

Titolo unità didattica: Function e procedure

[05]

Titolo modulo : Procedure

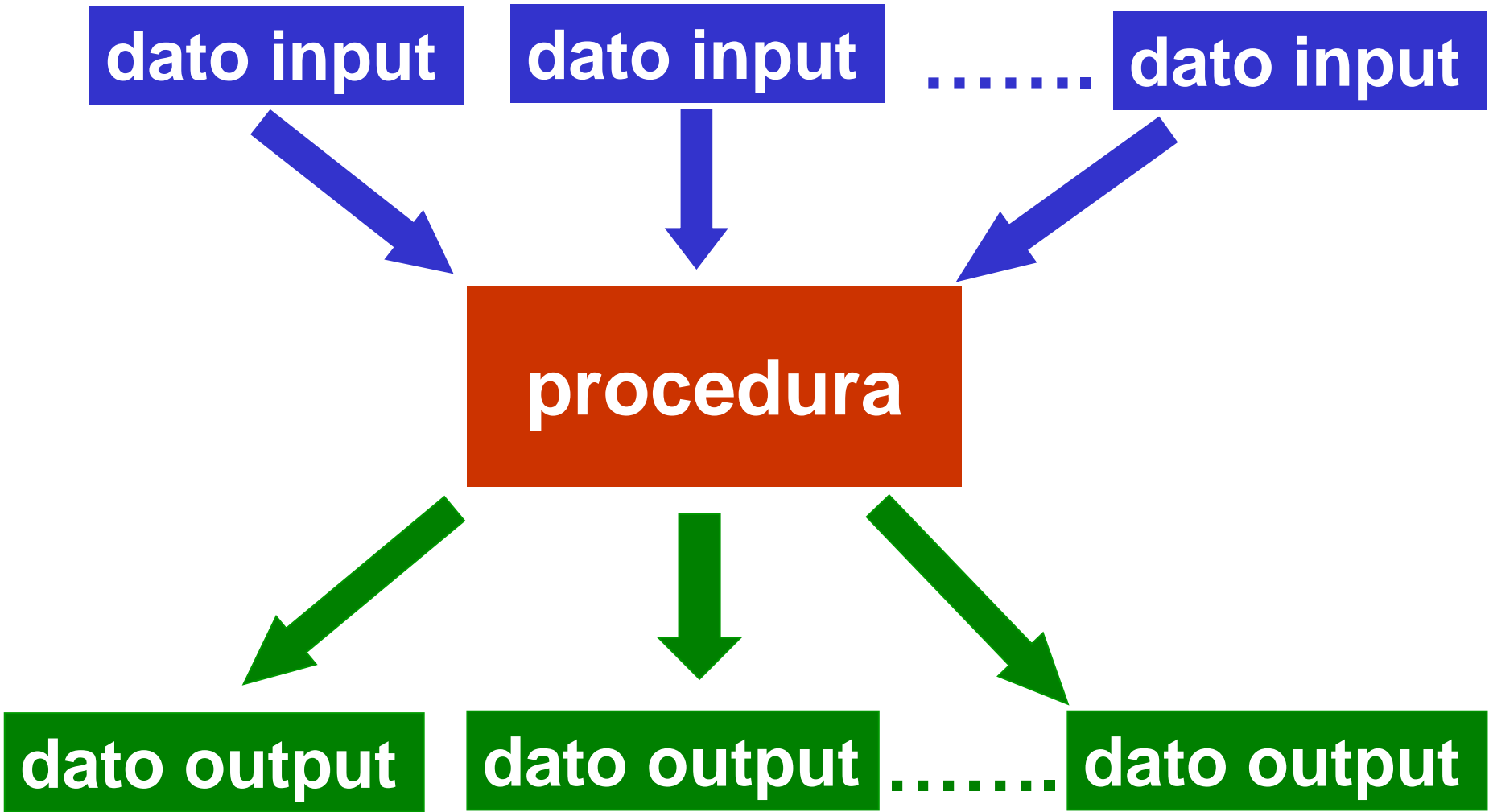
[02-T]

Organizzazione di algoritmi come procedure

Argomenti trattati:

- ✓ procedure
- ✓ parametri di input, parametri di output
- ✓ parametri di input/output

Prerequisiti richiesti: P1-05-01-T



la **intestazione di una procedura** deve specificare

- ✓ il **nome** della procedura
- ✓ i parametri di **input**
- ✓ i parametri di **output**
- ✓ i parametri di **input/output**

```
void <nome>  
( in:    <parametri input>;  
  out:   <parametri output>;  
  inout: <parametri input/output> )
```

ATTENZIONE: **in**, **out**, **inout** **NON** fanno parte del linguaggio C

procedura: intestazione

```
void <nome>  
    (in: <variabili>; out: <variabili>;  
    inout: <variabili>)
```

le variabili che appaiono

- ✓ dopo **in:** sono i **parametri di input**
- ✓ dopo **out:** sono i **parametri di output**
- ✓ dopo **inout:** sono **parametri sia di input sia di output**

ATTENZIONE: **in**, **out**, **inout** **NON** fanno parte del linguaggio **C**

Esempio di intestazione

```
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

- la **procedura circ** ha un unico **parametro di input**, la variabile **r**
- la procedura ha un unico **parametro di output**, la variabile **c**
- non ci sono istruzioni di **I/O** nella procedura coinvolgenti i parametri

Esempio di intestazione

```
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

- la variabile **r** è lasciata indefinita (nessun valore associato) nel corpo della procedura
- la procedura **deve** contenere una istruzione che assegni un valore al parametro di uscita (dato di output), cioè alla variabile **c**
- **non** ci deve essere **nessuna** istruzione **return**

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

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

in C i **parametri di input** sono passati per **valore** (copia)

in C i **parametri di output** e (**inout**) sono passati per **riferimento** (indirizzo)

Esempio di chiamata

Programma di esempio di `usa_circ`


```
float raggio, circonferenza;
```

```
read (raggio) ;
```

```
circ(raggio, circonferenza) ;
```

```
printf (circonferenza) ;
```

attivazione
della
procedura



ATTENZIONE: da modificare in C

- il valore della variabile **raggio** è l'**argomento di input** da associare al **parametro di input r**
- l'argomento di output **circonferenza** riceve il valore associato al **parametro di output c**, al termine dell'esecuzione della procedura

ATTENZIONE: è una SEMPLIFICAZIONE (errata) dell'attivazione in linguaggio C

Programma di esempio di `usa_circ`

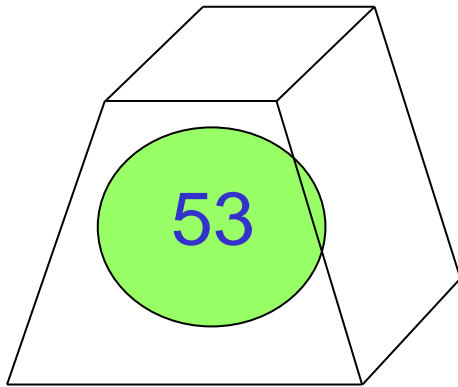
```
float raggio, circonferenza;  
read (raggio) ;  
circ(raggio, circonferenza) ;  
printf (circonferenza) ;
```

```
#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) ;  
}
```

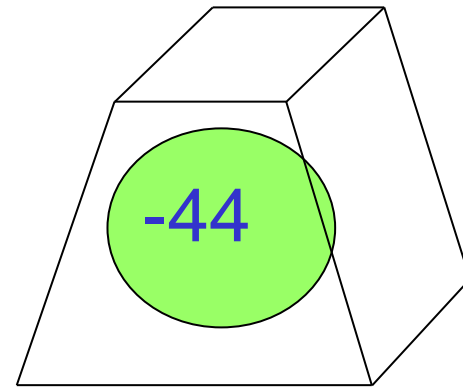
Problema:

algoritmo per lo scambio del valore di due variabili
(di tipo intero)

alfa, **beta** di tipo **int**



alfa



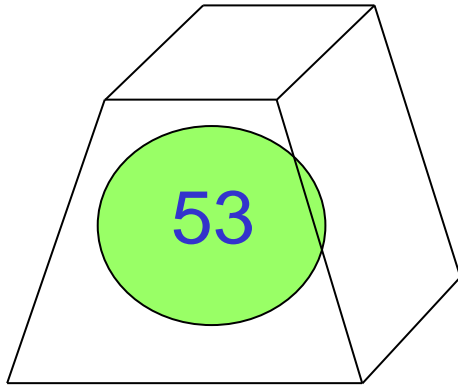
beta

Problema:

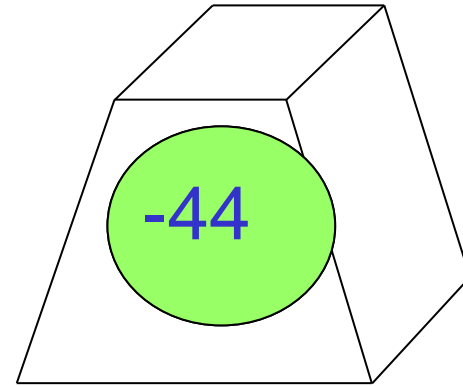
algoritmo per lo scambio del valore di due variabili
(di tipo intero)

alfa = beta

beta = alfa



alfa



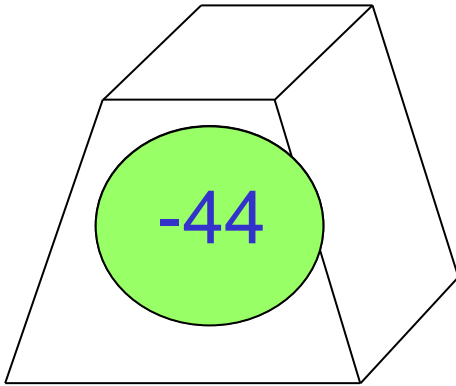
beta

Problema:

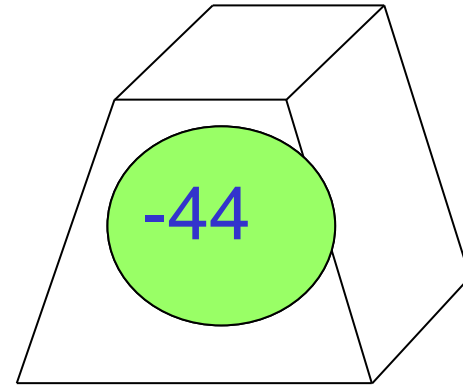
algoritmo per lo scambio del valore di due variabili
(di tipo intero)

alfa = beta

beta = alfa



alfa



beta

Problema:

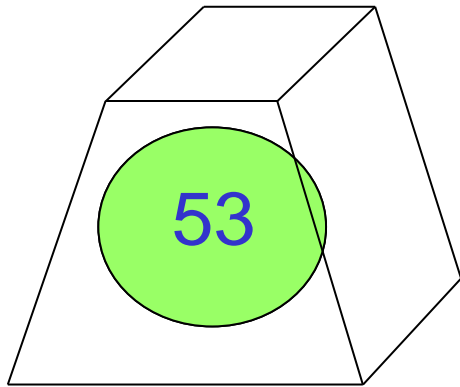
algoritmo per lo scambio del valore di due variabili
(di tipo intero)

```
temp = alfa
```

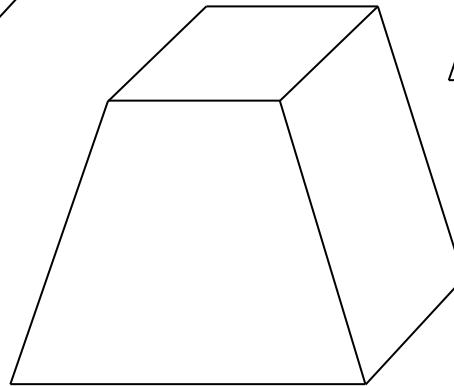
```
alfa = beta
```

```
beta = temp
```

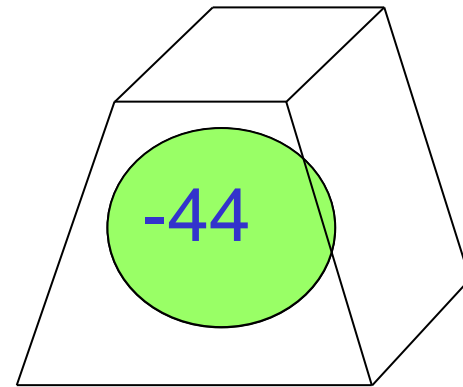
alfa, beta, temp
di tipo **int**



alfa



temp



beta

Problema:

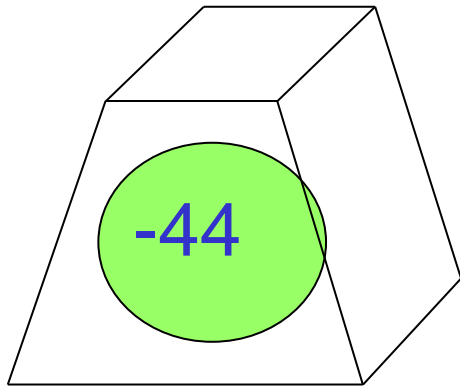
algoritmo per lo scambio del valore di due variabili
(di tipo intero)

temp = alfa

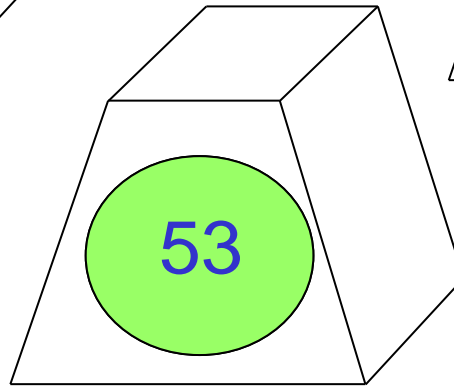
alfa = beta

beta = temp

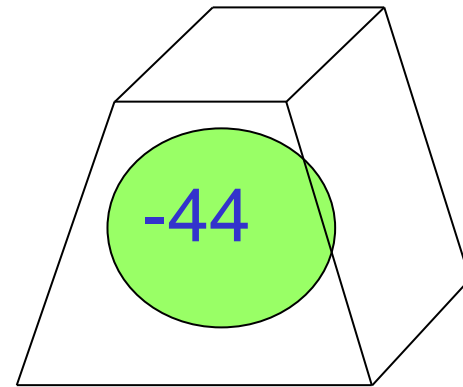
alfa, beta, temp
di tipo **int**



alfa



temp



beta

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

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

ATTENZIONE: da modificare in C

- la procedura **scambiare_i** ha due **parametri di input/output**, le variabili **var1** e **var2**
- la procedura usa la variabile (locale) **temp**
- la procedura modifica i valori associati ai parametri di input/output **var1, var2**

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

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

ATTENZIONE: da modificare in C

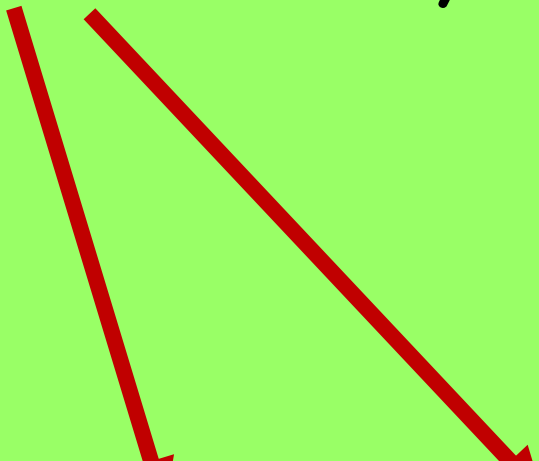
Programma che usa scambiare_i

```
int alfa, beta;
read (alfa, beta);
scambiare_i (alfa, beta);
printf (alfa, beta);
```

ATTENZIONE: da modificare in C

**ATTENZIONE: è una SEMPLIFICAZIONE (errata)
dell'attivazione in linguaggio C**


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



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

Programma che usa scambiare_i

```
int alfa, beta;  
read (alfa, beta) ;  
scambiare_i (alfa, beta);  
printf (alfa, beta) ;
```

```
#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) ;  
}
```

```
#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;
}
```