

**Esercizio** Controllare se un valore dato appare in una sequenza data

Stato iniziale:  $\{\text{dim} \rightsquigarrow K, \text{val} \rightsquigarrow V, c[0] \rightsquigarrow V_0, \dots, c[K-1] \rightsquigarrow V_{K-1}\}$

Stato finale:  $\{\text{trovato} \rightsquigarrow b = 1 \text{ se } \exists j \in [0, K-1] \wedge V_j = V, b = 0 \text{ altr.}\}$

**Esercizio** Controllare se un valore dato appare in una sequenza data

Stato iniziale:  $\{\text{dim} \rightsquigarrow K, \text{val} \rightsquigarrow V, c[0] \rightsquigarrow V_0, \dots, c[K-1] \rightsquigarrow V_{K-1}\}$

Stato finale:  $\{\text{trovato} \rightsquigarrow b = 1 \text{ se } \exists j \in [0, K-1] \wedge V_j = V, b = 0 \text{ altr.}\}$

- ▶ Di nuovo, utilizziamo un ciclo controllato da una variabile di controllo  $i$  che assume tutti i valori nell'intervallo  $[0, \text{dim}-1]$ .
- ▶ Devo usare anche una variabile di controllo **trovato** che mi dica se l'elemento è stato trovato oppure no.
- ▶ Dal ciclo si esce quindi quando:
  - ▶ o ho appena trovato l'elemento cercato
  - ▶ o sono arrivato alla fine della sequenza.

```
i = 0; trovato = 0;
while ((i <= dim-1) && (trovato == 0))
{
    if (c[i] == val)
        trovato = 1;
    i = i + 1;
}
```

## Caveat

- ▶ La doppia condizione  $((i \leq \text{dim}-1) \ \&\& \ (\text{trovato} == 0))$  corrisponde al fatto che si può uscire dal ciclo:
  - ▶ se sono arrivato alla fine della sequenza,
  - ▶ o se ho appena trovato l'elemento cercato.
- ▶ usare solo la prima non è scorretto **ma** risulta *inefficiente*. Ad es., in una sequenza di 10.000 elementi in cui il valore si trova nella prima posizione, non fermarsi appena lo si è trovato comporta che si facciano comunque tutti i 9999 confronti.
- ▶ uscire non appena si trova l'elemento cercato si può ottenere anche in altri modi, come ad esempio
  - ▶ forzando il valore del contatore  $i$  ad assumere il valore  $\text{dim}$  per falsificare la prima condizione

In questi i casi tuttavia, viene pregiudicata la leggibilità del codice ed anche minata la possibilità di fare analisi di correttezza. Per questo motivo, sono **fortemente sconsigliati**.

## Esercizi proposti

**Problema 1:** Calcolare il numero di occorrenze di un valore dato in una sequenza data

Stato iniziale:  $\{ \text{dim} \rightsquigarrow K, \text{val} \rightsquigarrow V, c[0] \rightsquigarrow V_0, \dots, c[K-1] \rightsquigarrow V_{K-1} \} \quad K > 0$

Stato finale:  $\{ \text{occ} \rightsquigarrow \#\{ j \mid j \in [0, K-1] \wedge V_j = V \} \}$

**Problema 2:** Calcolare contemporaneamente il massimo e il minimo di una sequenza data di interi

Stato iniziale:  $\{ \text{dim} \rightsquigarrow K, c[0] \rightsquigarrow V_0, \dots, c[K-1] \rightsquigarrow V_{K-1} \}$  con  $K > 0$

Stato finale:  $\{ \text{massimo} \rightsquigarrow \max(\{V_0, \dots, V_{K-1}\}), \text{minimo} \rightsquigarrow \min(\{V_0, \dots, V_{K-1}\}) \}$

**Problema 3:** Calcolare il numero di occorrenze del valore massimo in una sequenza di interi

Stato iniziale: ?

Stato finale: ?

**Problema 4:** Calcolare la media di una sequenza data di interi

Stato iniziale: ?

Stato finale: ?