

Esercizio 1

Restituire la somma di un array di interi in maniera ricorsiva

Esercizio 2

Creare una funzione ricorsiva che restituisca 1 se un array di interi è ordinato in maniera crescente, 0 altrimenti.

Esercizio 3

Creare una funzione ricorsiva che restituisca 1 se un numero passato per parametro è presente in un array di interi, 0 altrimenti.

Esercizio 4

Scrivere una funzione **ricorsiva** che permetta di convertire un numero da una base a un'altra. Questo, per qualsiasi base di partenza e destinazione (scelte dall'utente).

Esercizio 5

Si scriva un programma che, considerando una lista di interi:

- Ne permetta l'inizializzazione
- Permetta l'inserimento di un nuovo elemento (in coda)
- Calcoli ricorsivamente la somma di tutti gli elementi

Esercizio 6

Realizzare una lista di studenti che abbia come campi nome, cognome, anno di nascita e media esami.

Si realizzino funzioni (bonus se ricorsive) che permettano di:

- Inserire uno studente all'inizio della lista
- Inserire uno studente alla fine della lista
- Inserire uno studente in posizione arbitraria nella lista
- Ordinare la lista in base ai voti degli studenti
- Restituire gli studenti nati nel 1992

Esercizio 7

Si scriva un programma che utilizzi un'evoluzione del tipo stringa che chiameremo *stringa-lista*.

Sappiamo che in C una stringa è un array di caratteri, il nostro programma utilizzerà *stringa-lista* come lista di caratteri.

Permetterà di eseguire le seguenti operazioni:

- Inizializzazione di una *stringa-lista* a partire da una stringa standard (array di caratteri)
- Visualizzazione della *stringa-lista*
- Calcolo della lunghezza della *stringa-lista*
- Calcolo, in maniera ricorsiva, del numero di vocali
- Ricerca di una stringa (array di caratteri) all'interno della *stringa-lista*
- Inserimento di un'intera parola in input (data come array di caratteri) all'interno della *stringa-lista* nella posizione specificata dall'utente

Esercizio 1

Si scriva un programma C che definisca una struttura che contenga i seguenti dati:

- Nome (stringa di lunghezza 20)
- Cognome (stringa di lunghezza 20)
- Sesso (stringa di lunghezza 10)
- NumeroVoli (int)

Li si memorizzi in un vettore di strutture "manager".

Implementare le funzioni seguenti attraverso un menu (switch/case).

Scrivere tre funzioni per:

- Inserire un nuovo manager nell'array
- Stampare a terminale il nome e il cognome dei manager che hanno effettuato un numero di voli maggiori di 20.
- Controllare chi tra i manager donna e i manager uomini ha una media complessiva di voli più alta.

Esercizio 2

Partendo dalla struttura “laureato” così definita:

- Cognome (stringa di lunghezza 25);
- Nome (stringa di lunghezza 25);
- Corso di laurea (stringa di lunghezza 25)
- Voto di laurea (intero)
- Lode (intero: 0 senza lode, 1 con lode)
- Numero anni di iscrizione (intero: 3 triennale, 5 specialistica)

scrivere un programma che permetta di gestire attraverso un array, tutti gli studenti laureati in una determinata sessione di laurea.

Il programma presenta un menù che permette di (ogni voce del menu va eseguita per mezzo di una funzione):

1. Inserire i valori all'interno dell'array.
2. Dato un voto soglia *vsoglia* e un numero di anni di iscrizione *annisoglia*, visualizzare cognome, nome e voto di laurea degli studenti aventi un voto maggiore o uguale a *vsoglia* ed un numero di anni di iscrizione inferiore ad *annisoglia*.
3. Dato un corso di laurea visualizzare i laureati con 110 o 110 e lode.

Esercizio 3

Scrivere un programma che prenda in ingresso **massimo** 10 indirizzi web (per es: “www.polimi.it, ftp.pippo.com”) controllando che siano lunghi almeno quattro caratteri.

Il numero di indirizzi da inserire viene richiesto all’utente all’inizio del programma.

Il programma, attraverso delle funzioni, deve:

1. Completare ogni indirizzo con il relativo protocollo
(es: “http://www.polimi.it, ftp://ftp.pippo.com”)
2. Scrivere gli indirizzi sul terminale e se l’indirizzo da stampare è “http://www.polimi.it”, il programma dovrà visualizzare “SITO DEL POLI”.

N.B.: Ogni indirizzo-stringa è un array. Occorre quindi utilizzare un array di stringhe. Definire la lunghezza massima dell’indirizzo pari a 100.

Esercizio 4

Date le strutture dati “punto”, “segmento” e “triangolo”, scrivere un programma che permetta all’utente, tramite menu e funzioni, di effettuare le seguenti operazioni:

1. Dati 3 punti in input, verificare che questi siano allineati.
2. Dato un segmento in input, calcolare il punto medio del segmento stesso.
3. Dati 3 punti in input, calcolare il perimetro del triangolo definito dai punti stessi.
4. Dati 3 punti in input, calcolare il baricentro del triangolo definito dai punti stessi.

Ricordiamo che:

- Tre punti sono allineati se e solo se

$$\frac{y_3 - y_1}{y_2 - y_1} = \frac{x_3 - x_1}{x_2 - x_1}$$

- Il punto medio di un segmento è

$$x_M = \frac{x_A + x_B}{2} \quad ; \quad y_M = \frac{y_A + y_B}{2}$$

- La lunghezza del segmento AB è

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Il baricentro di un triangolo è

$$x_G = \frac{x_A + x_B + x_C}{3} \quad ; \quad y_G = \frac{y_A + y_B + y_C}{3}$$

Esercizio 5

Scrivere un programma che permetta, tramite una funzione, di inserire un testo (compresi gli spazi) di al massimo 1024 caratteri. Attraverso un menu, il programma deve:

- Inserire un nuovo testo
- Data una parola, eliminare la sua prima occorrenza
- Data una parola, stampare la posizione di tutte le occorrenze di tale parola

Sia nel caso di eliminazione che di ricerca utilizzare una funzione/procedura.

Esercizio 6

Dati N prodotti in M categorie, realizzare:

- Un array di N elementi, contenente i prezzi di N oggetti
- Un array di N elementi, dove, per ogni elemento dell'array precedente, venga specificata una sua categoria (numero da 0 a M)
- Un array di M elementi, contenente gli sconti da applicare a ciascuna categoria. I valori di questo array devono essere casuali.
- Una funzione che permetta di ottenere l'array degli N prodotti scontati e visualizzarli.

Esercizio 7

Realizzare una funzione che, dato in ingresso una stringa, la filtri attraverso una maschera (un array di elementi 0/1: 1 elemento consentito, 0 elemento eliminato).

Stringa	N	e	l		m	e	z	z	o		d	e	l		c	a	m	m	i	n		d	i		n	o	s	t	r	a
Maschera	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Risultato		e	l			e														n										a

Il risultato deve essere fornito sotto forma di stringa. (es. precedente "elena")

Si calcoli la lunghezza della stringa così costruita e la riduzione percentuale del testo rispetto l'originale (es. precedente, lunghezza iniziale 30 caratteri, finale 5, riduzione dell' 83.3%)

Esercizio 8

Si scriva una funzione in linguaggio C che prenda in ingresso il seguente array di 22 interi

```
int array_in[DIM] = {67, 79, 77, 80, 76, 73, 77, 69, 78,  
                    84, 73, 33, 72, 97, 105, 70, 105,  
                    110, 105, 116, 111, 33};
```

ed un puntatore ad un secondo array di char. La funzione riempie il secondo array con i valori del primo array convertiti nel rispettivo carattere (secondo il codice ASCII). Non utilizzare gli indici per gli array, ma solo la logica dei puntatori.

Esercizio 5

Realizzare le seguenti funzioni, che presa in input una stringa:

- Restituiscano tutti i caratteri in maiuscolo
`toUpperCase("Hello World") = "HELLO WORLD"`
- Inverta il case dei caratteri
`invertCase("Hello World") = "hELLO wORLD"`
- Converta il Camel Case in Snake Case
`invertCase("HelloWorld") = "Hello_World"`

Esercizio 3

Scrivere una funzione che permetta di convertire un numero da una base a un'altra. Questo, per qualsiasi base di partenza e destinazione (scelte dall'utente)