ARRAY MULTIDIMENSIONALI

- ☐ Un array multidimensionale è un array i cui elementi sono a loro volta un array.
- La dichiarazione è del tutto analoga a quella di un array monodimensionale, con la differenza che si dovranno specificare tante coppie di parentesi quadre quante sono le dimensioni dell'array multidimensionale.
- □ Concettualmente un array multidimensionale è una matrice e per accedere ad un suo elemento devo conoscere gli indici della riga e colonna corrispondenti alla posizione dell'elemento.





STRUTTURA DI UN ARRAY MULTIDIMENSIONALE

Row0

Row1

Row2

1	2	3
4	5	6
7	8	9

- L'inizializzazione di un array multidimensionale si effettua specificando tra parentesi graffe i valori di ciascuna riga.
- L'accesso agli elementi dell'array, richiederà tanti cicli for innestati quanti sono gli array monodimensionali costituiscono l'array multidimensionale.





ACCESSO AD UN ARRAY MULTIDIMENSIONALE

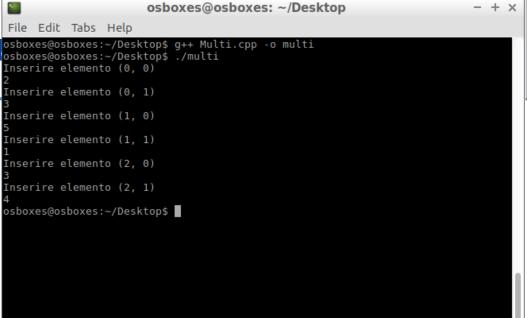
```
#include<iostream>
const int DIM 1 = 3;
const int DIM 2 = 4;
using namespace std;
int main()
    int A[DIM 1][DIM 2];
    int num;
    for(int i=0; i<DIM 1; i++)</pre>
        for (int j=0; j<DIM 2; j++)
            cout<<"Inserire elemento ("<<i<", "<<j<<")"<<endl;</pre>
            cin>>num;
            A[i][j]=num;
    //additional code
    return 0;
```





ACCESSO AD UN ARRAY M

```
Inserire elemento (1, 0)
#include<iostream>
                                            Inserire elemento (1, 1)
                                            Inserire elemento (2, 0)
const int DIM 1 = 3;
                                            Inserire elemento (2, 1)
const int DIM 2 = 4;
                                            osboxes@osboxes:~/Desktop$
using namespace std;
int main()
    int A[DIM 1][DIM 2];
    int num;
    for(int i=0; i<DIM 1; i++)</pre>
         for (int j=0; j<DIM 2; j++)
              cout<<"Inserire elemento ("<<i<<", "<<j<<")"<<endl;</pre>
              cin>>num;
              A[i][j]=num;
    //additional code
    return 0;
```







20

ESERCIZI

- 1. Dato un vettore di 10 interi, scrivere un programma che acquisisca il vettore da terminale, e ne stampi a video somma e media degli elementi.
- 2. Dato il seguente vettore v=[1 2 3 4 5 6 7 8 9 10], scrivere un programma che cerca uno degli elementi scelto dall'utente stampando a video la posizione corrispondente.
- 3. Inserire un vettore di interi allocato staticamente da tastiera, cercare gli elementi pari stampandoli a video.
- 4. Scrivere un programma che acquisisca una matrice 3x3 da terminale e stampa a video la somma degli elementi sulla diagonale principale.





STRINGHE STILE C

☐ In C non esiste un tipo predefinito in grado di memorizzar stringhe di caratteri.
☐ Una stringa è definita come un array di char con terminator NULL ('\0') e si parla di stringhe stile C (C-style string) .
☐ Un array di caratteri però non è un tipo di dato e quindi non possibile applicare su di esso gli operatori definiti dal linguaggi C.
☐ Ad esempio non è possibile usare l'operatore + per concatenar due stringhe stile C.





DICHIARAZIONE ED INIZIALIZZAZIONE DI UNA STRINGA

☐ Consideriamo il seguente array di caratteri;	
char s[10];	
☐ In s sarà possibile memorizzare 9 caratteri (l'ultimo è riservato al carattere di terminazione).	
Il tentativo di memorizzare una stringa di caratteri con lunghezza maggiore di 9 darà luogo ad un errore di segmentation fault.	
L'inizializzazione di un array di caratteri è analoga a quella di un array di qualsiasi altro tipo e può avvenire come segue:	
char mystring[] = {'H','e','l','o','\0'}; char mystring[] = "Hello";	





ASSEGNAZIONE DOPO LA DICHIARAZIONE

- Assegnazioni del tipo mystring[] = "Hello" non sono consentite, in quanto l'assegnazione è ammessa solo al momento della dichiarazione dell'array.
- Per assegnare una stringa di caratteri ad un array precedentemente dichiarato, si devono fare tante assegnazioni quanti sono i caratteri dell'array:

```
mystring[0]='H';
mystring[1]='e';
mystring[2]='l';
mystring[3]='l';
mystring[4]='o';
mystring[5]='\0';
```





STRINGHE I/O

```
#include <iostream>
 2
       #include <cstdio>
      using namespace std;
     int main(){
          const int len = 100;
 8
          char str[len];
 9
 10
          cin >> str;
 11
          cout << str << endl;
12
          scanf("%s",str);
13
          printf("%s",str);
$ ./esercizio
prova
prova
pippo
```

ATTENZIONE!!!

Non c'è controllo sulla lunghezza della stringa!





ESERCIZI

- □ Dato un array di caratteri, scrivere un programma che chieda una lettera all'utente e verifichi se la lettera è presente nell'array
- □ Data una stringa contare i caratteri nella stringa
- Data una stringa e un numero n compreso tra 0 e la lunghezza della stringa, estrarre la sottostringa di lunghezza n. Gestire le situazioni di errore



