

# Definizione di funzione in C

```
int somma3(int x, int y)
{ int z = 1;
  x++; y++;
  return x+y+z;
}
```

- Nella parte **<corpo>** possono essere presenti definizioni e/o dichiarazioni locali (**parte dichiarazioni**) e un insieme di istruzioni (**parte istruzioni**).
- I dati riferiti nel corpo possono essere **costanti**, **variabili**, oppure **parametri formali**.
- All'interno del corpo, i **parametri formali** vengono trattati come **variabili**.

# Chiamata di Funzione

La chiamata di funzione è un'espressione della forma:

```
somma3(45, 26);  
somma3(24, a); /* a di tipo int */
```

dove:

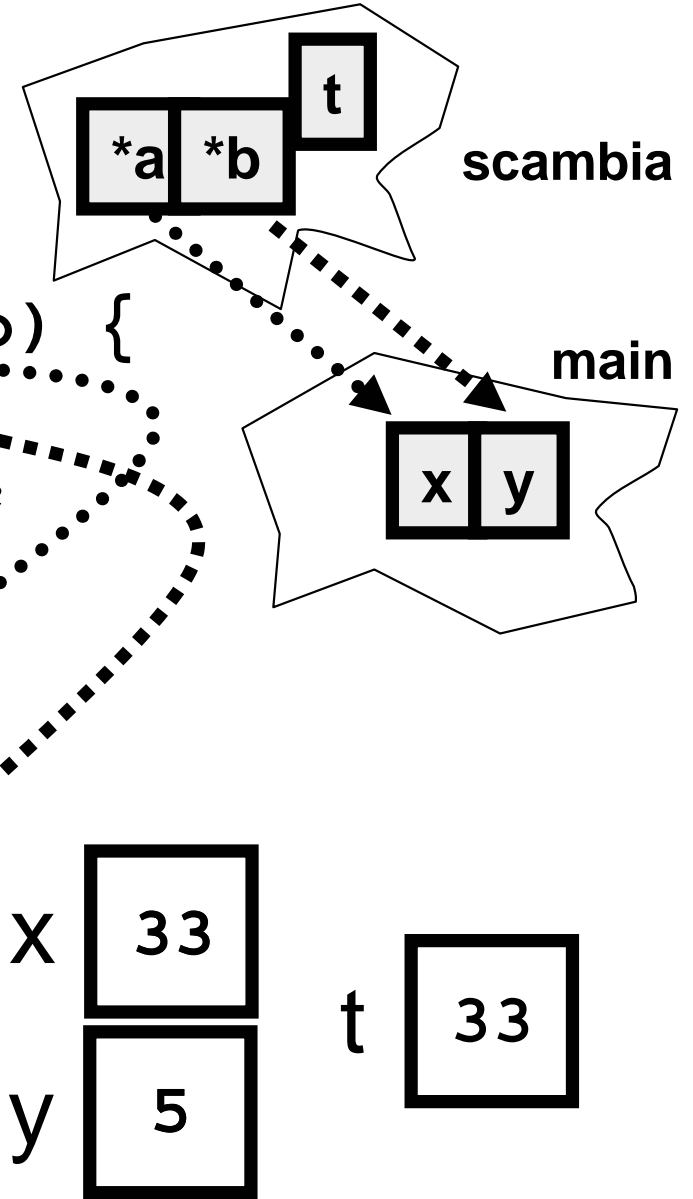
```
<parametri-effettivi> ::=  
[ <espressione> ] { , <espressione> }
```

# Esempio: scambio di valori tra variabili

variabile d'appoggio

```
void scambia(int* a, int* b) {  
    int t;  
    t = *a; *a = *b; *b = t;  
}
```

```
main() {  
    int y = 5, x = 33;  
    scambia(&x, &y);  
}
```



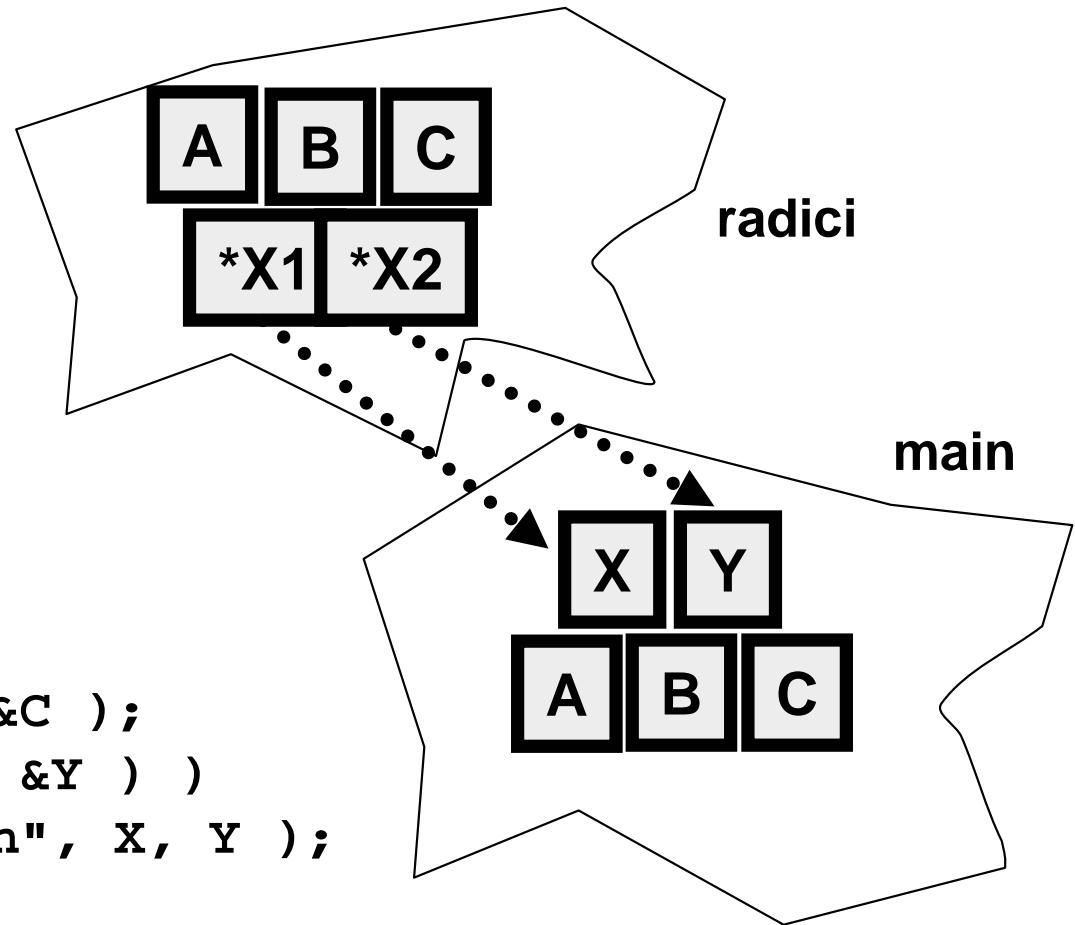
# Equazione di secondo grado: $Ax^2 + Bx + C = 0$

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

```
#include <math.h>
```

```
int radici(float A, float B, float C, float *X1, float *X2)
{
    float D;
    if ( !A ) return 0;
    D = B*B-4*A*C;
    if ( D<0 ) return 0;
    D=sqrt(D);
    *X1 = (-B+D)/(2*A);
    *X2 = (-B-D)/(2*A);
    return 1;
}
```

```
main()
{
    float A,B,C,X,Y;
    scanf( "%f%f%f", &A, &B, &C );
    if ( radici( A, B, C, &X, &Y ) )
        printf( "%f%f\n", X, Y );
}
```



# ESERCIZIO: Ordinamento di un vettore

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

```
#define N 5
```

```
int leggi(int a[], int dim) {...}; /* lettura dati */
```

```
void stampa(int a[], int dim) {...}; /* stampa */
```

```
void ordina (int a[], int dim) {...}; /* ordina */
```

```
void scambia(int* a, int* b) /* ..vista prima.. */
```

```
main () {
```

```
    int i, n, a[N];
```

```
    n=leggi(a, N);
```

```
    ordina(a, n);
```

```
    stampa(a, n);
```

```
}
```

# Esempio

```
#include <stdio.h>
main()
{int i=0;
  while (i<=3)
  { /* BLOCCO 1 */
    int j=4; /* def. locale al blocco 1*/
    j=j+i;
    i++;

    { /* BLOCCO 2: interno al blocco 1*/
      float i=j; /*locale al blocco 2*/
      printf("%f\t%d\t",i,j);
    }
    printf("%d\t\n",i);
  }
}
```