

Variabili e memoria

**Corso di
Laboratorio di
Calcolo**

Variabili e costanti

```
a = 2;
```

```
b = 3.5;
```

```
c = 2.;
```

0xfffffa000	
0xfffffa001	
0xfffffa002	00000010
0xfffffa003	
0xfffffa004	
0xfffffa005	
0xfffffa006	
0xfffffa007	

Variabili e costanti

```
a = 2;
```

```
b = 3.5;
```

```
c = 2.;
```

Nota: per brevità
le variabili intere
sono rappresentate
usando solo 8 bit

0xfffffa000	
0xfffffa001	
0xfffffa002	00000010
0xfffffa003	
0xfffffa004	
0xfffffa005	
0xfffffa006	
0xfffffa007	

Variabili e costanti

`a = 2;`

`b = 3.5;`

`c = 2.;`

$3.5 = (1 + 0.75) * 2^1$
NOTA: l'esponente è
rappresentato in
eccesso (cap.1,p.19)

0xfffffa000	
0xfffffa001	
0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	

Variabili e costanti

`a = 2;`

`b = 3.5;`

`c = 2.;`

`2. = (1 + 0) * 21`

0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	00000000
0xfffffa009	00000000
0xfffffa00a	00000000

Variabili e costanti

```
a = 2;
```

```
b = 3.5;
```

```
c = 2.;
```

```
d = 'f' ;
```

f ↔ 102 (66₁₆)

0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	00000000
0xfffffa009	00000000
0xfffffa00a	00000000
0xfffffa00b	01100110

Operatori matematici

`c = a + b;`

0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	00000000
0xfffffa009	00000000
0xfffffa00a	00000000

Operatori matematici

`c = a + b;`

Nota: nel registro della CPU il numero 2 viene “promosso” a `float`

0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	00000000
0xfffffa009	00000000
0xfffffa00a	00000000

Operatori matematici

`c = a + b;`

0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	00000000
0xfffffa009	00000000
0xfffffa00a	00000000

Operatori matematici

`c = a + b;`

$$5.5 = (1 + 0.375) * 2^2$$

0xfffffa002	00000010
0xfffffa003	01000000
0xfffffa004	01100000
0xfffffa005	00000000
0xfffffa006	00000000
0xfffffa007	01000000
0xfffffa008	10110000
0xfffffa009	00000000
0xfffffa00a	00000000

Operatori speciali

```
a = 2;
```

```
b = ++a;
```

0xfffffa002	000000010
0xfffffa003	

Operatori speciali

```
a = 2;
```

```
b = ++a;
```

0xfffffa002	00000011
0xfffffa003	00000011

Operatori speciali

```
a = 2;
```

```
b = ++a;
```

```
a = 2;
```

```
b = a++;
```

0xfffffa002	000000010
0xfffffa003	

Operatori speciali

```
a = 2;
```

```
b = ++a;
```

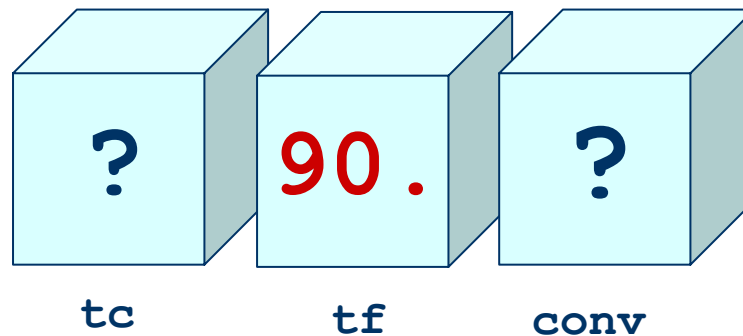
```
a = 2;
```

```
b = a++;
```

0xfffffa002	00000011
0xfffffa003	00000010

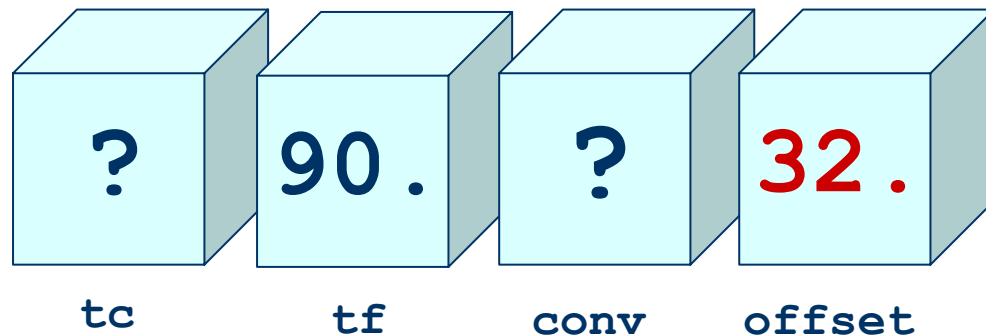
Programmi e variabili

```
main() {  
    float tc, tf = 90., conv;  
    float offset = 32.;  
    conv = 5./9.;  
    tc = (tf - offset) * conv;  
}
```



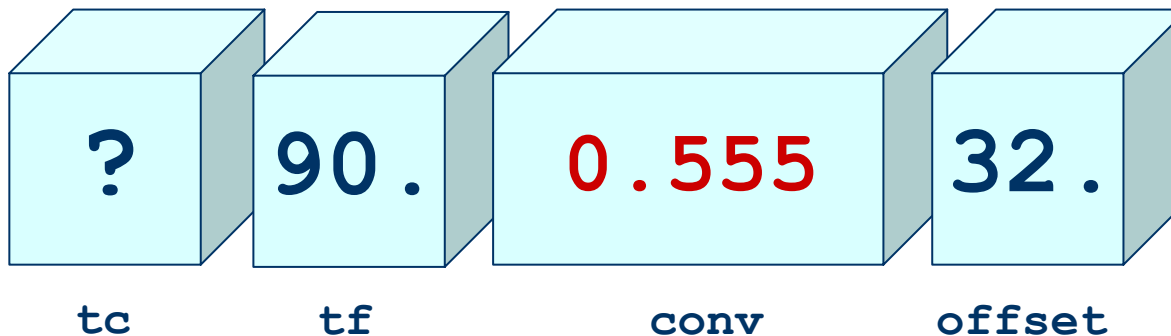
Programmi e variabili

```
main() {  
    float tc, tf = 90., conv;  
    float offset = 32.;  
    conv = 5./9.;  
    tc = (tf - offset) * conv;  
}
```



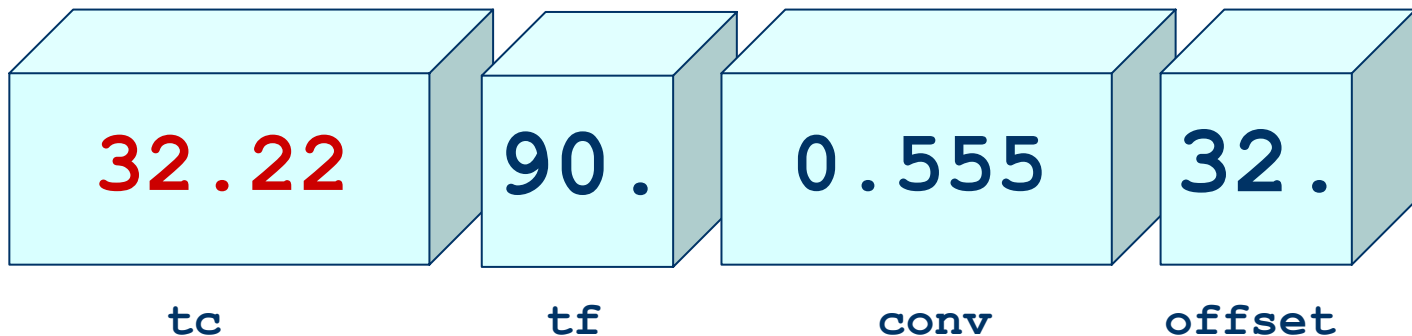
Programmi e variabili

```
main() {  
    float tc, tf = 90., conv;  
    float offset = 32.;  
    conv = 5./9.;  
    tc = (tf - offset) * conv;  
}
```



Programmi e variabili

```
main() {  
    float tc, tf = 90., conv;  
    float offset = 32.;  
    conv = 5./9.;  
    tc = (tf - offset) * conv;  
}
```



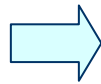
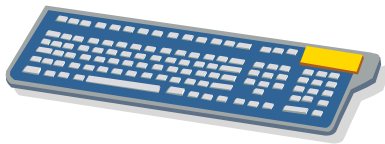
- Cosa accade se eseguo il programma?
- Nulla, perché non ci sono istruzioni di *output*!

Input/Output

```
scanf ("%f", &tf) ;
```

Input/Output

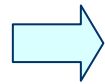
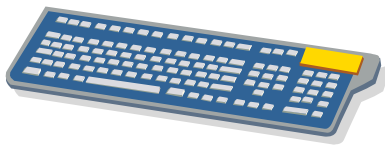
```
scanf ("%f", &tf) ;
```



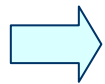
90

Input/Output

```
scanf ("%f", &tf) ;
```



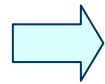
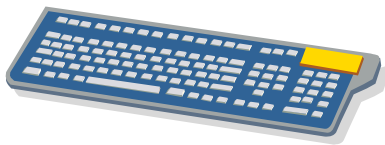
$$90 = (1 + 0.40625) 2^6$$



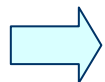
010000101011010000000000...

Input/Output

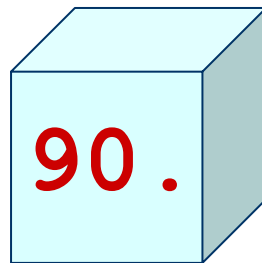
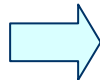
```
scanf ("%f", &tf) ;
```



$$90 = (1 + 0.40625) 2^6$$



010000101011010000000000...



Input/Output

```
printf("%f F = %f C\n", tf, tc);
```

Input/Output

```
printf("%f F = %f C\n", tf, tc);
```



Input/Output

```
printf("%f F = %f C\n", tf, tc);
```



90.000000 F = ...

Input/Output

```
printf("%f F = %f C\n", tf, tc);
```



90.000000 F = 32.222222 C...

Input/Output

```
printf("%f F = %f C\n", tf, tc);
```



```
90.000000 F = 32.222222 C  
>
```

Importanza della rappresentazione

```
(tc - offset) * 5. / 9. ;
```

```
(tc - offset) * 5. / 9 ;
```

```
(tc - offset) * 5 / 9 ;
```

```
5 / 9 * (tf - offset) ;
```

#include

Consente di includere il contenuto di un altro file

#define

Definisce un simbolo (Attenzione!!!)

#ifdef, #ifndef, #else

Compilazione condizionale