

## 4. ESERCIZI ancora iterazione, lettura dei caratteri, array

### ESERCIZIO 4.1

Si modifichi il programma dell'esercizio 3.11 utilizzando caratteri per rappresentare la risposta dell'utente alla domanda che operazione vuoi effettuare. Si sperimenti la soluzione utilizzando sia la funzione `getchar()` che la funzione `scanf` con la soppressione dell'input.

### ESERCIZIO 4.2

Si scriva un programma C che legge dall'input un carattere terminatore (stabilito ogni volta dall'utente) e legge e stampa una sequenza di caratteri terminando appena incontra il carattere terminatore. Si utilizzino le funzioni `getchar` e `putchar`. Si esegua il programma digitando tutti i caratteri e poi premendo invio e digitando invece un carattere per volta premendo invio dopo ogni carattere.

### ESERCIZIO 4.3

Si scriva un programma C che legge dall'input un intero positivo in decimale, rappresentabile con un byte e stampa il numero corrispondente in base 2.

### ESERCIZIO 4.4

Si scriva un programma C che legge dall'input una sequenza di caratteri terminando la lettura appena viene letto un qualunque carattere diverso da '0' e '1'. Interpretando la sequenza di caratteri letti come un numero intero senza segno, in base 2, il programma deve convertire il numero in base 10 e stamparlo.

### ESERCIZIO 4.5

Si scriva un programma C che legge dall'input  $k$  caratteri e li memorizza in una array A. Successivamente legge dall'input un carattere X e stampa carattere contenuto oppure carattere non contenuto, se in A esiste almeno un'occorrenza di X.

### ESERCIZIO 4.6

Si scriva un programma C che legge dall'input  $k$  caratteri e li memorizza in una array X. Successivamente legge dall'input un carattere X e stampa il numero di occorrenze del carattere X nell'array A.

### ESERCIZIO 4.7

Si scriva un programma C che legge dall'input  $k$  interi e li memorizza in una array  $a$  ed effettua lo shift a sinistra degli elementi, ovvero per  $i \in [0, k - 2]$  ( $a[i] = a[i + 1]$ ). In questo modo  $a[k - 1]$  comparirà 2 volte in posizione  $k-1$  e  $k-2$ , mentre il valore in  $a[0]$  verrà perso.

#### ESERCIZIO 4.7.1

Variante dell'esercizio 4.7: shift circolare a sinistra, l'elemento in posizione 0 viene memorizzato in posizione  $k-1$ .

### ESERCIZIO 4.8

Si scriva un programma C che legge dall'input  $k$  interi e li memorizza in una array  $a$  ed effettua lo shift a destra degli elementi, ovvero per  $i \in [k - 1, 1]$  ( $a[i] = a[i - 1]$ ). In questo modo  $a[0]$  comparirà 2 volte in posizione 0 e 1, mentre il valore in  $a[k - 1]$  verrà perso.

#### ESERCIZIO 4.8.1

Variante dell'esercizio 4.7: shift circolare a destra, l'elemento in posizione  $k-1$  viene memorizzato in posizione 0.

### ESERCIZIO 4.8

#### ESERCIZIO 4.8.1

### ESERCIZIO 4.9

Si scriva un programma C che legge dall'input 2 sequenze binarie (costituita da solo 0 e 1) di 8 caratteri e interpretandole come 2 interi con segno, in base 2, calcola la loro somma e la stampa.