

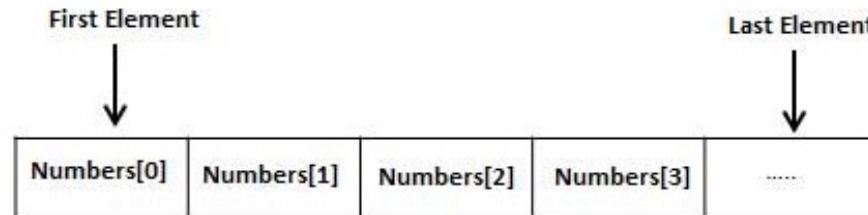
Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

Array

Prof. Carmen De Maio

Riepilogo Array

- Un **array** è un insieme ordinato di elementi dello stesso tipo;
- ogni elemento è individuabile attraverso un **indice** che dà la posizione rispetto il primo elemento.
- L'elemento *i*-esimo di un array **V** contenente **m** elementi è indicato con **V[i]**.
- Gli array sono memorizzati in memoria in locazioni contigue.



Riepilogo Array

- **Array ad una dimensione**

- In C gli array vanno dichiarati utilizzando un valore intero a partire dall'indice 0.

type arrayName [arraySize];

Ad esempio:

`int vettore[10]`

e' un array di 10 elementi interi da `vettore[0]`, ..., `vettore[9]`.

- Nota:

In C non viene fatta nessuna verifica sul superamento dei limiti di un array, per cui per evitare di scrivere oltre il limite deve prestare attenzione chi scrive il codice.

Esercizio 1

- Scrivere un programma che, dati due vettori di interi di dimensione N , ne costruisca un terzo di dimensione $2N$ i cui elementi di posizione pari siano gli elementi del primo vettore e gli elementi di posizione dispari siano gli elementi del secondo vettore.
- I valori dei vettori sono chiesti all'utente

Esercizio 2

Ricerca delle posizioni del massimo e del minimo

- Data in input una sequenza di numeri interi, terminati dal valore -999 (sentinella)
- scrivere un programma che, utilizzando un vettore di interi **vett** per memorizzare tutti i dati inseriti dall'utente, ricerchi il **minimo** e il **massimo** della sequenza e ne stampi a video anche le relative posizioni.

Esempio di ESECUZIONE:

Inserire una sequenza di numeri interi (terminare con -999): 5 -100 0 15 -9 -999

Numero di elementi inseriti: 5

Il minimo è il 2° numero: -100

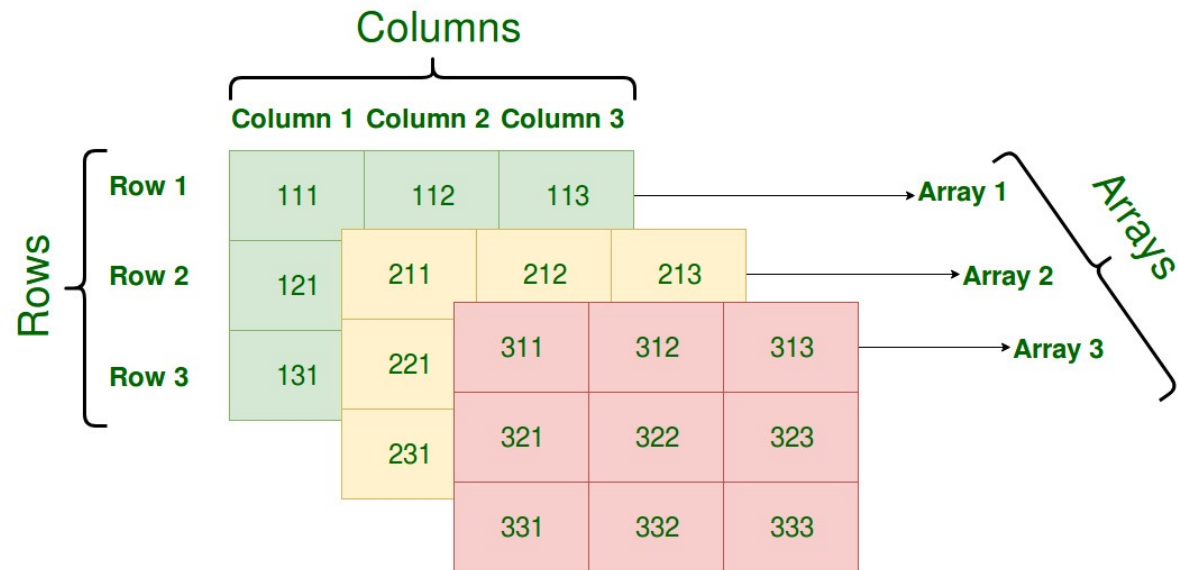
Il massimo è il 4° numero: 15

Esercizio 3

- Scrivere un programma che crei un array di 7 interi forniti dall'utente, controlli se l'array è ordinato in ordine **strettamente decrescente** e stampi a video sia l'array che il risultato*

*NOTA- il risultato è:

"E' strettamente decrescente " oppure " Non è strettamente decrescente : pos X " dove X è la posizione del primo elemento fuori ordine.



Array bidimensionali

Prof. Carmen De Maio

Matrici (array bidimensionali)

- Un **array bidimensionale**, anche detto **matrice**, è una variabile strutturata tale che:
 - gli elementi sono tutti dello stesso tipo e
 - il meccanismo di accesso (diretto) ai suoi elementi consiste di due espressioni intere (indici)

Esempi:

```
int C[4][3];
```

```
float f[M][N];
```

```
char b[2][3]={{'a', 'b', 'c'},{'d', 'e', 'f'}}
```

```
int A[ ][3]= { 1,2,3,4,5,6,7,8,9 };
```


Matrici (array bidimensionali)

- Le matrici possono essere rappresentate per comodità in **formato tabellare**

	Colonna 0	Colonna 1	Colonna 2
Riga 0	C[0][0]	C[0][1]	C[0][2]
Riga 1	C[1][0]	C[1][1]	C[1][2]
Riga 2	C[2][0]	C[2][1]	C[2][2]

- Ogni elemento viene scritto utilizzando **due coppie di parentesi quadre**:
 - la prima coppia di parentesi contiene l'indice della riga,
 - la seconda coppia l'indice della colonna.

Matrici (array bidimensionali)

- Se inseriamo nella matrice C dei valori interi otteniamo la seguente rappresentazione
per esempio: $C[0][0]=3$; $C[2][1]=2$; $C[3][2]=-6$, ...

	Colonna 0	Colonna 1	Colonna 2
Riga 0	3	-5	7
Riga 1	4	8	-1
Riga 2	0	2	6
Riga 3	9	1	-6

Esempi

- Creare una matrice nxm

```
printf("inserisci la matrice\n");
for(i=0; i< m; i++)
    for(j=0; j< n; j++) {
        scanf("%d",&s);
        a[i][j] = s;
    }
```

- Stampare una matrice nxm

```
printf("la matrice è:\n");
for(i=0; i< n; i++) {
    for(j=0; j< m; j++)
        printf("%d ",s[i][j]);
    printf("\n");
}
```

Esempio definizione Matrici

- Una classe composta da 32 studenti ha sostenuto durante l'anno 5 compiti in classe. Supponiamo di voler scrivere un programma che calcoli la media dei voti ottenuti dagli studenti (TODO) .
- Si potrebbe allora dichiarare una matrice del tipo: ***int A[32][5]*** in cui inserire i voti riportati da ciascun studente in ciascun compito. Siccome però in una classe ci potrebbero essere più studenti o i compiti potrebbero essere di più, conviene adoperare per maggior generalità la seguente dichiarazione:

int const R=40; int const C=8; int A[R][C]; int n=32, m=5.

Esempio - Matrici

- una classe di 32 studenti ha sostenuto durante l'anno 5 compiti in classe. Supponiamo di voler scrivere un programma che stampi per ogni studente la somma e la media dei voti ottenuti

dichiariamo una matrice A[][] di interi contenente ...

su ogni riga i voti di ogni studente (32), ...

```
int const R=40;  
int const C=8;  
int A[R][C];  
int n=32;  
int m=5;
```

su ogni colonna i voti di ogni compito

Compito	0	1	2	3	4	5	6	7
Studente	0	9	6	0	7	7	NON	USATA
1	5	6	6	0	6			
2	8	6	7	0	0			
3	4	6	5	4	4			
...		NON	USATA
...			
31	2	4	4	5	4			
...	NON USATA							
39								

Esempio - Matrici

- una classe di 32 studenti ha sostenuto durante l'anno 5 compiti in classe. Supponiamo di voler scrivere un programma che stampi per ogni studente la somma e la media dei voti ottenuti

dichiariamo una matrice $A[] []$
di interi contenente ...

Voto 0 è un compito non consegnato

```
int const R=40;
int const C=8;
int A[R][C];
int n=32;
int m=5;
```

$A[3][2]$ è il voto del 3° compito del 4° studente

Compito	0	1	2	3	4	5	6	7
0	9	6	0	7	7	NON	USATA	A
1	5	6	6	0	6			
2	8	6	7	0	0			
3	4	6	5	4	4			
...	NON USATA		
...			
31	2	4	4	5	4			
...	NON USATA							
39								

ESERCIZIO 5

- Scrivere un programma che inizializzi e visualizzi una matrice 10x10 con i dati della tavola pitagorica

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

ESERCIZIO 6

- Realizzare un programma che permetta il caricamento di una matrice A quadrata di interi.
 - Il programma verificherà se la matrice è di tipo simmetrico.
 - Visualizzare la matrice A.
- Esempio di Matrice simmetrica

1	7	3	8
7	6	4	11
3	4	0	13
8	11	13	5

ESERCIZIO 7

- Realizzare un programma che permetta il caricamento di una matrice A quadrata di interi.
 - Il programma caricherà un vettore B di interi nel seguente modo:
 - ogni elemento alla posizione i-esima del vettore B sarà uguale al prodotto di tutti i valori della matrice A alla riga i-esima divisibili per 3. Se non ci sono valori divisibili per 3 il prodotto è 0.
 - Calcolare e visualizzare il valore massimo del vettore B.
 - Visualizzare la matrice A ed il vettore B.

A

1	6	3	8
9	6	4	11
18	12	1	15
0	1	1	2

B

18
54
3240
0

max = 3240

Homework 1

- Scrivere un programma che consenta a due giocatori umani di giocare a “tris”.
 - Il programma innanzi tutto chiede i nomi dei 2 giocatori (il primo giocatore è associato al simbolo X, il secondo al simbolo O),
 - chiede a ciascun giocatore, alternativamente, di indicare in quale casella segnare il proprio simbolo.

NOTA: usare la seguente numerazione per indicare le caselle.

1	2	3
4	5	6
7	8	9

- Ad ogni turno il programma, dopo aver validato la mossa, visualizza a video la scacchiera.
- Il programma termina non appena uno dei 2 giocatori fa tris, visualizzando un apposito messaggio di vittoria, oppure quando tutte le caselle della scacchiera sono occupate, quindi la partita è pari.