



# ESERCITAZIONE 4

FONDAMENTI DI INFORMATICA

8 novembre 2019

# Matrici



# *Esercizio 1*

## **Somma di matrici**

Scrivere un programma che chiede all'utente i 25 x 2 elementi di due matrici 5x5. Salva in una terza matrice il valore della somma delle due matrici e la stampa a video.

# *Esercizio 2*

## **Crivello di Eratostene**

Il Crivello di Eratostene è una tecnica per calcolare i numeri primi. Si disegna una tabella 10x10 con i numeri da 1 a 100 ordinati. I numeri possono essere liberi o cancellati. Inizialmente tutti i numeri sono liberi, eccetto l'1 che è cancellato (non essendo numero primo, per definizione).

Si sceglie il primo numero libero (il 2, al primo giro) e si cancellano tutti i suoi multipli (4,6,8,...).

Si sceglie il successivo numero libero (questa volta il 3) e si cancellano tutti i suoi multipli che non siano già stati cancellati (9,15,21,...).

Si procede in questo modo fino alla fine della tabella.

Alla fine i numeri non cancellati saranno tutti e soli i numeri primi tra 0 e 100.

Implementare il Crivello di Eratostene usando una matrice 10x10 e stampare la matrice risultante al termine dell'esecuzione in forma tabellare.

# Struct

## Dichiarazione

```
typedef struct {  
    tipo1 campo1;  
    tipo2 campo2;  
    tipo3 campo3;  
    ...  
} nome_struct;
```

## Uso

```
nome_struct s;  
tipo1 t;  
s.campo1 = v1;  
s.campo2 = v2;  
...  
t = s.campo1;
```

# *Esercizio 3*

## **Frazioni**

Definire un tipo di dato strutturato per rappresentare una frazione in termini di numeratore e denominatore (due numeri interi).

Scrivere un programma che acquisisce due frazioni ed esegue la somma.

Il programma deve in seguito semplificare il risultato dell'operazione rappresentando l'eventuale segno meno nel numeratore.

# Stringhe



# STRINGHE

Sintassi

Array di `char` terminati da `'\0'`.

`char stringa[8];` contiene fino a 7 caratteri

```
scanf("%s", stringa);  
printf("%s", stringa);  
gets(stringa);
```

Lettura stringa terminata da blank  
Stampa stringa  
Lettura stringa terminata da `'\n'`

```
strlen(stringa);  
strcmp(stringa1, stringa2);  
strcpy(dest, src);  
strcat(dest, src);
```

Lunghezza stringa  
Compara due stringhe (ordine lessicografico)  
Copia src in dest  
Concatena src a dest



# *Esercizio 4*

## **Reverse**

Scrivere un programma che data una parola in ingresso, inverta i caratteri della parola e la stampi a video.

# *Esercizio 5*

## **Voto più alto esame**

Si scriva un programma in linguaggio C che mostri a video il nome dello studente che ha ottenuto il punteggio più alto in un esame.

Il programma dovrà leggere il numero di studenti iscritti  $N$  (con  $N < 100$ ), il nome (con un numero di caratteri  $< 20$ ) e il voto di ciascuno studente.

# *Esercizio 6*

## **Durata CD**

Specificare le strutture dati C necessarie ad un programma per la gestione di un archivio di CD musicali (al massimo 100). Ogni CD è caratterizzato da un titolo e un autore (entrambe stringhe di al massimo 30 caratteri), un anno di pubblicazione, il numero di tracce, la durata complessiva (rappresentata in termini di ore, minuti e secondi); inoltre si vuole memorizzare anche il prezzo del disco ed il numero di copie disponibili per la vendita.

Scrivere poi un programma che acquisisce un numero  $n$  di CD e visualizza il CD con durata massima.

# *Esercizio 7*

## **Ricerca parola**

Si scriva un programma in linguaggio C che riceva in ingresso due parole inserite da tastiera.

Si consideri che ciascuna parola può contenere al massimo 30 caratteri. Il programma deve verificare se la seconda parola inserita è contenuta almeno una volta all'interno della prima parola (ossia se la seconda parola è una sottostringa della prima parola).

# *Esercizio 8*

## **Sostituzione parola**

Si scriva un programma in linguaggio C che riceva in ingresso due parole inserite da tastiera.

Si consideri che ciascuna parola può contenere al massimo 30 caratteri. Il programma deve sostituire ogni occorrenza della seconda parola nella prima parola con una sequenza di caratteri '\*'.

Ad esempio, inserite le parole:

```
poliabcabcpolie poli
```

il programma deve visualizzare la parola

```
****abcabc****
```

# *Esercizio 12 – a casa*

## **Sostituisci carattere**

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase è terminata dall'introduzione del carattere di invio e contiene complessivamente al più 100 caratteri. Il programma deve svolgere le seguenti operazioni:

- visualizzare la frase inserita
- costruire una nuova frase in cui tutte le occorrenze del carattere '?' sono sostituite con il carattere di ritorno di linea '\n'. Il programma deve memorizzare la nuova frase in una opportuna variabile
- visualizzare la nuova frase.