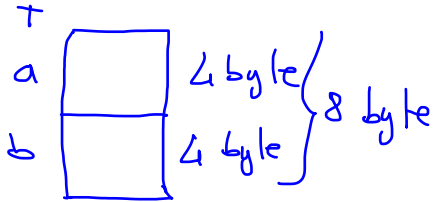


```
typedef struct {
    int a;
    float b;
} T
```



```
typedef union {
    int a;
    float b;
}
```



000. 010
0100.0010
 32 bit

} + main() {

T v;
 8 v.a = 2;
 5 v.b = 3.5;

↑ float
 ✓ a + v.b
150.0 3.5
 153.5
 ? float

✓ 235 main

```
typedef union {
    int dato_intero;
    float dato_reale;
} Data;
```

```
typedef struct {
    enum {intero, reale} tipo;
    Data dato;
} Numero;
```

↑ discriminante
 ← union

```
Numero n;
n.dato.dato_reale = 3.5;
n.tipo = reale;
```

```
if (n.tipo == reale)
    ...
else
    ...
```

```
switch (n.tipo) {
    case reale: ...; break;
    case intero: ...; break;
    ...
}
```

$$\frac{2}{3} \cdot \pi$$

$$0,66667 \cdot$$

$$\frac{2}{8} \cdot \frac{4}{5} = \frac{8}{18}$$

```
typedef struct {
    char nome[30];
    char cognome[30];
    int eta;
} persona;
```

```
typedef persona Classe[20];
Classe c;
```

nome
cognome
eta

M	A	R	I	A	D	.								
R	O	S	I	G	-									
25														

c[0]

G	I	L	I	A										
B	A	R	C	H										

c[9]

c[19]

espressione

printf("%c", c[9].cognome[0])

c[9].cognome[0]

c[9].cognome[1]

```
int main() {
```

```
    int a[10] = {1, 10};
```

```
    int *p;
```

```
    p = a;
```

```
    f(p)
```

```
    f(a);
```

```
    p = g(a); g(a);
```

```
void f(int b[]) {
```

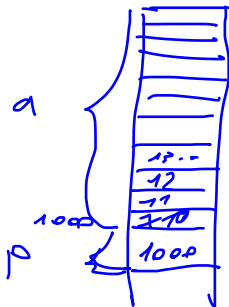
```
    b[5] = 0;
```

```
    *(b + 5) = 0;
```

```
int *g
```

```
    int *c;  
    void {  
        int c[10] = {10, 11, ..., 19};
```

```
        c[0] = 10, c[1] = 11, ...;  
        return c;  
    }
```

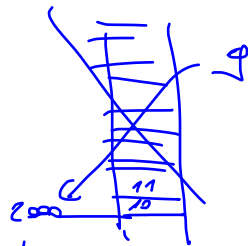


main

*p → 1

a[0] → 1

p[0] → 1

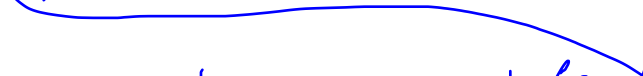


```
int main ( ) {  
    Name n; = {"Mano"};  
    char name[30];  
    g(name);  
    h(n);  
}
```

```
void g(char c[]);  
    c[0] = 'A';  
}
```

```
typedef struct {  
    char name[30];  
} Name;
```

```
void h (Name n) {  
    n.name[0] = 'A';  
}
```



2 {

4	1	0
5	8	7

3

riga

4	1	0
---	---	---

riga

5	8	7
---	---	---

M
typerdef int Riga[3]
Riga M[2];

int M[2][3];
RIGAS COLUMNS

7
0
8
1
5
4

M[1]

1016	7
1016	8
1016	5
1000	0
1004	1
1000	4

M[2]

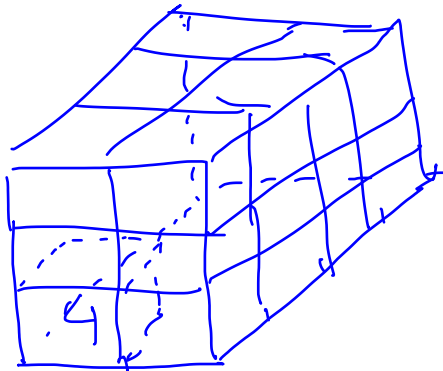
M[0]

$$M[1][1] = 8$$

$1000 + 3 * 4 + 1 * 4 = 1016$
 ↑ ↑
 dim riga dim int

FORTRAN
PER COLUMN

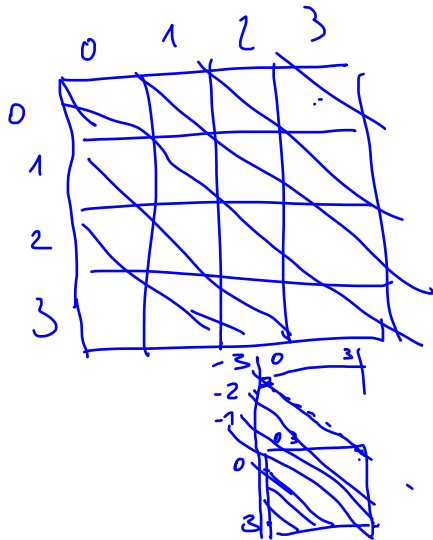
```
int M[2][3][4];
```



```
(char, int) a[10];  
a[3]
```

```
typedef union {  
    char c;  
    int i;  
} IntOrChar
```

```
IntOrChar a[10];
```

int $M[4][4]$

[
 $M[0][3]$
 $M[0][2]$ $M[1][3]$
 $M[0][1]$ $M[1][2]$ $M[2][3]$
 $M[0][0]$ $M[1][1]$ $M[2][2]$ $M[3][3]$
 $M[1][0]$ $M[2][1]$ $M[3][2]$
 $M[2][0]$ $M[3][1]$
 $M[3][0]$
]