7 Scrivete un programma che allochi dinamicamente un vettore di interi. La dimensione del vettore dovrà essere immessa dalla tastiera. Gli elementi del vettore dovranno essere impostati con valori immessi dalla tastiera. Visualizzate i valori del vettore e quindi allocate nuovamente la sua memoria a metà del numero corrente di elementi. Visualizzate i valori rimasti nel vettore per confermare che corrispondano alla prima metà dei valori contenuti nel vettore originale.
- 14.8 Scrivete un programma che riceva come argomenti due nomi di file dalla riga di comando, legga uno per volta i caratteri dal primo e li scriva in ordine inverso nel secondo.
- 14.9 Scrivete un programma che usi l'istruzione **goto** per simulare una struttura di iterazione nidificata che visualizzi un quadrato di asterischi come il seguente:

```
* * * * *
*       *
*       *
*       *
*       *
* * * * *
```

Il programma dovrà usare soltanto le seguenti tre istruzioni **printf**:

```
printf("*");
printf(" ");
printf("\n");
```