

La presente esercitazione presuppone conoscenza delle seguenti parti del linguaggio C: *main, return, commenti, variabili, identificatori, tipi scalari, assegnazione, costanti letterali, printf (base), scanf (deprecata), operazioni aritmetiche, if...else, operatori relazionali, operatori Booleani, getchar, putchar, ciclo while, libreria ctype.h, const, #define, exit, programmazione a stati, gets, sscanf, stringhe, traduzione di algoritmi matematici; switch, break, #ifdef, math.h, cast, printf (completa), ciclo for; argc, argv; definizione di funzioni; file sequenziali, fopen, fclose, fgetc, fprintf; string.h; vettori, matrici; variabili static e globali.*

### Esercizio 1

Scrivere un programma in linguaggio C che dichiari un vettore da 20 numeri interi e ne inizializzi tutti i valori con i primi 20 numeri interi positivi pari. In seguito il programma deve stampare in ordine tutti i numeri del vettore sulla stessa riga separati da una virgola (,). Infine, stampare il vettore in ordine inverso.

### Esercizio 2

Scrivere un programma in linguaggio C che legga una serie di numeri interi dallo standard input in quantità non superiore a 100. L'introduzione dei dati termina quando si incontra il numero zero. Il programma deve quindi richiedere all'utente l'operazione da svolgere sui dati introdotti: col carattere N si richiede di conoscere il numero di elementi introdotti, S ne calcola la somma, M ne calcola la media, A ne calcola il minimo e Z il massimo. Le operazioni terminano quando l'utente introduce il carattere Q (per Quit).

### Esercizio 3

Scrivere un programma in linguaggio C che legga da un file i numeri interi (massimo 100) contenuti in esso, uno per riga. Il nome del file viene passato come primo parametro sulla linea di comando.

Successivamente il programma riceve su standard input dei numeri interi, uno per riga, e per ciascun numero introdotto deve indicare se esso è presente o meno fra i valori contenuti nel file.

Ad esempio, se il programma (chiamato `cerca_num.c`) viene attivato nel seguente modo:

```
cerca_num dati.txt
```

e se il file `dati.txt` contiene le seguenti righe:

```
1
6
28
-10
103
-25
```

e se il programma riceve il seguente input:

```
10
25
-10
6
100
```

il programma genera il seguente output:

```
10 : assente
25 : assente
-10 : presente
6 : presente
-25 : presente
```

#### **Esercizio 4**

Scrivere un programma in linguaggio C che dichiari e inizializzi una matrice come variabile globale di  $5 \times 5$  numeri interi. Il programma deve leggere dallo standard input numeri interi (finché ce ne sono) indicando se i valori letti sono contenuti o meno nella matrice; in caso positivo deve stampare le coordinate dove si trova la prima occorrenza del numero letto.

Per esempio, se il programma (chiamato `cerca_mat.c`) definisce la matrice:

```
12 20 23 34 21
27 15 35 55 110
10 92 21 40 83
81 64 99 86 1
13 49 80 87 100
```

e se il programma riceve il seguente input:

```
2
12
13
25
99
100
```

il programma genera il seguente output:

```
2 : NO
12 : SI (1,1)
13 : SI (5,1)
25 : NO
99 : SI (4,3)
100 : SI (5,5)
```