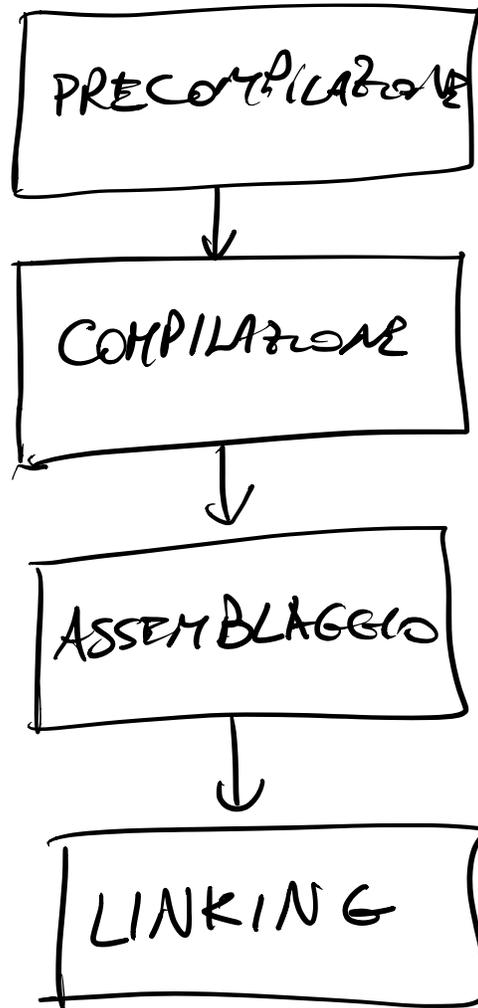


# COMPILAZIONE



# FUNZIONI DI LIBRERIA

- includere header file ←
- linking: collegare le librerie appropriate

## TIPICOMPOSTI

GO	C
struct	sì, identiche
map	no
tipi funzione	sì, diff. sintattiche
slice	no
interface	no
array	sì, con nome diff.
puntatori	sì, molto permissivi!
↑ string	↑ no

```
var x { struct {  
    a, b int  
}
```

```
struct {  
    int a, b  
} x; ←
```

---

typedef

```
typedef struct {  
    int a, b
```

```
} x;
```

```
x y, z;
```

# PUNTORI

var p \*int

var x \*persona

int \*p;

persona \*x;

\* / &

```
typedef struct {  
    int g, m, a  
}
```

```
data d;  
data *p;
```

```
d.g = 29;    d.m = 11;    d.a = 1968;  
p = &d;  
(*p).a = 2020;
```

GO

p.a = 2020;

p->a = 2020;

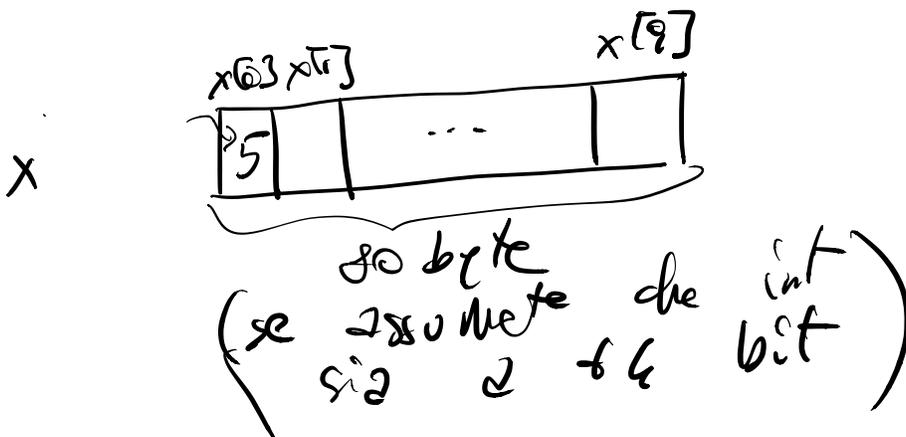
C

# ARRAY

		GO		C
<u>var</u>	x	[10]int		<u>int</u> x[10];
<u>var</u>	p	[100]persona		persona p[100];
<u>var</u>	q	[100]*persona		*q[100];

C89: statica  
 C99: dinamica

- Il nome di un array  
 è sinonimo di  
 &(x[0])



```
int x[10];  
int *p;
```

```
p = x; ← p = &(x[0]);  
*p = 5; ↔ x[0] = 5;
```

```
void fill (int *x, int n) {  
    int i;  
    for (i = 0; i < n; i++)  
        x[i] = 1;  
}
```

```
void main () {  
    int a[10], b[100];  
    fill (a, 10);  
    fill (b, 100);  
}
```