

# Il linguaggio C

Funzioni: frame di chiamata

# Esempio di funzione

```
#include <stdio.h>
int max (int a, int b);

int main (void){
    int x;
    x = max(8+2,2);
    printf("Il massimo è %d \n", x);
    return 0;
}

int max (int a, int b) {
    int tmp;
    tmp = (a < b)? b : a
    return tmp;
}
```

# Funzioni C

- Tutti i parametri sono passati *per valore*
  - una copia del valore viene messa sullo stack all'atto della chiamata
  - la copia viene rimossa quando la funzione termina e non è più accessibile
- Variabili locali
  - Il corpo di una funzione contiene le variabili locali
    - sono allocate sullo stack,
    - sono accessibili solo a quella funzione
    - perdono il valore fra un'invocazione e l'altra

# Esempio di funzione (.1)

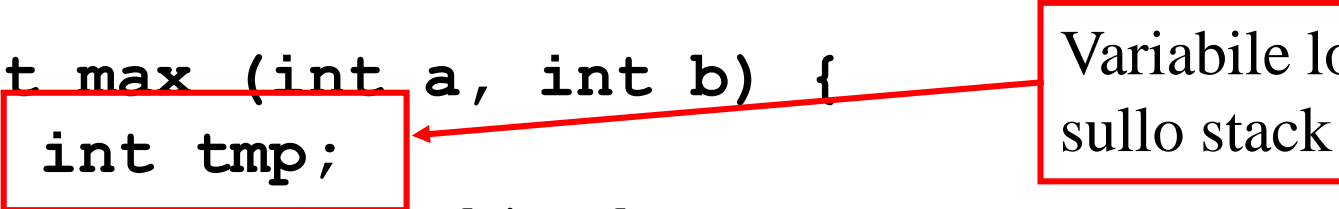
```
#include <stdio.h>
int max (int a, int b);
int main (void) {
    int x;
    x = max(8+2, 2);
    printf("Il massimo è %d \n", x);
    return 0;
}
```

```
int max (int a, int b) {
    int tmp;
    tmp = (a < b) ? b : a;
    return tmp; /* definizione */
}
```

Parametri attuali  
Vengono valutati  
Il loro valore viene  
copiato sullo stack

# Esempio di funzione (.2)

```
#include <stdio.h>
int max (int a, int b);
int main (void) {
    int x;
    x = max(8+2, 2);
    printf("Il massimo è %d \n", x);
    return 0;
}
int max (int a, int b) {
    int tmp;
    tmp = (a < b) ? b : a;
    return tmp; /* definizione */
}
```



Variabile locale  
sullo stack

# Esempio di funzione: il frame (.3)



```
    x= max(8+2,2) ;  
    printf("Il massimo è %d \n",x) ;  
    return 0 ;  
}
```

```
int max (int a, int b) {  
    int tmp ;  
    tmp = (a < b) ? b : a  
    return tmp ;  
}
```



# Esempio di funzione: il frame (.4)

⇒ `x = max(8+2, 2);`

`XX: printf("Il massimo è %d \n", x);`

`return 0;`

`}`

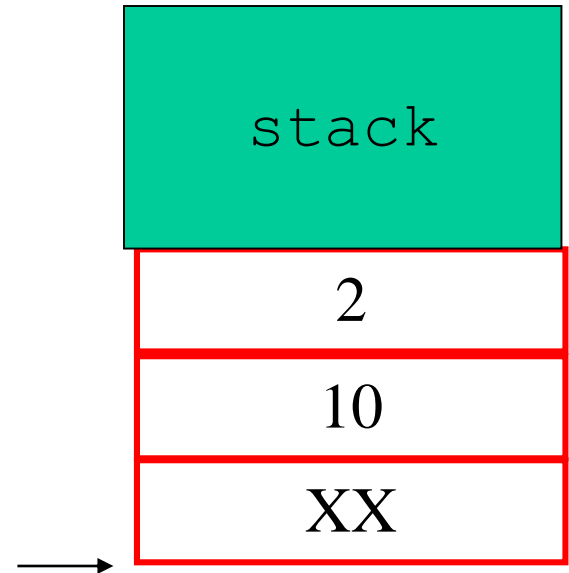
```
int max (int a, int b) {
```

```
    int tmp;
```

```
    tmp = (a < b) ? b : a
```

```
    return tmp;
```

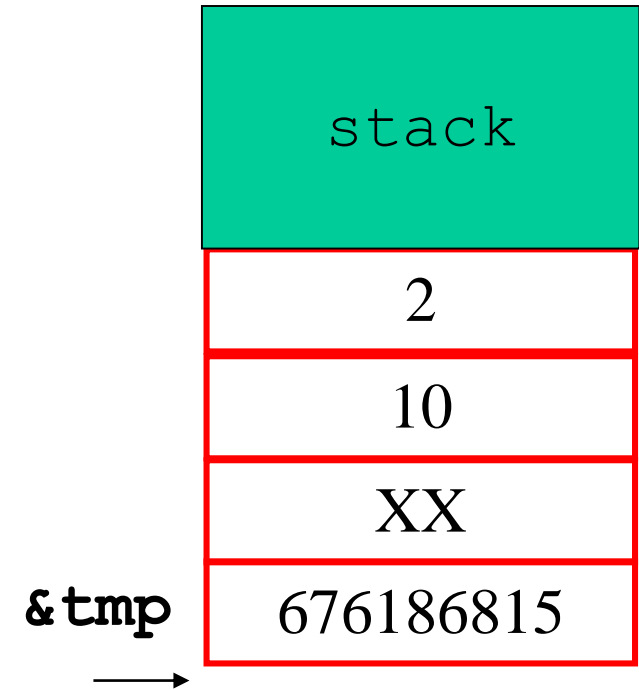
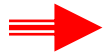
```
}
```



# Esempio di funzione: il frame (.5)

```
x= max(8+2,2) ;  
XX: printf("Il massimo è %d \n",x) ;  
return 0 ;  
}
```

```
int max (int a, int b) {  
    int tmp;  
    tmp = (a < b) ? b : a  
    return tmp;  
}
```

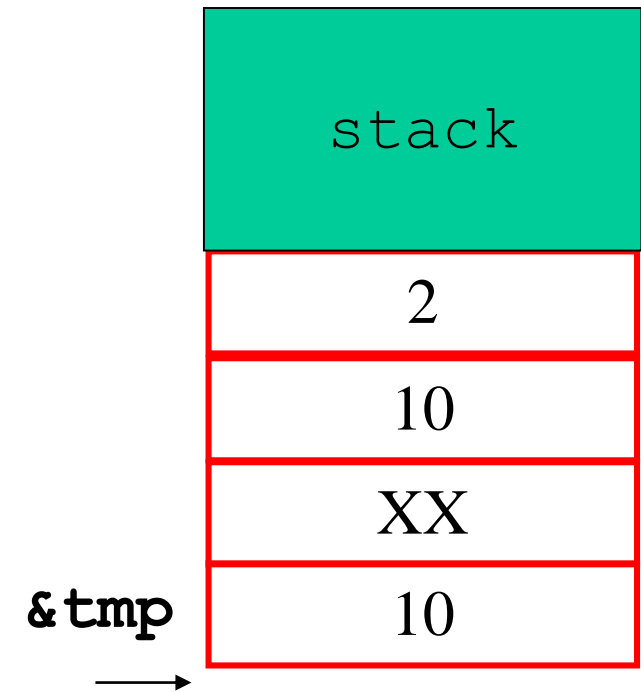




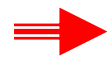
# Esempio di funzione: il frame (.6)

```
    x= max(8+2,2) );  
XX: printf("Il massimo è %d \n",x) ;  
    return 0;  
}
```

```
⇒ int max (int a, int b) {  
    int tmp;  
    tmp = (a < b)? b : a  
    return tmp;  
}
```

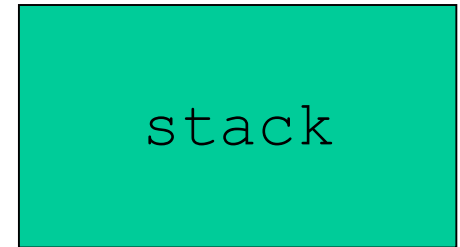


# Esempio di funzione: il frame (.7)



```
x= max(8+2,2) ;  
printf("Il massimo è %d \n",x)  
return 0;  
}
```

```
int max (int a, int b) {  
    int tmp;  
    tmp = (a < b)? b : a  
    return tmp;  
}
```



# Alcune considerazioni

- Usare le variabili locali invece che dinamiche (malloc etc) ovunque è possibile
  - Se si conosce la size e vengono usate solo nella funzione
- Pericolosissimo ritornare i puntatori alle var locali fuori dalla funzione
  - Le aree di memoria che hanno vita fuori dalla funzioni devono stare sullo heap (malloc etc)
- Le modifiche ai parametri della funzione non hanno alcun effetto all'esterno