0,0	0,1	0,2	0,3	0,4	qui	
1,0						
2,0			(25)			
						4,6

0	1	2	3	4																														34

#define RIGHE 50

#define COLONNE 70

Int tabella[RIGHE][COLONNE];

int tabella1[RIGHE1][COLONNE1]

// leggere da tastiera numerorighe per esempio 5

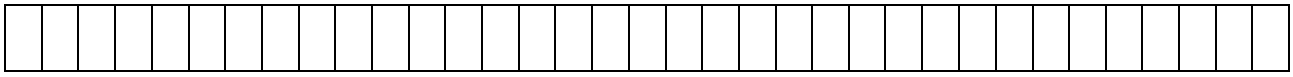
//leggere da tastiera numerocolonne per esempio 7

tabella[2][3]=25;

// IN alternativa al vettore statico

int *tabella;

tabella=(int *)
malloc(sizeof(int)*numerorighe*numerocolonne);



*(Tabella+2+3)

//array dinamico a due dimensioni VERE

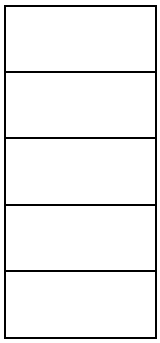
Int ** tabella;

tabella=(int **) malloc(sizeof(int *)*numerocolonne)

Int *	Int *	Int *	Int *	Int *	Int*	Int *

```
i=0;
while(i<numerocolonne)
//parentesi graffa
*(tabella+i)=(int) malloc(sizeof(int)*numerorighe);
i=i+1;
//parentesigraffa
```

Int *	Int *	Int *	Int *	Int *	Int*	Int *



```
*(*(tabella+3)+2)=25;
```

//oppure

```
Int ** tabella;

tabella=(int **) malloc(sizeof(int *)*numerorighe)




i=0;
while(i<numerorighe)
//parentesi graffa
*(tabella+i)=(int) malloc(sizeof(int)*numerocolonne);
i=i+1;
//parentesigraffa
```

Int *	Int	Int	Int	Int	Int	Int	Int
Int *							
Int *							
Int *							
Int *							

((tabella+2)+3)=25;

((tabella+3)+1)=*(*(tabella+2)+3);

```
//ancora meglio

typedef struct {
	int ** valori;
	int righe;
	int colonne;
} tabella;



tabella t1, t2, t3;

//leggererighet1
T1.righe=righet1;
//leggerecolonnet1
T1.colonne=colonnet1;
//allocare spazio a valori
T1.valori=(int **) malloc(sizeof(int *)*t1.righe)
```

```
i=0;
while(i<t1.righe) {
*(t1.valori+i)=(int) malloc(sizeof(int)*t1.colonne);
i=i+1;
}
//inserire I valori della tabella t1
ESERCIZIO CHE FARETE VOI

//leggererighet2
T2.righe=righet2;
//leggerecolonnet2
T2.colonne=colonnet2;
//allocare spazio a valori
T2.valori=(int **) malloc(sizeof(int *)*t2.righe)
```

```
i=0;
while(i<t1.righe) {
*(t2.valori+i)=(int) malloc(sizeof(int)*t2.colonne);
i=i+1;
}
//inserire I valori della tabella t2
```
ESERCIZIO CHE FARETE VOI

```
//se non vi piaccion i vettori dinamici
#define MAX 100

typedef struct {
	int valori[MAX][MAX];
	int righe;
	int colonne;
} tabella;

tabella t1;

//leggere righet1
T1.righe=righet1;
//leggere colonne t1
T1.colonne=colonnet1
// Inserire i valori in t1
ESERCIZIO PER VOI
```

t1[2][3]

((t1+2)+3)

Elemento in riga 2 e colonna3 di t1

```
// terza possibilita' di media difficolta'
typedef struct {
	int *valori;
	int righe;
	int colonne;
} tabella;

//leggere righet1
T1.righe=righet1;
//leggere colonne t1
T1.colonne=colonnet1
//allocare lo spazio

T1.valori=(int *) malloc(sizeof(int)*t1.righe*t1.colonne);
// Inserire i valori in t1
//trovare formula
ESERCIZIO PER VOI


T1[2][3]=25;  //sbagliato
T1[2+3]=25; //sbagliato
```

((t1+2)+3)=25; //sbagliato

*(t1+2+3)=25; //sbagliato

*(t1.valori+FORMULA)=25; //corretto

t1.valori[FORMULA]=25; //corretto

//Generalita’ sui tipi di dato

Tipo di dato (tabella) ->

1. Struttura del tipo di dato
2. Operazioni definite per il tipo di dato

Esempi di operazioni

- Inserire elementi
- Stampare tabella
- Moltiplicazione di due tabelle
- Somma di due tabelle
- ….

ALMENO QUESTE DI ACCESSO AL DATO VANNO DEFINITE:

- Leggere campi
- Scrivere campi

Per ogni campo una funzione di lettura e una funzione di scrittura

CASO STATICO

1. LeggereRighe
2. LeggereColonne
3. LeggereValore
4. ScrivereRighe
5. ScrivereColonne
6. ScrivereValore

//LeggereRighe

Riceve in input una tabella e restituisce il valore del campo righe

//Leggerecolonne

…

//LeggereValore

Riceve in input una tabella e le due coordinate del valore da leggere restituisce in output il valore contenuto in quelle coordinate

//ScrivereRighe

Riceve in input la tabella e il numero di righe e restituisce in output il record modificato

//ScrivereColonne

…

//ScrivereValore

Riceve in input la tabella le coordinate del valore da modificare e il nuovo valore da scrivere e restituisce in output il record modificato

LeggereRighe

INPUT

t tabella di cui leggere il numero di righe – tabella

OUTPUT

righe – numero di righe di t – intero - >0

ALGORITMO

righe=campo righe di t

```
int LeggereRighe(tabella t)
{int righe;
righe = t.righe;
return righe;
}
```

```
int LeggereValore(tabella t, int r, int c)
{
return t.valori[r][c];
}

tabella ScrivereRighe(tabella t, int righe)
{
t.righe=righe;
return t;
}

tabella ScrivereValore(tabella t, int r, int c, int valore)
{
r.valori[r][c]=valore;
return t;
}
```

ESERCIZIO PER IL 18 DICEMBRE (CASO STATICO)

- Realizzare pseudocodice e codice delle 6 funzioni di accesso
- Realizzare pseudocodice e codice delle funzioni di inserimento di tutti i valori e di stampa di tutti i valori di una tabella