

# Preparazione Midterm

## Esercitazione 9

8/11/2024 - Alessandro Montenegro

# Funzioni

## Gestione Strutture

- Quando si fa un assegnamento con strutture, esse vengono copiate elemento per elemento
- Anche se all'interno c'è un array (o una stringa)
- In tal caso viene allocato un nuovo array e ogni elemento del primo viene copiato nel secondo
- Gli array nelle strutture punteranno ad aree di memoria diverse
- Tutto questo vale anche per il passaggio per valore con una struttura

# Funzioni

## Gestione Strutture

**NON è possibile**

restituire un array di  
tipi Dato!

**Ma è possibile**

restituire un Dato con  
all'interno un array!

```
1  #include <stdio.h>
2
3  #define N 3
4
5  typedef struct {
6      int n;
7      int v[N];
8  } Dato;
9
10 Dato f();
11
12 int main(){
13     Dato d1 = {3, {1,2,3}};
14     Dato d2 = d1;
15     Dato d3 = f();
16 }
17
18 Dato f(){
19     Dato tmp = {2, {1,2}};
20     return tmp;
21 }
```



# Funzioni

## Quando servono i doppi puntatori?

```
void libroCostoso2(Scaffale s, int n, Libro *lc)
{
    int i, i_max = 0;
    Libro l;

    for(i = 0; i < n; i++)
        if(s[i].costo > s[i_max].costo)
            i_max = i;

    l = s[i_max];
    *lc = s[i_max]; ?
    *lc = l; ?
    lc = &s[i_max]; ?
}
```

# Funzioni

## Quando servono i doppi puntatori?

```
void libroCostoso2(Scaffale s, int n, Libro *lc)
{
    int i, i_max = 0;
    Libro l;

    for(i = 0; i < n; i++)
        if(s[i].costo > s[i_max].costo)
            i_max = i;

    l = s[i_max];
    *lc = s[i_max]; 👍
    *lc = l; ?
    lc = &s[i_max]; ?
}
```

# Funzioni

## Quando servono i doppi puntatori?

```
void libroCostoso2(Scaffale s, int n, Libro *lc)
{
    int i, i_max = 0;
    Libro l;

    for(i = 0; i < n; i++)
        if(s[i].costo > s[i_max].costo)
            i_max = i;

    l = s[i_max];
    *lc = s[i_max]; 👍
    *lc = l; 👍
    lc = &s[i_max]; ?
}
```




# Funzioni

## Quando servono i doppi puntatori?

In questo caso stiamo cambiando il valore di una variabile passata per valore, quindi **non la modifichiamo nel main**

```
void libroCostoso2(Scaffale s, int n, Libro *lc)
{
    int i, i_max = 0;
    Libro l;

    for(i = 0; i < n; i++)
        if(s[i].costo > s[i_max].costo)
            i_max = i;

    l = s[i_max];
    *lc = s[i_max]; 
    *lc = l; 
    lc = &s[i_max]; 
}
```

Per fare una cosa del genere servirebbe un **doppio puntatore**

# TDE Settembre 2024

## Esercizio 1

Si scriva una funzione **conta** che riceve in ingresso una stringa **S** contenente più parole separate da virgole e contenenti solamente lettere minuscole (oltre ad eventuali parametri aggiuntivi utili per lo sviluppo della funzione).

La funzione **conta** deve restituire al chiamante la lettera dell'alfabeto che compare più frequentemente tra le iniziali delle parole (si supponga per semplicità sia unica).

Si scriva quindi una seconda funzione **elimina** che riceve in ingresso la stringa **S** (contenente più parole e solo caratteri minuscoli), un carattere **C** (oltre ad eventuali parametri aggiuntivi ritenuti utili per lo sviluppo della funzione), e riporta al chiamante una stringa **T** contenente le sole parole di **S** che iniziano per **C**, separate da virgole. Nel main si stampi il vettore restituito dalla funzione **elimina**.



# TDE Marzo 2017

## Esercizio 2

Scrivere una funzione che riceve due stringhe: *parola* e *elimina*. La funzione cerca in *parola* tutti i caratteri che compongono la stringa *elimina*, e li rimuove solamente se li trova tutti e nell'ordine in cui compaiono in *elimina*, anche se non consecutivi. L'operazione viene ripetuta finché in *parola* è contenuta un'intera istanza della stringa *elimina*. La funzione, oltre a modificare l'array *parola*, restituisce il numero di volte che ha eliminato i caratteri dell'intera stringa *elimina* da *parola*. Nell'eliminare caratteri non devono essere lasciati buchi, ma *parola* dev'essere ricompattata.

### Esempio

parola: amaarrreeemmmarrreeaaaarrmae

elimina: mare

La funzione trova tutte le lettere di "mare" in "aMAaRrrEeeemmmarrreeaaaarrmae" che diventa "aarreeemmmarrreeaaaarrmae"

La funzione trova tutte le lettere di "mare" in "aarreeeMmmmARrrEeaaaarrmae" che diventa "aarreeemmmrreeaaaarrmae"

La funzione trova tutte le lettere di "mare" in "aarreeemmMrreAaaaRrrmaE" che diventa "aarreeemmrreeaaarrma"

Poi non trova più tutte le lettere e restituisce 3

# TDE Gennaio 2022

## Esercizio 3

Si scriva una funzione **rimuoviValori** che prende in ingresso una matrice M di interi di dimensioni NxN (con #define N 4), un intero x e ulteriori variabili aggiuntive, se necessarie. La funzione identifica i valori di M che sono a distanza minima da x e modifica M mettendo a zero tutti i valori che stanno sulla parallela alla diagonale principale passante per i valori sopra identificati. Si guardino gli esempi sotto per evitare fraintendimenti su quali elementi mettere a zero.

N.B. Qualora ci fossero più valori in M alla stessa distanza da x, la funzione rimuoviValori ripete l'operazione per tutti i valori a distanza minima. N.B. Si consiglia fortemente l'uso di funzioni ausiliarie per produrre un codice semplice, leggibile e compatto

Si scriva nel main l'invocazione alla funzione rimuoviValori e si stampi la matrice nel main, non in rimuoviValori.

# TDE Gennaio 2023

## Esercizio 4

Si una funzione riordina che prende in ingresso due vettori  $\mathbf{u}$  e  $\mathbf{v}$  (più eventuali variabili necessarie) e che riordina gli elementi del primo vettore passato in input ( $\mathbf{u}$  nel nostro caso) nel seguente modo:

- tutti gli elementi di  $\mathbf{u}$  che non compaiono in  $\mathbf{v}$  saranno nelle prime posizioni di  $\mathbf{u}$  dopo l'invocazione
- tutti gli elementi di  $\mathbf{u}$  che compaiono in  $\mathbf{v}$  saranno nelle ultime posizioni di  $\mathbf{u}$  dopo l'invocazione

Si faccia riferimento all'esempio sotto per eventuali dubbi su come modificare  $\mathbf{u}$ .

Si invochi quindi la funzione relativamente al seguente codice e si stampi il contenuto del vettore  $\mathbf{u}$  a schermo.

N.B. non è possibile stampare all'interno della funzione riordina

$u = [4, 1, 3, 7, 0]$  e  $v = [1, 6, 8, 4] \rightarrow u = [3, 7, 0, 4, 1]$

# Alleanze tra Università

## Esercizio 5

Definire un tipo di dato **University** che descrive un'università. Ogni università è caratterizzata da una stringa che indica il **nome** per **esteso** (ad esempio, "Politecnico di Milano", "École Polytechnique Fédérale de Lausanne", etc.), una stringa che descrive il **nome corto** (ad esempio, "Polimi", "EPFL", etc), due caratteri che contengono la **sigla della nazione** in cui si trova la sede principale (ad esempio, "IT", "CH", etc), un intero che rappresenta il **numero di studenti** immatricolati (ad esempio, 42496, 13445, etc) e un numero reale che descrive il punteggio ottenuto dall'università secondo il **ranking** SDD (ad esempio, 46.7, 277.0, etc).

Definire un tipo di dato **Alleanza** che descrive una lista di **università** che collaborano insieme e il **numero di università** nell'alleanza.

Implementare le seguenti funzioni:

1. `ricerca()`: la funzione verifica se è presente nell'alleanza un'università con un dato nome esteso. Se presente, ritorna le informazioni dell'università, altrimenti segnala al chiamante l'assenza.
2. `filtraPerNazione()`: la funzione restituisce una nuova lista contenente solo le università dell'alleanza che appartengono a una specifica nazione dall'utente.
3. `sostituisci()`: la funzione sostituisce un'università nell'alleanza, identificata dal nome corto, con un'altra università passata in ingresso. La funzione ritorna l'università appena rimossa. La funzione comunica al chiamante se la sostituzione è avvenuta o meno.
4. `massimoStudentiPerNazione()`: la funzione ricerca, per ogni nazione nell'alleanza, l'università con il maggior numero di studenti, e restituisce un lista con le università con il numero massimo di studenti per nazione.

# Contatti

Alessandro Montenegro

Mail: [alessandro.montenegro@polimi.it](mailto:alessandro.montenegro@polimi.it)

Sito: <https://montenegroalessandro.github.io/InfoA2425/index.html>