



POLITECNICO
MILANO 1863



POLITECNICO
MILANO 1863

Fondamenti di Informatica

Esercitazione 10

Paolo Cudrano
paolo.cudrano@polimi.it

08-11-2023

Recap

Recap: sottoprogrammi (funzioni)

Motivazioni

- Riutilizzo del codice
 - Evita la duplicazione (→ incongruenza → inconsistenza)
- Migliore organizzazione del codice
- Collaborazione tra più persone

Importante!

1 funzione risolve 1 problema

(può avere più input e più output/effetti, ma collegati a 1 problema)

Recap: sottoprogrammi (funzioni)

Utilizzo

```
// Prototipo
int somma_numeri(int, int);

// Implementazione
int somma_numeri (int a, int b) {
    int val = a + b;
    return val;
}

// Chiamata (nel main o altra funzione)
int somma, n1=1, n2=5;
somma = somma_numeri(n1, n2);
```

Recap: struttura codice

```
/* Inclusione librerie */  
#include ...
```

```
/* Definizione costanti */  
#define ...
```

```
/* Definizione nuovi tipi */  
typedef ...
```

```
/* Prototipo funzioni */  
<tipo> nome_funzione([<argomenti>]);
```

```
/* Main */  
int main([<argomenti main>]) {  
    // ... implementazione ...  
}
```

```
/* Definizione funzioni */  
<tipo> nome_funzione([<argomenti>]) {  
    // ... implementazione ...  
}
```

Domande?

Esercizio 1: lettura da tastiera controllata

Si definisca una funzione che legga da tastiera un intero in un determinato range, estremi inclusi, e lo ritorni al chiamante. Se l'utente inserisce un valore esterno al range, la funzione richiede di ripetere l'inserimento.

Esercizio 1b (a casa): lettura da tastiera di array

Si definisca una funzione che legga da tastiera un array di interi. La funzione non deve necessariamente riempire interamente l'array.

Esercizio 2: stampa array a video

Si definisca una funzione che, dato un array di interi monodimensionale, ne stampi a video il contenuto.

Esercizi 2b (a casa): altre utility

Si definiscano funzioni che:

- data una matrice di interi, ne stampi a video il contenuto
- data una variabile di tipo `data_t`, stampi a video la data contenuta con la formattazione `dd/mm/yyyy`.
 - Variante: usare la formattazione `dd M yyyy` (dove M indica il mese scritto a parole)
(Suggerimento: definire una funzione ausiliaria che, dato un mese come valore intero, ritorni la corrispondente stringa)
- data una variabile di tipo `studente_t`, stampi a video le informazioni contenute in maniera formattata (intestazione, poi lista di campo: valore)

Esercizio 3: ricerca minimo

Si definisca una funzione che trova il valore minimo in un array di interi.

Recap: ordinamento vettore (Bubble Sort)

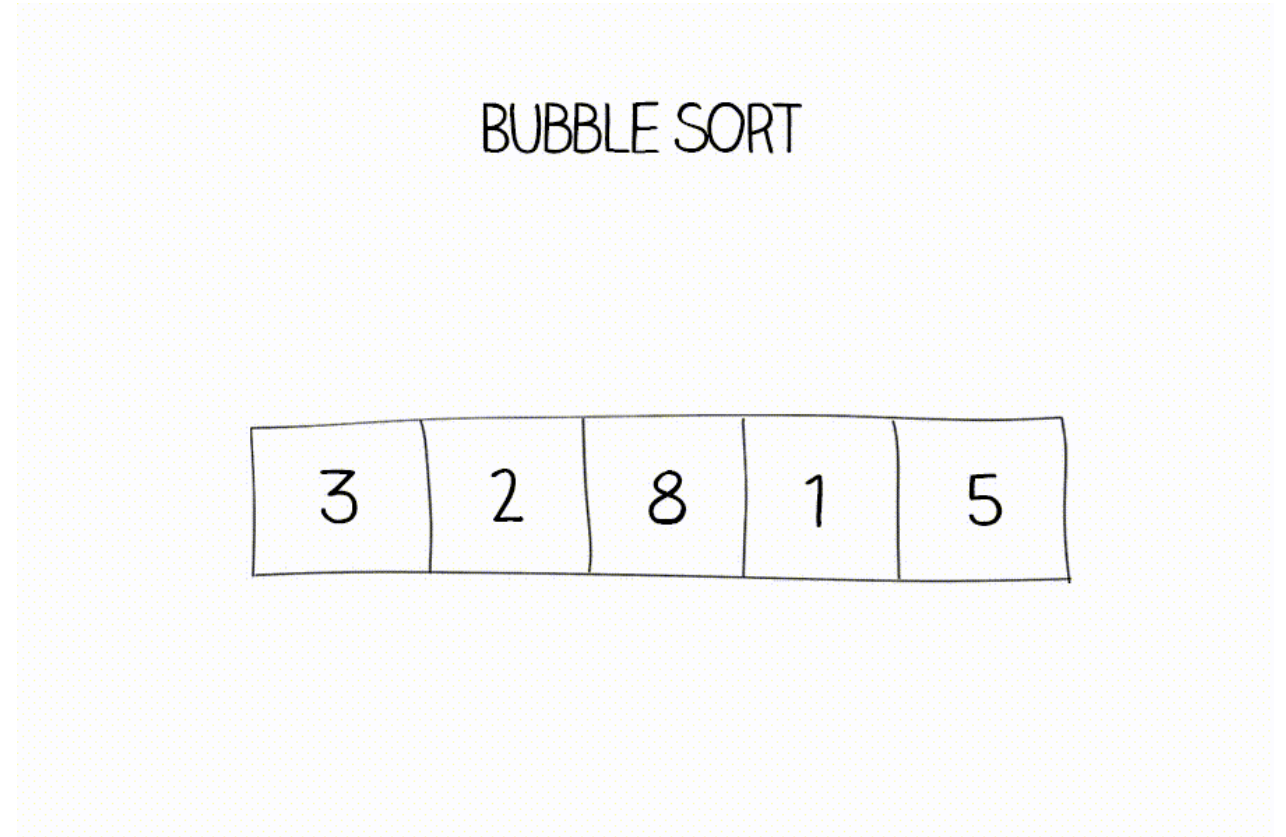
Si scriva un programma in linguaggio C che ordini gli elementi di un vettore.

Si utilizzi l'algoritmo Bubble Sort.

Importante: terminare il Bubble Sort anticipatamente appena ci si accorge che il vettore è ordinato.

Visualizzazione:

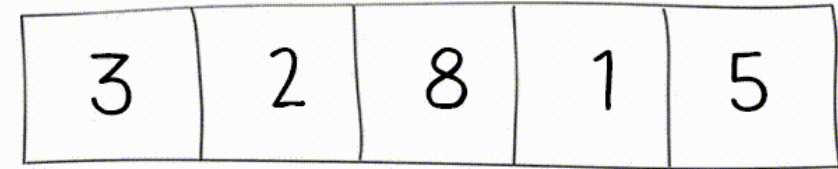
<https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/visualize/>



Esercizio 4: ordinamento vettore (Selection Sort)

Si definisca una funzione che ordini gli elementi di un vettore utilizzando l'algoritmo Selection Sort.

SELECTION SORT



Visualizzazione:

<https://www.hackerearth.com/practice/algorithms/sorting/selection-sort/visualize/>

Esercizio 5 (a casa): fusione (merge) di array ordinati

Si definisca una funzione che, dati due array di interi già ordinati, generi l'array *unione* dei due array, mantenendo l'ordinamento.

Variante: l'array risultante non contiene duplicati.

Esercizio 6 (a casa): sostituzione stringa

Si definisca una funzione che riceve in input le stringhe `source`, `find`, `replace` e sostituisca in `source` tutte le occorrenze di `find` con `replace`.

Suggerimento: definire un'altra funzione `cerca_sottstringa`, contenente l'algoritmo visto alle esercitazioni passate.