

4 Stringhe e Matrici

Stringhe Le stringhe di caratteri sono gestite in C come dei vettori di `char` con alla fine un “tappo” dato dal carattere `'\0'`. E' possibile acquisire un'intera stringa di caratteri in una sola istruzione grazie all'istruzione `scanf('%s', variabile);` oppure `gets(variabile)`. Entrambi devono essere seguiti dall'istruzione `fflush(stdin);` che serve ad evitare errori nella successiva acquisizione di stringhe. Mentre l'istruzione `scanf('%s', variabile);` ferma la sua acquisizione al primo spazio, l'istruzione `gets(variabile)` inserisce nella variabile tutto l'input fino al primo invio.

Poichè le stringhe necessitano di un tappo, ogni volta che ne dichiaro una devo aggiungere un elemento per il tappo, ossia se voglio una stringa lunga `N_MAX` dovrò dichiarare un array di `N_MAX+1` elementi.

Esiste una libreria di C che gestisce le stringhe `string.h` essa ci permette di:

- ottenere la lunghezza di una stringa con la funzione `strlen(stringa);`
- confrontare due stringhe `strcmp(stringa1, stringa2)` e restituisce zero se sono uguali, un numero negativo se `stringa1` viene prima in ordine lessicografico di `stringa2` e un numero positivo se `stringa2` viene prima in ordine lessicografico di `stringa1` (se i caratteri sono tutti maiuscoli o tutti minuscoli);
- copiare una stringa `stringa2` in un'altra stringa `stringa1` con la funzione `strcpy(stringa1, stringa2);`
- concatenare alla stringa `stringa1` la stringa `stringa2` con la funzione `strcat(stringa1, stringa2);`

Matrici Le matrici in C vengono definite come vettori di vettori. Ad esempio per dichiarare una matrice bidimensionale dovremo scrivere:

```
#define DIM1 n_righe
#define DIM2 n_colonne

void main(){

tipo_variabile nome_variabile[DIM1][DIM2];

istruzioni;
```

```
}
```

dove `n_righe` e `n_colonne` sono degli interi che specificano il numero di righe e colonne della matrice. Nel caso di dimensioni superiori a 2, dovremo mettere tante parentesi quadre e specificare le grandezze massime per ogni dimensione.

4.1 Esercizi

Esercizio 4.1

Date due stringhe immesse dall'utente, implementare la stessa funzionalità della `strcat()` senza utilizzare la funzione `strlen()`. Considerare delle stringhe in ingresso di al massimo 20 caratteri.

Esercizio 4.2

1. Scrivere un programma che determini se una frase acquisita da tastiera è palindroma. Controllare se sono palindrome le seguenti frasi (senza spazi e maiuscole):
 - Eri un nano non annuire
 - Ad una vera pia donna dei simili fili misi e annodai: pareva nuda
 - O mordo tua nuora, o ari un autodromo
 - Occorre portar aratro per Rocco
2. (Bonus) Modificare il programma precedente per considerare stringhe in input contenenti anche caratteri speciali e spazi.

Esercizio 4.3

Implementare la seguente variante del cifrario di Cesare¹. Dopo aver acquisito un messaggio di massimo 160 caratteri (alfabetici minuscoli), il programma dovrà chiedere all'utente una chiave (numero intero K , $1 < K < 25$).

Il programma stamperà il messaggio cifrato, ottenuto traslando ogni lettera di K posizioni in avanti (dove K è la chiave).

¹http://it.wikipedia.org/wiki/Cifrario_di_Cesare

Dopo aver stampato il messaggio cifrato, il programma chiede all'utente di inserire un messaggio (che si assume sia stato cifrato con lo stesso algoritmo di cui sopra). Tale messaggio sarà dunque decifrato dal programma, il quale dovrà svolgere l'operazione inversa.

Infine, il messaggio decifrato sarà stampato a video.

Suggerimento: i caratteri minuscoli dalla a alla z corrispondono nella tabella ASCII alle cifre intere dalla 97 alla 122.

Esercizio 4.4

1. Scrivere un programma che converta una stringa in alfabeto farfallino. Nell'alfabeto farfallino, ogni vocale è raddoppiata, e tra le due vocali è inserita la lettera 'f'. Ad esempio, "ciao" diventa "cifafofo".
2. (bonus) Scrivere il medesimo programma utilizzando soltanto una variabile per memorizzare la stringa tradotta (ovvero senza copiare la traduzione in una nuova variabile).

Esercizio 4.5

Scrivere un programma che considera solo le lettere maiuscole e minuscole di una stringa letta da tastiera (di dimensione massima 256 elementi). Il programma dovrà calcolare le occorrenze di ogni carattere nella stringa. Le occorrenze saranno poi stampate a video a video come istogramma con gli asterischi nel seguente modo:

```
str = Ciao Come Stai? Non c'c' male GRAZIE! Mi trovo molto bene in questa citta'.
a |
b | *
c | **
d |
e | *****
f |
g |
h |
i | *****
j |
k |
l | **
m | ***
n | ***
o | *****
p |
q | *
r | *
s | *
t | *****
u | *
v | *
w |
x |
```

```
Y |
A |
B |
C | **
D |
E | *
F |
G | *
H |
I | *
J |
K |
L |
M | *
N | *
O |
P |
Q |
R | *
S | *
T |
U |
V |
W |
X |
Y |
```

Esercizio 4.6

Scrivere un programma che determini se due parole acquisite da tastiera sono una l'anagramma dell'altra.

Esercizio 4.7

Scrivere un programma che chieda all'utente la dimensione effettiva di una matrice quadrata, che dovrà poi essere acquisita dal programma.

Successivamente, il programma dovrà controllare se la matrice è diagonale (i.e., nessuno zero sulla diagonale principale e solo zeri fuori da essa).

Estendere il programma per controllare anche se la matrice inserita è simmetrica.

Esercizio 4.8

Scrivere un programma che inizializzi una matrice 10×10 con i valori della tavola pitagorica.

Dopodiché, il programma dovrà stampare la matrice seguendo un percorso a spirale in senso orario, a partire dalla cella $m[0][0]$.

Per semplicità si può assumere la matrice quadrata.

Esercizio 4.9

(TdE Febbraio 2012, variante) Data m una matrice di dimensione $n \times n$ (costante simbolica) di numeri interi nell'intervallo $[0, 9]$:

1. si scriva un frammento di programma in linguaggio C (con le relative dichiarazioni di variabili necessarie al corretto funzionamento), che trovi il numero più frequente (si ipotizzi che tale numero sia unico).
2. si scriva un frammento che memorizzi in un vettore e stampi a schermo tutti i numeri presenti nella matrice con valore minore del valore più frequente e successivamente memorizzi in un altro vettore tutti quelli con valore maggiore di quello più frequente e li stampi;
3. aggiungere al programma di cui sopra un frammento di codice che legga tutti e soli i valori memorizzati nel secondo dei due vettori e stampi a video se i valori che vi sono contenuti sono o meno monotoni crescenti, ovvero se per ogni i si ha $a_i \leq a_{i+1}$.

Esercizio 4.10

Scrivere un programma che acquisisca una matrice di interi positivi con `DIM_R` righe e `DIM_C` colonne. Il programma deve stampare l'indice della riga della matrice con somma massima.