

VARIABLE

<u>var</u> (<u>int</u> x, y;
	<u>int</u>	
	<u>float64</u>	<u>double</u> f;
	<u>bool</u>	<u>int</u> z;
)		

```
#include <stdio.h>
```

```
void main() {  
    int x, y, z;
```

```
    z = x;
```

```
    y = x * x;
```

```
    printf ("%d \n", y / z - x);
```

```
}
```

```
#include <stdio.h>
```

```
void main () {
```

```
    int n, d;
```

```
    scanf ("%d", &n);
```

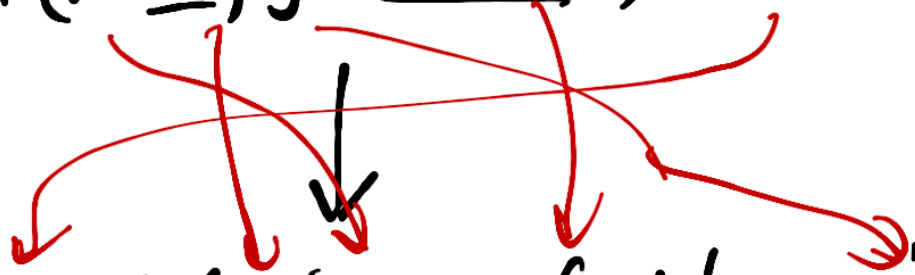
```
    d = 2*n;
```

```
    printf ("%d\n", d);
```

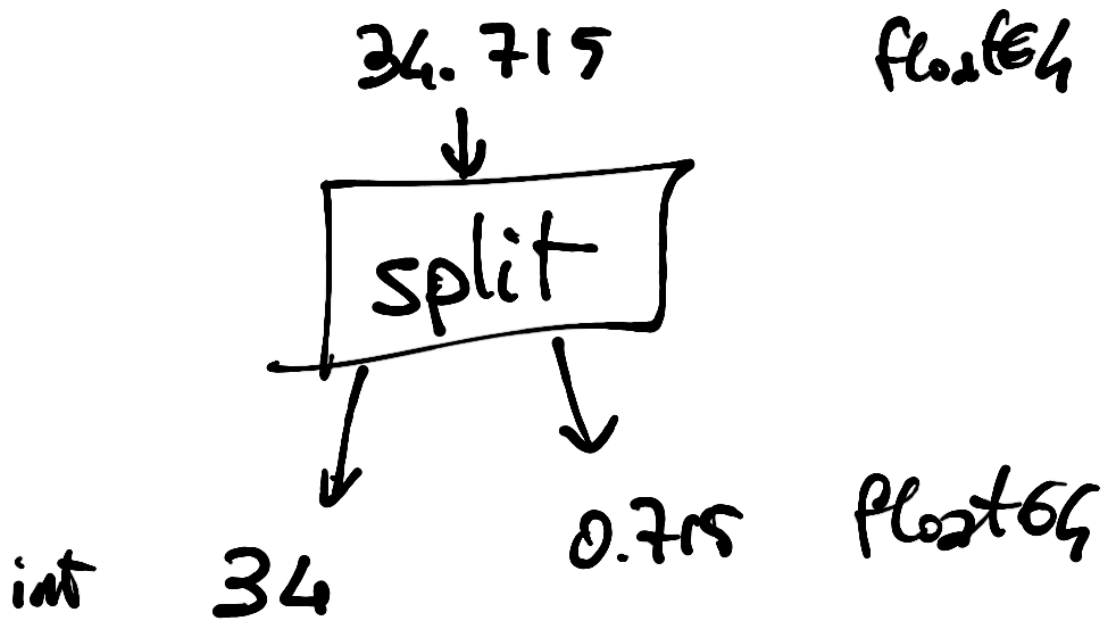
```
}
```

~~func~~ f(x int, y float) bool

int f(int x, double y)



void f(int x, double y)



```

func
    split(x float64) (int, float64)
    var
        i int
        f float64
    )
    i = int(x)
    f = x - float64(i)
    return i, f

```

}

```
func splitp(x float64, pi *int, pf  
            * float64) {  
    *pi = int(x)  
    *pf = x - float64(*pi)  
}
```

```
var (a int  
      b float64)
```

```
a, b = split(13.456)
```

```
splitp(13.456, &a, &b)
```

ESPRESSO W1

- C estremamente più
veloce

double f = 3.14;

int x = 6;

long l = 7000000000;

f = l * x / f

↑ ↑ ↑
long int double

- CAST

l = (int) (f * 2)

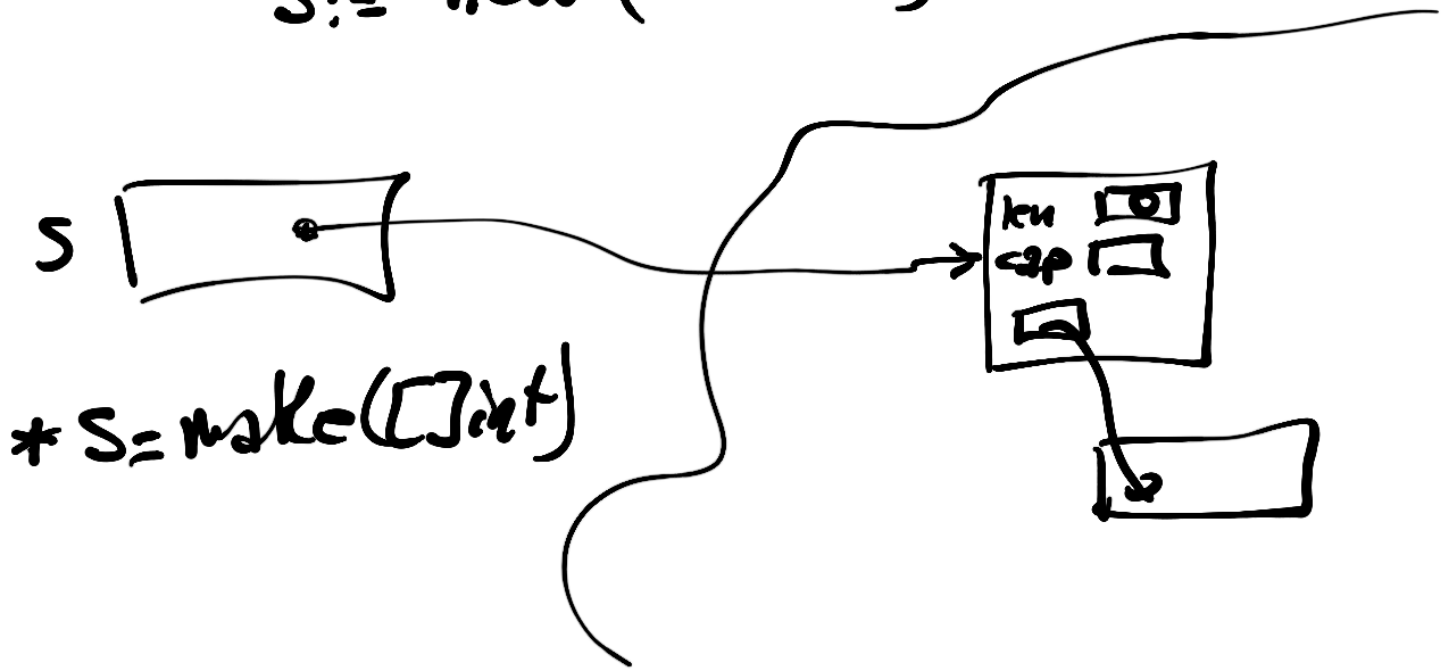
int somma, n;
double media;

...

~~media = somma / n;~~

media = (double) (somma) / n;

S := new ([] int) S * [] int



TIP

GO

C

INTERI
INTERI SENZA S.

int long
unsigned int
unsigned (long
 (int)
- double
float
-
- *

bool
FLOATING POINT
COMPLEX
string

struct
map
array
slice
pointer
interface
tipi funzionali

si
-
si *
-
si *
-
si

PUNTORI

var p *int

var a *persona

int *p;
persona a #2;

*p = 3
*p = (*p + 1)

*p = 3
*p = (*p + 1)

(*a).nome = "Paolo" → OK
a.nome = "Paolo" → a → nome = "Paolo"

var i int
p = &i

int i;
p = &i

GO

nil

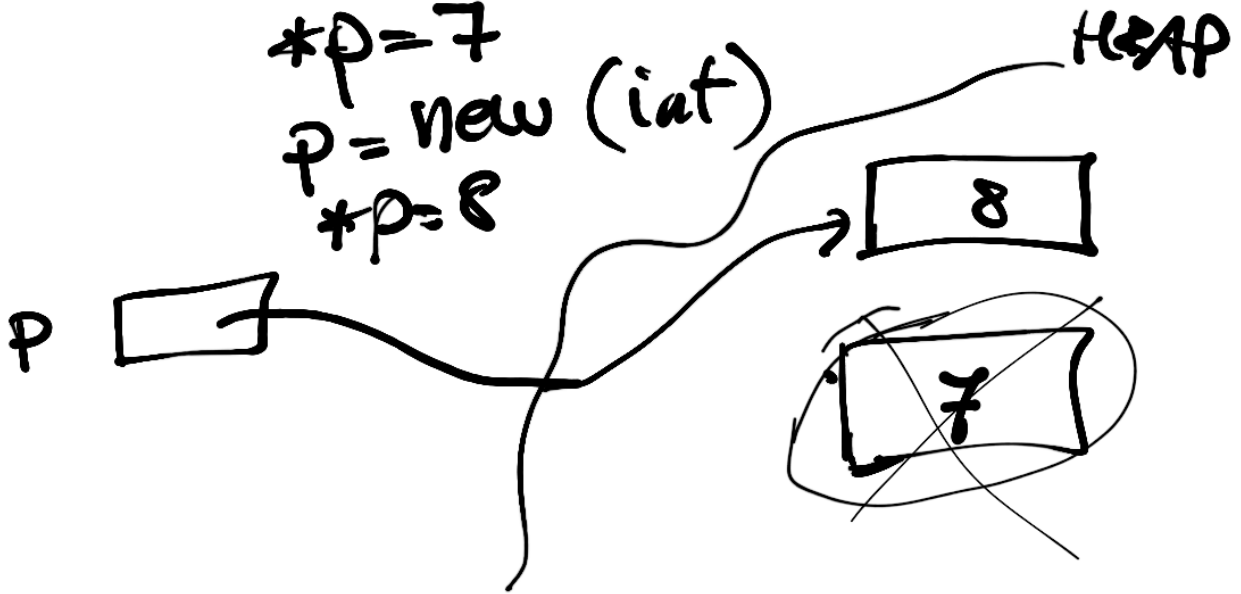
var p *int

p = new (int)

*p = 7

p = new (int)

*p = 8



C

#include <stdlib.h>
int *p;

p = (int *) malloc(sizeof(int));

void * ≡ void*

free(p);

NULL

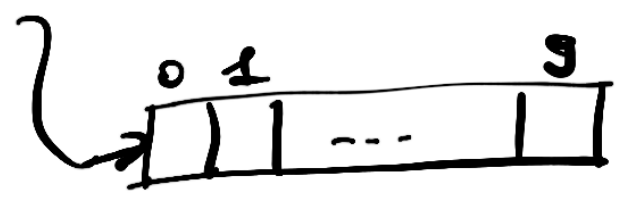
ARRAY

```
var (  
  x [10] int  
  y [100] * persons  
)
```

```
int x [10];  
persons * y [100];
```

ARRAY NON ESISTONO!

```
int x [10];  
x è il puntatore
```



$$x = \&(x[0])$$

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

```
void main() {  
    int a [10];  
    fill (a, 10, 3);  
}
```

