I VETTORI MULTIDIMENSIONALI

Il linguaggio C



DEFINIZIONE DI VETTORE

- Un vettore può essere definito come una "collezione organizzata di oggetti":
 - "collezione": implica che gli oggetti contenuti nel vettore siano dello stesso tipo;
 - "organizzata": implica che sia possibile identificare univocamente tutti gli oggetti del vettore.





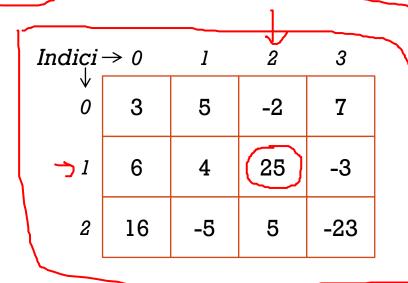
I VETTORI MULTIDIMENSIONALI

- I vettori multidimensionali (o matrici) sono vettori i cui elementi sono a loro volta dei vettori.
- Il numero delle dimensioni non è limitato a priori, anche se normalmente non si superano le tre dimensioni.
- Ogni elemento di un vettore multidimensionale è identificato da tanti indici quante sono le dimensioni del vettore:
 - vettore monodimensionale: l indice;
 - vettore bidimensionale: 2 indici;
 - vettore tridimensionale: 3 indici;
 - ecc.



I VETTORI MULTIDIMENSIONALI

$Indici \rightarrow 0$	1	2	3	4	5	6	7	_
3	5	-2	7	6	4	25	-3	



COME SI DICHIARA UN VETTORE MULTIDIMENSIONALE?

- La dichiarazione di una vettore multidimensionale è simile a quella di un vettore e si compone di 3 elementi:
 - il tipo di dati;
 - 2. il nome del vettore multidimensionale;
 - 3. il numero di elementi, per ogni dimensione del vettore del vettore, scritto tra parentesi quadre.





INIZIALIZZAZIONE DI UN VETTORE MULTIDIMENSIONALE

- Come per tutti gli altri tipi di vettori, anche i vettori multidimensionali, all'atto della dichiarazione, non vengono inizializzati, a meno che non lo si faccia in modo esplicito.
- L'inizializzazione di un vettore multidimensionale può avvenire in molti modi, i due principali sono:
 - contemporaneamente alla dichiarazione dello stesso;
 - in un secondo momento, assegnando esplicitamente <u>ad ogni</u> <u>elemento</u> del vettore il valore voluto.



INIZIALIZZAZIONE DI UN VETTORE MULTIDIMENSIONALE

 Per dichiarare e contemporaneamente inizializzare un vettore multidimensionale si deve procedere come nell'esempio seguente:

 $int \ vet[3][4] = \{\{-1, 4, 27, -5\}, \{10, -4, 2, -15\}, \{-21, 14, 7, 12\}\}$

 Per inizializzare un vettore multidimensionale assegnando esplicitamente ad ogni elemento il valore voluto:

```
int vet[10][5], i, j;
for(f) = 0; i < 10; i++)
{
    for(f) = 0; j < 5; j++)
    {
       vet[i][j] = 0;
    }
}</pre>
```

VETTORI DI STRINGHE

 Un vettore di stringhe non è altro che un vettore bidimensionale di tipo «char» in cui ogni stringa viene memorizzata in una diversa «riga» del vettore.

Indici → 0		1	2	3	4	5	6	7
0	С	i	a	0	\0			
1	P	r	0	v	a	\0		
2	V	е	t	t	0	r	е	\0



Crea e visualizza una tavola Pitagorica 10 x 10.

```
#include <stdio.h>
#include <stdlib.h>
int main()
     int tavolaPitagorica[10][10], i, j;
     /* Creo la tavola Pitagorica */
     for(i = 0; i < 10; i++)
          for(j = 0; j < 10; j++)
                 tavolaPitagorica[i][j] = (i + 1) * (j + 1);
```

```
/* Visualizzo la tavola Pitagorica */
   printf("\nTAVOLA PITAGORICA 10X10\n\n");
for(i = 0; i < 10; i++)
    \longrightarrow for(j = 0; j < 10; j++)
              printf("%3d", tavolaPitagorica[i][j]);
         printf("\n");
   return 0;
```

 Dato un «dizionario» di 20 parole (max 30 caratteri) inserite dall'utente, calcolare la lunghezza media delle parole inserite.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
     char dizionario[20][31], i;
     float media = 0;
     /* Creo il dizionario */
     printf("\nInserisci 20 parole:\n");
     for(i = 0; i < 20; i++)
Nicola Agosti – Sito web: www.nicolaagosti.it
```

```
/* Calcolo la lunghezza media delle parole */

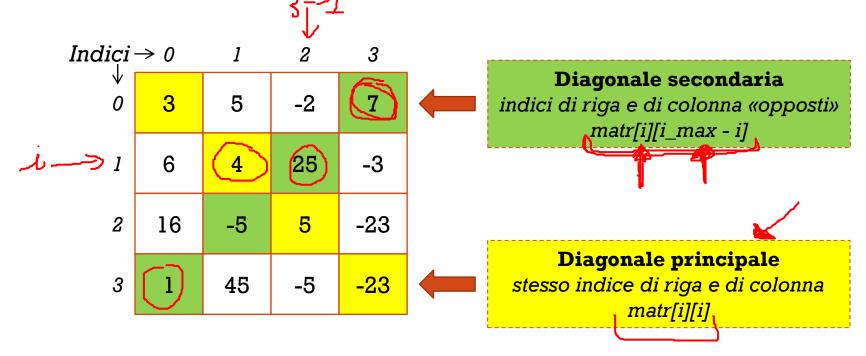
for(i = 0; i < 20; i++)
{
    media += strlen(dizionario[i]);
}

printf("\nLa lunghezza media delle parole e': %f", media / 20);

return 0;
}
```

MATRICI QUADRATE

 Una matrice quadrata è un vettore bidimensionale che ha un ugual numero di righe e di colonne.



GENERAZIONE DI NUMERI (PSEUDO)CASUALI

- int rand(void): ritorna un numero intero (pseudo)casuale nell'intervallo $[0, RAND_MAX]$ con RAND_MAX ≥ 32767 .
- **void srand(unsigned int seed)**: usa *seed* come seme per generare una nuova sequenza di numeri (pseudo)casuali, numeri che verranno poi ritornati dalla funzione *rand*.
- Le funzioni rand e srand sono definite nella libreria «stdlib.h» che deve essere inclusa.
- time_t time(time_t *pt): ritorna il numero di secondi trascorsi a partire dal 01/01/1970.
- La funzione time è definita nella libreria «time.h» che deve essere inclusa.



GENERAZIONE DI NUMERI (PSEUDO) CASUALI

- Per generare un numero pseudo casuale, normalmente, si eseguono i seguenti passaggi:
 - si richiama, una sola volta prima della prima chiamata a rand, la funzione srand passandole come parametro il valore ritornato dalla funzione time.

srand(time(NULL));

2. si richiama, quante volte è necessario, la funzione *rand*. Ogni chiamata ritornerà un nuovo numero (pseudo)casuale.

num = rand();



ESERCIZI

- **Esercizio 1:** scrivere un programma che, data una matrice (vettore bidimensionale) 25x25, assegni ad ogni elemento della matrice il valore 1 se la somma degli indici è pari, 0 altrimenti.
- Esercizio 2: scrivere un programma che data una matrice di numeri interi 10x10 generata casualmente, calcoli e visualizzi:
 - La media di ogni riga.
 - La media di ogni colonna.
 - La media dei valori della diagonale principale e di quella secondaria.
- Esercizio 3: scrivere un programma che chieda all'utente di inserire 20 parole (max 30 caratteri) e verifichi se ci siano o meno parole duplicate tra le parole inserite. Le parole devono essere memorizzate in un vettore di stringhe di dimensioni opportune.