

# FUNZIONI

$$Y = f(X)$$

↓  
VAR.  
DIPENDENTE  
(OUTPUT)

↑  
VAR.  
INDEPENDENT  
(INPUT)

ES.

$$y = x^2$$

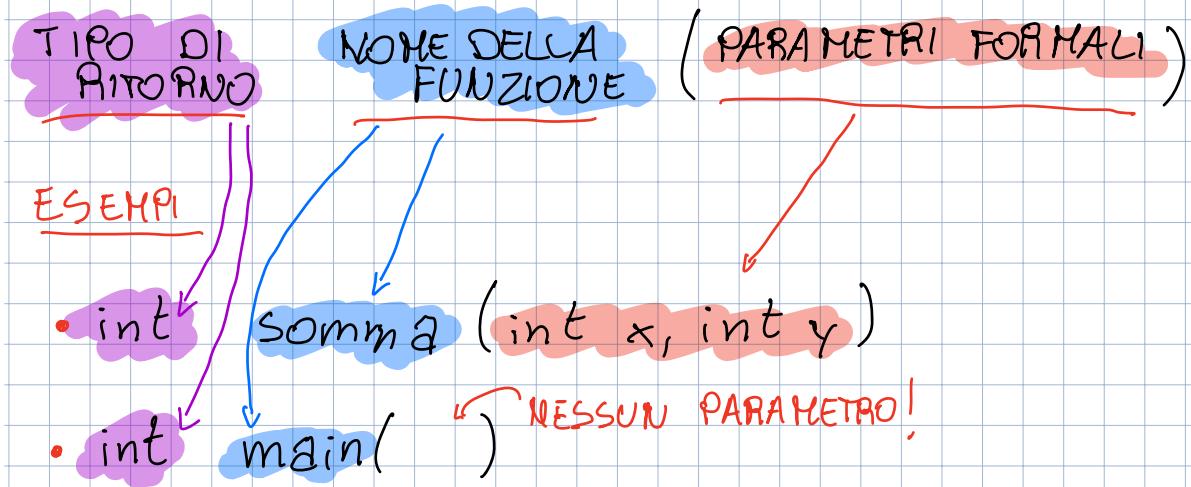
$$\text{SE } x = 3 \rightarrow y = 9$$

$$x = 5 \rightarrow y = 25$$

```
#include
// LIBRERIE
;
// EVENTUALI VARIABILI GLOBALE
// DICHIARAZIONE & PROTOTIPO DELLE FUNZIONI
```

```
int main() {
    ;
    // CHIAMATE DI FUNZIONI
    ;
    return(0);
}
// DEFINIZIONE DI FUNZIONE
```

# FUNZIONI IN C.



```
#include <stdio.h>
int somma (int x, int y); N.B.
```

```
int main( ) {
    int a=6;
    int b=5;
    int sum = somma (a, b); ASSEGNAMENTO // CHIAMA DI FUNZIONE
    printf ("%d\n", sum);
}
```

PASSAGGIO DI VALORE:  
COPIARE IL VALORE DI UNA  
VARIABILE ALL'INTERNO DI  
UN PARAMETRO FORMALE  
DI UNA FUNZIONE

*N.B. L'ORDINE  
DELLE VARIABILI  
NELLA CHIAMATA  
DI FUNZIONE  
E' IMPORTANTE!*

*N.B. IL VALORE NELLA  
VARIABILE a VIENE  
[COPIATO] DENTRO LA  
VARIABILE x. IL VALORE  
NELL'VARIABILE b  
VIENE [COPIATO] DENTRO  
LA VARIABILE y.*

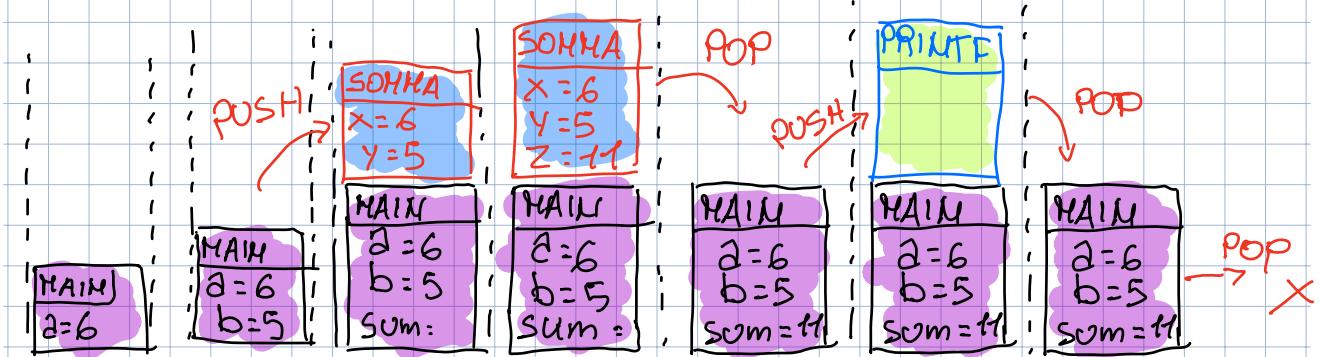
```

int somma(int x, int y) {
    int z = x + y;
    return (z);
}

```

### STACK

FUNZIONE DI LIBRERIA



## FUNZ. INCREMENTO

```
int increments (int x);
```

```
void incremento2 (int x); ← NON DA RETURN
```

```
int main () {
```

```
    int x = 5;
```

```
    x = increments (x);
```

```
    printf ("%d\n", x); // STAMPA 6
```

```
    incremento2 (x); ← NON VIENE CAMBIATA LA VARIABILE  
    x NEL MAIN()
```

```
    printf ("%d\n", x); // STAMPA 6
```

```
}
```

```
int increments (int x) {
```

```
    x++;
```

```
    return x;
```

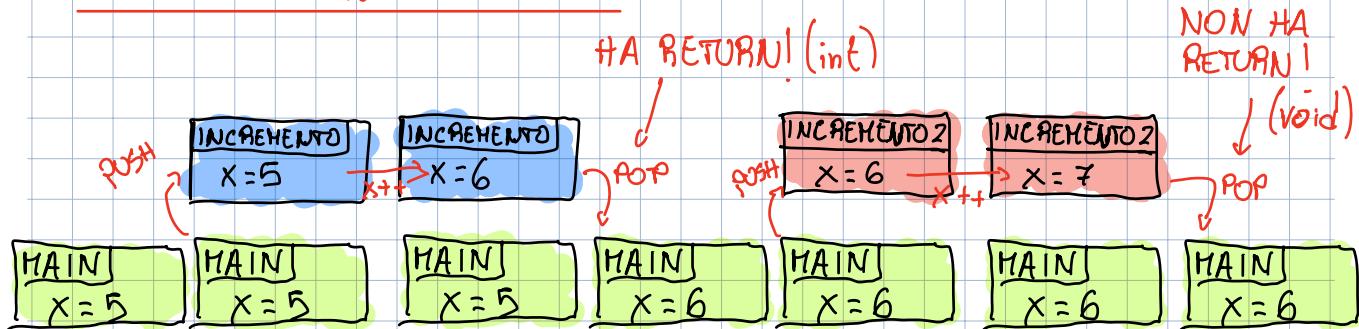
```
}
```

```
void incremento2 (int x) {
```

```
    x++;
```

```
}
```

## STACK ES. PAGINA PRECEDENTE



## USO DEI PUNTATORI NELLE FUNZIONI

### FUNZIONI CON PASSAGGIO DI RIFERIMENTO

```
int increments (int x);
void incremento2 (int *x);
```

↑  
PUNTATORE

```
int main () {
```

```
    int x = 5;
```

```
    x = increments (x);
```

```
    printf ("%d\n", x); // 6
```

```
    incremento2 (&x);
```

```
    printf ("%d\n", x); // 7
```

```
}
```

```
int increments (int x) {
```

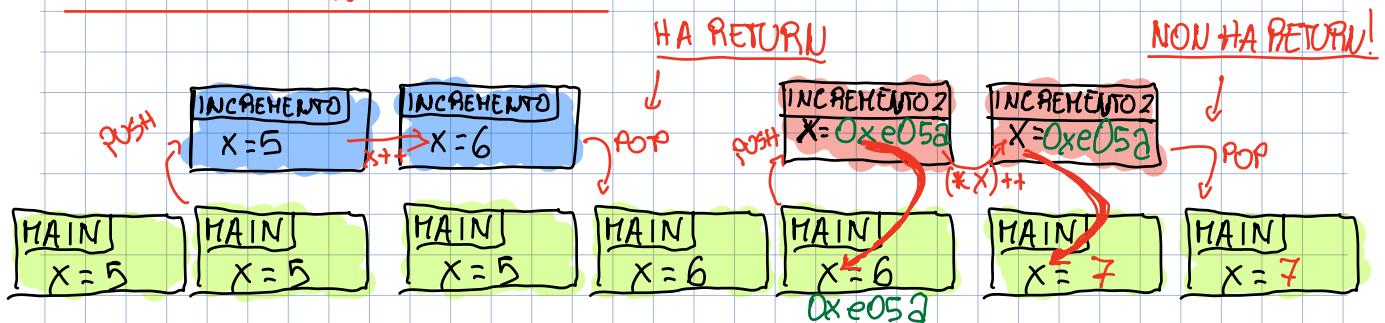
```
    x++;

```

```
    return x;
```

```
void incremento2 (int *x) {
    (*x)++;
}
```

## STACK ES. PAGINA PRECEDENTE



## LO SWAP

```
int main() {
```

```
    int a=5, b=6;    a [ 5 ]   b [ 6 ]
```

```
    int tmp;      tmp [ ]
```

```
    tmp = a;
```

```
    a = b;
```

```
    b = tmp;
```

```
}
```

} LO SWAP

```
tmp [ 5 ]   a [ 5 ]   b [ 6 ]
```

```
tmp [ 5 ]   a [ 5 6 ]   b [ 6 ]
```

```
tmp [ 5 ]   a [ 5 6 ]   b [ 6 5 ]
```

↑      ↑

E' AVVENUTO LO  
SCAMBIO!

```
— . — . — . — . — . — .
```

```
int main() {
```

```
    int a=5, b=6;
```

```
    int a = swap(a,b);
```

```
}
```

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

```
    int tmp;
```

```
    tmp = a;
```

```
    a = b;
```

```
    b = tmp;
```

```
    return a;
```

N.B. POSSO RESTITUIRE UN SOLO

VALORE! NON POSSO FAR E  
LO SCAMBIO!

## SWAP CON PUNTATORI

```
void SWAP( int *a, int *b);  
int main() {  
    int a=5, b=6;  
    swap(&a, &b); // CHIAMATA DI FUNZIONE  
}
```

```
void SWAP( int *a, int *b) {
```

```
    int tmp;
```

```
    tmp = *a;
```

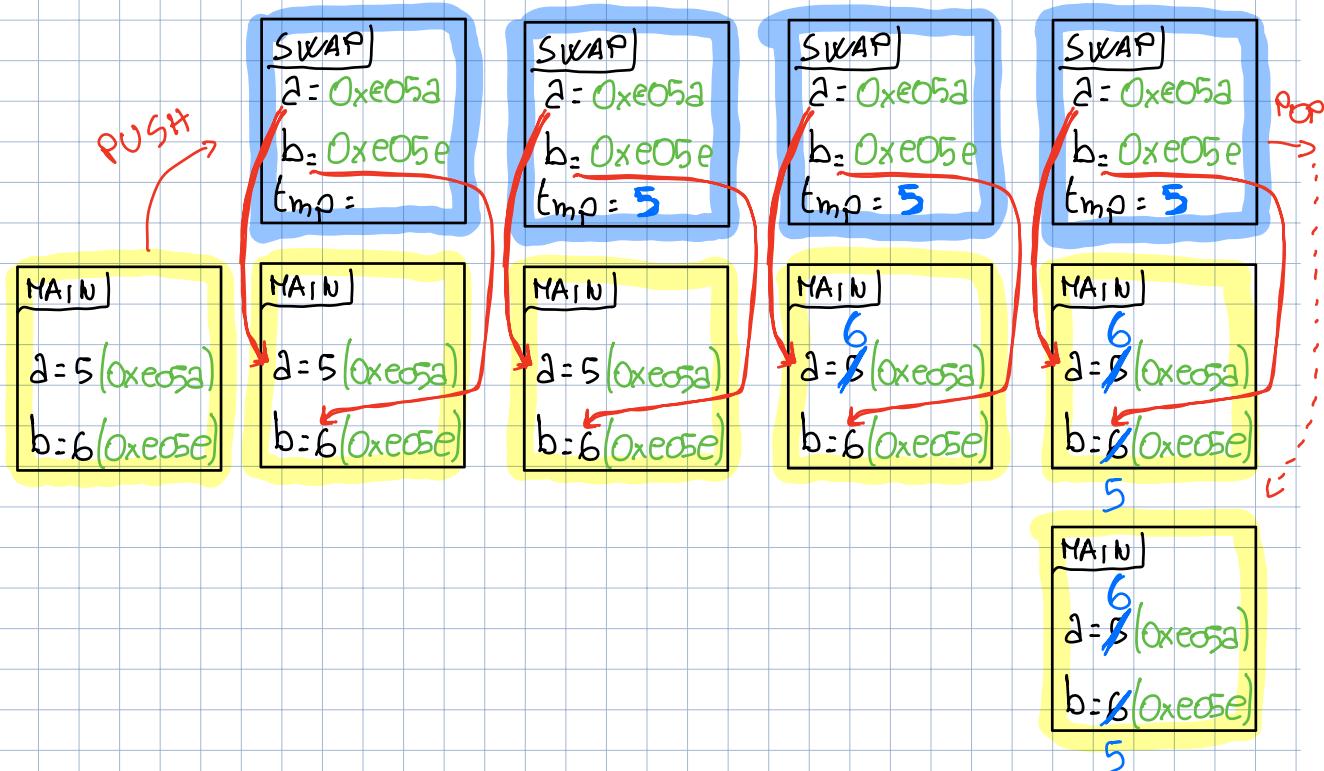
```
    *a = *b;
```

```
    *b = tmp;
```

```
}
```

PUSH

POP



## ES. CARATTERE MAIUSCOLO

```
void inserisci (char *var);
```

```
void maiuscolo (char *let);
```

```
int main () {
```

```
    char let;
```

```
    inserisci (&let); // char *var = &let;
```

```
    printf ("%c", let);
```

```
    maiuscolo (&let);
```

```
    printf ("%c", let);
```

```
}
```

```
void inserisci (char *var) {
```

```
    do {
```

```
        printf ("Inserisci una lettera minuscola ");
```

```
        scanf ("%c", var); // s(*var)
```

```
    } while ((*var) < 'a' || (*var) > 'z');
```

```
}
```

```
void maiuscolo (char *let) {
```

```
    (*let) = (*let) - 32; // (*let) -= 32;
```

```
}
```

## STACK ES. PRECEDENTE

