

## Capitolo 4 - Caratteri e Stringhe

### Caratteri

#### 0. Carattere successivo

Scrivi un programma che richiede all'utente un carattere e mostra il carattere successivo del codice ASCII.

---

#### 1. Lettera, cifra o altro?

Scrivi un programma che richiede all'utente un carattere ASCII standard (7 bit) e comunica se il carattere inserito è una lettera, una cifra o "altro". Nel caso in cui il carattere sia una lettera, il programma deve anche dire se si tratta di una vocale o di una consonante.

---

#### 2. Da codice ASCII a carattere

Scrivi un programma che richiede all'utente un numero intero compreso tra 0 e 255 e visualizza il carattere corrispondente (secondo il codice ASCII).

---

#### 3. Da carattere a codice ASCII

Scrivi un programma che richiede all'utente un carattere e visualizza il codice ASCII corrispondente.

---

#### 4. Conversione minuscolo/MAIUSCOLO

Scrivi un programma che richiede all'utente una lettera minuscola e mostra la corrispondente lettera MAIUSCOLA. Il programma deve verificare che la lettera digitata sia una minuscola (in caso contrario deve richiedere di nuovo la lettera).

---

#### 5. Stampa del codice ASCII

Scrivi un programma che stampa tutti i caratteri del codice ASCII (con relativo codice).

---

#### 6. Sequenza di lettere terminata da punto

Scrivi un programma che richiede all'utente una sequenza di caratteri terminata dal carattere "punto" (cioè il programma continua a leggere nuovi caratteri finché non viene inserito il carattere '.'). Al termine dell'inserimento, il programma deve mostrare il numero di vocali e il

numero di consonanti inserite.

Esempio INPUT: prova. OUTPUT: vocali=2 consonanti=3

---

## 7. Lettera minima e massima

Scrivi un programma che richiede all'utente una sequenza di caratteri (l'inserimento continua finchè l'utente digita la lettera 's' come risposta alla domanda "vuoi inserire un altro carattere?"). Al termine dell'inserimento, il programma mostra la lettera più piccola e più grande tra quelle inserite. Esempio:

```
Inserisci un carattere: C
Vuoi inserire un altro carattere? s
Inserisci un carattere: B
Vuoi inserire un altro carattere? s
Inserisci un carattere: W
Vuoi inserire un altro carattere? n
Il carattere minore e' B e quello maggiore e' W
```

---

## Stringhe

## 8. Messaggi cifrati (cifrario di Cesare)

Un semplice metodo crittografico per cifrare messaggi, già noto ai tempi di Giulio Cesare (100 a.C), consiste nel traslare ogni lettera di K posizioni in avanti nell'alfabeto. Ad esempio, se K=3, il messaggio "ciao" diventa "fldr". Nell'esempio precedente: il messaggio "ciao" è detto testo in chiaro, "fldr" è il testo cifrato, e il valore K è detto chiave di cifratura. Scrivi un programma che richiede all'utente un messaggio e un numero intero K. Il programma deve mostrare il messaggio cifrato, che si ottiene sostituendo ogni carattere del messaggio originale con il carattere che nel codice ASCII si trova a K posizioni di distanza.

---

## 9. Cifratura con password

Il cifrario di Cesare è facilmente decifrabile perchè ogni lettera del testo è traslata di una stessa quantità costante (k). Un metodo di cifratura più sicuro consiste nel traslare ogni lettera di una quantità variabile, che dipende da una password alfanumerica. Esempio:

Testo in chiaro: "ALICE I LOVE U"

Password: "GATTO"

Procedimento:

A	L	I	C	E		I		L	O	V	E		U
G	A	T	T	O	G	A	T	T	O	G	A	T	T
H	M	J	W	T	'	J	4	'	^	J	F	4	i

In pratica:

- Si ricopia la password sotto il testo (ripetendola finchè necessario per coprire tutta la lunghezza del testo)
- La prima lettera da cifrare ('A') è traslata di una quantità che dipende dalla prima lettera della password (ovvero 'G'). Poichè 'G' occupa la posizione 7 dell'alfabeto italiano, la lettera 'A' è traslata in avanti di 7 posizioni e quindi diventa 'H'.
- La seconda lettera del messaggio ('L') è cifrata usando la seconda lettera della password ('A'). Poichè 'A' occupa la posizione 1 nell'alfabeto italiano, la lettera 'L' è traslata in avanti di una posizione e diventa 'M'.
- e così via fino alla fine.

Per decifrare un messaggio, conoscendo la password, occorre ripetere il procedimento traslando all'indietro anzichè in avanti.

Scrivi un programma che consente di cifrare un testo con il procedimento descritto. Nota: Per semplicità supponi che password e testo da cifrare contengano solo lettere maiuscole.

---

## 10. Parola più lunga e più corta

Scrivi un programma che richiede all'utente N stringhe (valore di N scelto dall'utente) e mostra la stringa più lunga e la più corta fra quelle inserite. Esempio:

```
Quante parole vuoi inserire? 3
Inserisci la parola: mela
Inserisci la parola: lamponi
Inserisci la parola: uva
La parola di lunghezza minima è uva (lunghezza = 3)
La parola di lunghezza massima è lamponi (lunghezza = 7)
```

## 11. Prima e ultima parola in ordine alfabetico

Scrivi un programma che richiede all'utente N stringhe (valore di N scelto dall'utente) e mostra la prima e l'ultima parola secondo l'ordinamento alfabetico. Esempio: Quante parole vuoi inserire? 3

```
Inserisci la parola: lamponi Inserisci la parola: ananas Inserisci la
parola: uva La prima parola, in ordine alfabetico, è ananas L'ultima
parola, in ordine alfabetico, è uva
```

## 12. Prime 6 lettere del codice fiscale

I primi 6 caratteri del codice fiscale italiano si ottengono tramite il seguente algoritmo:

**Prime 3 lettere:** si prendono le prime 3 consonanti del cognome. Se le consonanti sono insufficienti, si prelevano anche le vocali, nell'ordine in cui compaiono, ma sempre dopo le consonanti (esempio: Rosi diventa RSO). Nel caso in cui un cognome abbia meno di tre lettere, la parte di codice viene completata aggiungendo la lettera X (esempio: Po diventa POX).

**Successive 3 lettere:** se il nome contiene 4 o più consonanti, si scelgono la prima, la terza e la quarta (per esempio: Gianfranco diventa GFR), altrimenti le prime tre in ordine ( esempio: Tiziana diventa TZN). Se il nome non ha consonanti a sufficienza, si prendono anche le vocali; in ogni caso le vocali vengono riportate dopo le consonanti ( esempio: Luca diventa LCU). Nel caso in cui il nome abbia meno di tre lettere la parte di codice viene completata aggiungendo la lettera X.

Scrivi un programma che richiede cognome e nome e mostra le prime 6 lettere del codice fiscale.

---

## Stringhe e funzioni

### 13. Quante parole?

Scrivi un programma che richiede una frase e comunica quante parole contiene. Esempio INPUT: "Quel ramo del lago di como" OUTPUT: 6.

Risolvi l'esercizio definendo la funzione `contaParole` che riceve una stringa e restituisce il numero di parole.

---

### 14. Parole palindrome

Si dicono palindrome le parole o frasi che rimangono uguali se lette "al contrario". Esempi di palindrome: "ANNA", "KAYAK", "ITOPINONAVEVANONIPOTI" . Scrivi una funzione che riceve come argomento una stringa, e restituisce *true* se la stringa e' palindroma, e *false* se non lo è.

---

### 15. Conversione di una stringa in maiuscolo (funzione)

Scrivi una funzione che riceve una stringa e la converte in maiuscolo. La funzione deve avere il seguente prototipo: `void maiuscolo(char* s);` Attenzione: la conversione in maiuscolo deve modificare esclusivamente le lettere dalla a alla z, e nessun altro carattere ( spazi, cifre, segni di punteggiatura , ecc.).

Esempio:

Stringa prima della conversione: "via Nazionale, 12/b"

Stringa dopo la conversione: "VIA NAZIONALE, 12/B" .

---

### 16. Funzione "sostituisci carattere"

Scrivi una funzione con tipo di ritorno `void` che riceve come argomento una stringa `s` e due caratteri (`c1` e `c2`). La funzione deve modificare la stringa `s` sostituendo ogni occorrenza del carattere `c1` con il carattere `c2`.

Esempio:

INPUT: `s="altalena" c1='a' c2='b'`

OUTPUT: `s="bltblenb"`

---

### 17. Funzione "Carattere più frequente"

Scrivi una funzione che riceve una stringa e restituisce il carattere che compare con maggior frequenza al suo interno (se due o più caratteri compaiono "a pari merito", restituirne uno qualsiasi fra questi). Esempio: INPUT: "aba bbc ba" OUTPUT: 'b'  
perchè 'b' è il carattere che compare il maggior numero di volte.

---

## 18. Ricerca della sequenza "ABC"

Scrivi una funzione che riceve una stringa e restituisce il primo indice in cui viene trovata la sequenza di lettere "ABC". Se la sequenza non è presente la funzione deve restituire -1. Esempio:  
INPUT: "BABBCABCABC"  
OUTPUT: 5  
perchè all'indice 5 inizia per la prima volta la sequenza ABC.

---