

Assegnamento

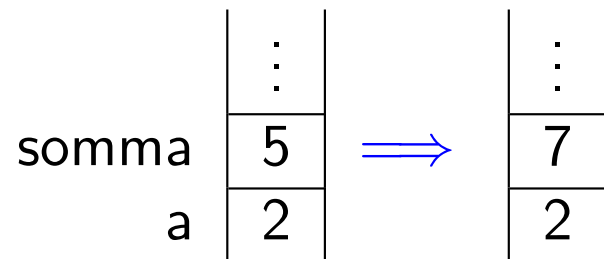
- Ricordiamo che l'esecuzione di `x = exp` corrisponde a:
 - 1 valutare il valore dell'espressione `exp` a destra di "=" (usando i valori correnti delle variabili);
 - 2 assegnare **poi** tale valore alla variabile `x` a sinistra di "=".

Esempio:

```
somma = 5;
```

```
a = 2;
```

```
somma = somma + a;
```



Esempio:

	a	b
<code>int a, b;</code>	?	?
<code>a = 2;</code>	2	?
<code>b = 3;</code>	2	3
<code>a = b;</code>	3	3
<code>a = a + b;</code>	6	3
<code>b = a + b;</code>	6	9

Osservazioni sull'assegnamento

- **Attenzione:** A sinistra di “=” ci deve essere un identificatore di **variabile**

⇒ denota la corrispondente associazione modificabile nello stato.

Esempio: Quali istruzioni sono corrette e quali no?

$a = a;$

SI corretta (anche se poco significativa ...)

$a = 2 * a;$

SI corretta (il valore associato ad a si raddoppia)

$5 = a;$

NO, 5 non denota una associazione modificabile nello stato ma un valore costante

$a + b = c;$

NO, $a+b$ è un'espressione, non una variabile!

Esempi di algoritmi

Vediamo dapprima due esercizi, in cui ci concentriamo sullo stato.

Esempio: Ordinare tre valori interi distinti tra loro

Stato iniziale: $\{x \rightsquigarrow A, y \rightsquigarrow B, z \rightsquigarrow C\}$ con A, B, C distinti tra loro

Stato finale: $\left\{ \begin{array}{l} x \rightsquigarrow \max(\{A, B, C\}), \\ y \rightsquigarrow \max(\{\{A, B, C\} \setminus \{\max(\{A, B, C\})\}), \\ z \rightsquigarrow \min(\{A, B, C\}) \end{array} \right\}$

Algoritmo

Passo 1. “Ordiniamo” x e y , portandoci nello stato intermedio

Stato 1: $\{x \rightsquigarrow \max(\{A, B\}), y \rightsquigarrow \min(\{A, B\}), z \rightsquigarrow C\}$

Passo 2. “Ordiniamo” x e z , portandoci nello stato intermedio

Stato 2: $\left\{ \begin{array}{l} x \rightsquigarrow \max(\{A, B, C\}), y \rightsquigarrow \min(\{A, B\}) \\ z \rightsquigarrow \min(\max(\{A, B\}), C) \end{array} \right\}$

Passo 3. “Ordiniamo” y e z , portandoci nello stato finale desiderato

Cuore della soluzione

```
if (x < y)
{
    temp = x;
    x = y;
    y = temp;
}
{ x  $\rightsquigarrow$  max({A,B}), y  $\rightsquigarrow$  min({A,B}), z  $\rightsquigarrow$  C }
if (x < z)
{
    temp = x;
    x = z;
    z = temp;
}
{ x  $\rightsquigarrow$  max({A,B,C}), y  $\rightsquigarrow$  min({A,B}), z  $\rightsquigarrow$  min(max({A,B}), C) }
if (y < z)
{
    temp = y;
    y = z;
    z = temp;
}
```

Stato finale

Esempio: Calcolare il massimo di tre interi.

Stato iniziale: $\{x \rightsquigarrow A, y \rightsquigarrow B, z \rightsquigarrow C\}$

Stato finale: $\{\text{massimo} \rightsquigarrow \max(\{A,B,C\})\}$

Cuore della soluzione 1 (migliore)

```
if (x > y)
    massimo = x;
else
    massimo = y;
    { max  $\rightsquigarrow$  max({A,B}) }
if (z > massimo)
    massimo = z;
```

Stato finale

Cuore della soluzione 2

```
if (x > y)
    if (x > z)
        massimo = x;
    else
        massimo = z;
else
    if (y > z)
        massimo = y;
    else
        massimo = z;
```

Vediamo ora due esercizi e la loro soluzione con tutti i dettagli. **Esempio:** Calcolare il massimo di N interi dati da linea di comando, con N dato da linea di comando

- Dobbiamo calcolare il massimo mano a mano che gli interi vengono inseriti
- Utilizziamo un ciclo controllato da una variabile di controllo i che assume tutti i valori fino a N .
- Facciamo in modo che, ad ogni iterazione, il massimo tra i valori fino a quel punto inseriti. Per ottenere questo a ogni iterazione confrontiamo l'elemento corrente con il "massimo in carica".
- Alla fine del ciclo abbiamo il massimo tra tutti i valori.

Soluzione:

```
#include <stdio.h>
int main()
{
    int n,corr;
    int max;
    printf("Introduci il numero di interi\n");
    scanf("%d", &n);
    printf("Introduci il primo intero della sequenza\n");
    scanf("%d", &max);
    while (n - 1 > 0)
    {
        printf("Introduci il nuovo intero della sequenza\n");
        scanf("%d", &corr);
        if (corr > max)
            max = corr;
        n = n-1;
    }
    printf("Il massimo e': %.d\n",max);
    return 0; }
```

Per calcolare il massimo di K elementi devo fare $K-1$ confronti, dato che sono proprio $K-1$ gli elementi che devono uscire perdenti.

Esempio: Calcolo del fattoriale di un numero naturale. Ricordiamo che:

$$n! = \begin{cases} 1 & \text{se } n = 0 \\ 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n & \text{se } n > 0 \end{cases}$$

Specifica:

Stato iniziale: $\{n \rightsquigarrow N\}$ con $N \geq 0$

Stato finale: $\{n \rightsquigarrow N, \text{fatt} \rightsquigarrow N!\}$

- Attenzione: la specifica impone che il valore di n non deve cambiare!