

# Procedure in C

## Titolo unità didattica: Function e Procedure

[05]

### Titolo modulo : Procedure in C

[04-C]

Function C che restituiscono più valori

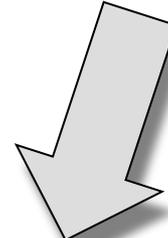
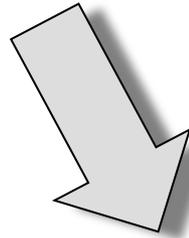
Argomenti trattati:

- ✓ **void function** in C
- ✓ passaggio dei parametri per riferimento simulato
- ✓ esempi di function C che implementano procedure

Prerequisiti richiesti: P1-03-04-T, P1-05-02-T, P1-05-03-T

function

procedure



function

in C una procedura viene realizzata  
mediante una function

una function C può avere un **numero  
qualunque** di valori da restituire

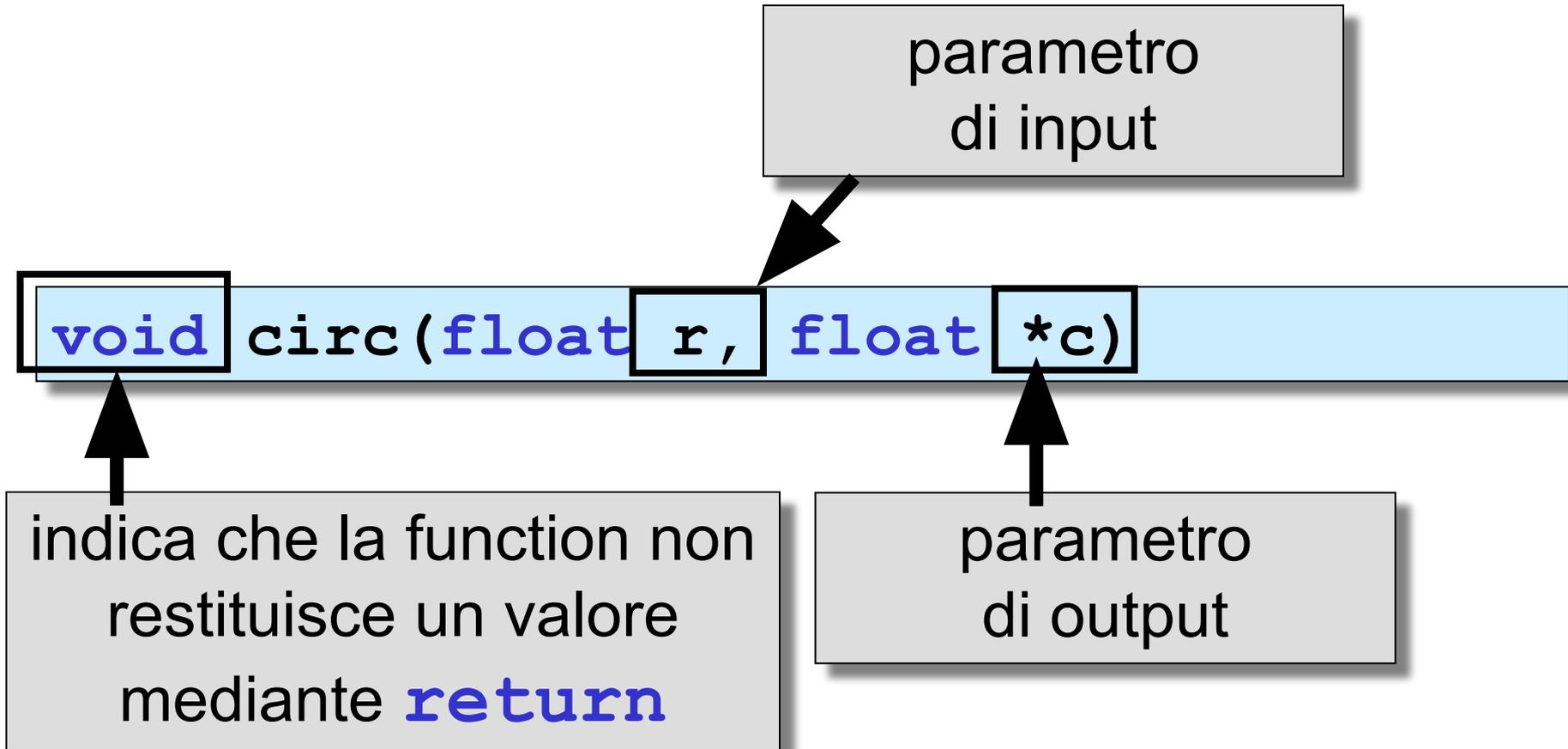
## void function in C

```
void circ(in: float r; out: float c)
{
  const float pi_greco = 3.1415926;
  c = 2.0*pi_greco*r ;
}
```

Attenzione: da modificare in C

```
void circ(float r, float *c)
{
  const float pi_greco = 3.1415926F;
  *c = 2.0F*pi_greco*r;
}
```

# intestazione di una `void` function



**i parametri di output devono essere puntatori**

# chiamata di una `void` function

```
main {  
float raggio, circonferenza;  
  read (raggio) ;  
  circ(raggio, circonferenza) ;  
  printf (circonferenza) ;  
}
```

Attenzione: da modificare in C

```
#include<stdio.h>  
void circ(float r,float *c) ;  
void main ()  
{  
  float raggio, circonferenza ;  
  printf("inserire il raggio: ") ;  
  scanf ("%f", &raggio) ;  
  circ(raggio, &circonferenza) ;  
  printf("circonferenza =%f\n", circonferenza) ;  
}
```

chiamata di una **void function**

```
circ(raggio, &circonferenza)
```

argomento  
di input

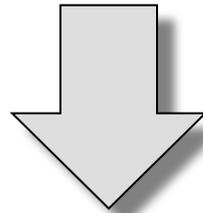
argomento  
di output

**gli argomenti di output devono essere indirizzi**

**operatore indirizzo di (&)**  
**indirizzo** di una variabile  
**puntatore** alla variabile

```
void circ(float r, float *c)
```

il parametro di output è un  
**puntatore** (a un **float**)



l'argomento corrispondente (output)  
deve essere un  
**puntatore** a un **float**  
oppure  
un **indirizzo** di un **float**

```
#include <stdio.h>
void circ(float r, float *c) ;
void main ()
{
    float raggio, circonferenza;
    scanf ("%f", &raggio) ;
    circ (raggio, &circonferenza) ;
    printf ("circonferenza=%f\n", circonferenza) ;
}
```

```
#include <stdio.h>
void circ(float, float *) ;
void main ()
{
    float raggio, circonferenza, *punt;
    punt = &circonferenza;
    scanf ("%f", &raggio) ;
    circ (raggio, punt) ;
    printf ("circonferenza=%f\n", circonferenza) ;
}
```

# il passaggio dei parametri in C è solo per valore

parametri di input: OK

parametri di output: ?

il passaggio dell'**indirizzo** di una variabile consente di accedere in **modo indiretto** alla variabile

```
#include<stdio.h>
void circ(float r, float *c);
void main ()
{
    float raggio, circonferenza;
    printf("inserire il raggio: ");
    scanf("%f", &raggio);
    circ(raggio, &circonferenza);
    printf("circonferenza
           =%f\n", circonferenza);
}
```

```
void circ(float r, float *c)
const float pi_greco =
    3.1415926F;
{
    *c = 2.0F*pi_greco*r;
}
```

il puntatore **c** punta alla variabile **circonferenza**

il passaggio dei parametri in C è solo per valore

**chiamante**

**function chiamata**

argomento: **indirizzo**

parametro: **puntatore**



**&variabile**    puntatore

passaggio dei parametri per  
**riferimento simulato**

procedura per lo scambio del valore di due variabili di tipo intero

```
void scambiare_i(input: int var1, int var2)
{ int var1, var2, temp;
  temp = var1 ;
  var1 = var2 ;
  var2 = temp ;
}
```

Attenzione: da modificare in C

```
main {
  int alfa, beta;
  read (alfa, beta) ;
  scambiare_i(alfa, beta) ;
  printf (alfa, beta) ;
}
```

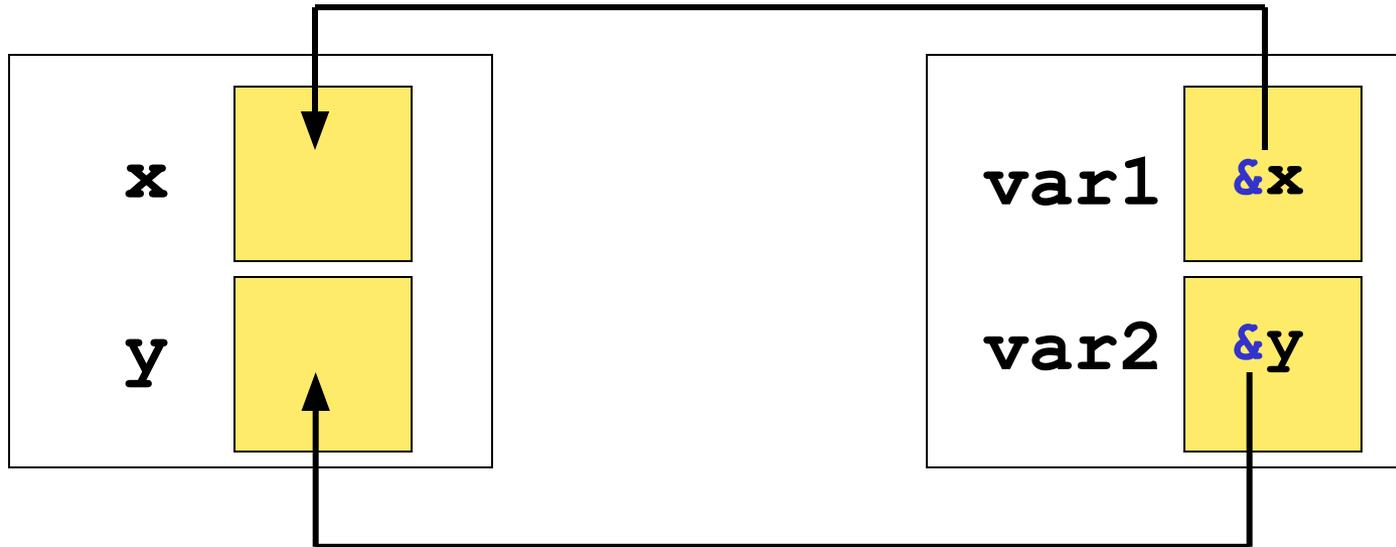
Attenzione: da modificare in C

```
#include <stdio.h>
void scambiare_i(int var1, int var2);
void main()
{
    int x, y;
    scanf ("%d%d", &x, &y);
    printf ("x=%d y=%d", x, y);
    scambiare_i(x, y);
    printf ("x=%d y=%d", x, y);
}
void scambiare_i(int var1, int var2)
{
    int temp;
    temp = var1;
    var1 = var2;
    var2 = temp;
}
```

```
#include <stdio.h>
void scambiare_i(int *var1, int *var2) ;
void main()
{
    int x,y;
    scanf ("%d%d" , &x, &y) ;
    printf ("x=%d y=%d" , x, y) ;
    scambiare_i (&x, &y) ;
    printf ("x=%d y=%d" , x, y) ;
}
void scambiare_i(int *var1, int *var2)
{
    int temp;
    temp = *var1;
    *var1 = *var2;
    *var2 = temp;
}
```

main

scambiare\_i



`*var1` è il valore di `x`

`*var2` è il valore di `y`

```
void main()
```

```
{
```

```
int x,y;
```

```
scanf ("%d%d" , &x , &y) ;
```

```
printf ("x=%d y=%d" , x , y) ;
```

```
scambiare i (&x , &y) ;
```

```
printf ("x=%d y=%d" , x , y) ;
```

```
}
```

```
void main()
```

definizione di **main** come **void** function,  
senza parametri

```
scanf ("%d%d" , &x , &y) ;
```

**scanf** è una function, chiamata  
con **1** argomento di **input** e  
**2** argomenti di **output**

```
printf ("x=%d y=%d" , x , y) ;
```

**printf** è una function,  
chiamata con **3** argomenti di  
**input**

# Esercizio 1

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

```
/** creare una funzione che elevi al quadrato un numero dato in input  
    passare in input l'indirizzo del numero e non il valore  
*/
```

# Esercizio 1

```
>#include <stdio.h>

int elev_potenza(int n);
void elev_potenza2(int *n);

int main(void) {
    int x, y;
    printf("Inserisci un numero da elevare al quadrato\n");
    scanf("%d", &x);
    y = elev_potenza(x);
    printf("con funzione = %d\n", y);

    elev_potenza2(&x);
    printf("con procedura = %d\n", x);

    return 0;
}

void elev_potenza2(int *n){
    //printf("%p\n", n);
    *n = *n * *n; // equivalente *n *= *n;
}

int elev_potenza(int n){
    int output = n * n ;
    return output;
}
```