

Esercizi su

Riferimenti Array

- Tracce extra
 - Sul sito del corso

- Per casa:
 - *funz_moltiplica.cc*

Primi esercizi sugli array

- *ins_stampa_array.cc*
- Per casa
 - *array_casuali.cc*

Array e funzioni

- *raddoppia_valori.cc*
- Per casa
 - *calcola_somma.cc*
- *array_pari.cc*
 - Il numero di elementi significativi del secondo array è noto a tempo di scrittura del programma?

Domande 1/2

- Come fate quando cercate una parola sul vocabolario?
- E' necessario scorrere tutte le pagine?
- In quanti passi più o meno trovate una parola in un vocabolario di, per esempio, 1000 pagine?

Domande 2/2

- Sareste riusciti a trovare una parola con la stessa velocità se l'ordine con cui le parole sono disposte nel vocabolario fosse stato casuale?
- Qual è quindi la proprietà che permette di trovare una parola così velocemente?

Ordinamento 1/2

- Si possono effettuare operazioni di ricerca (e non solo) all'interno di un vettore di elementi in modo estremamente efficiente
 - Se gli elementi sono ordinati
- Ad esempio, si possono effettuare ricerche binarie
 - Proprio quello l'approccio che usiamo quando cerchiamo una parola sul vocabolario

Ordinamento 2/2

- Ora che abbiamo capito che l'ordinamento è una proprietà importante, vediamo un algoritmo per rimettere in ordine gli elementi di un vettore
- Imparerete molto di più su questo argomento nel corso di *Algoritmi e Strutture Dati*

Bubble sort 1/7

- In un vettore ordinato (in senso ascendente), l'elemento in testa al vettore è necessariamente quello di valore minimo
- Possibile primo passo ordinamento:
 - trovare l'elemento di valore minimo e metterlo in testa al vettore

Bubble sort 2/7

- Ad esempio, dato il seguente vettore
 - 2 5 1 3
- Il primo passo del *bubblesort* è:
 - 1 5 2 3

Bubble sort 3/7

- Consideriamo ora solo la *porzione di vettore* che va dal secondo elemento all'ultimo.
- Affinché il vettore originario sia ordinato, in testa a tale porzione è necessario che vi sia l'elemento di valore minimo tra tutti gli elementi della porzione stessa.

Bubble sort 4/7

- Ovviamente tale elemento non potrà essere maggiore di quello in testa al vettore.
- Passo successivo:
 - trovare l'elemento di valore minimo nella porzione e metterlo in testa a tale porzione, scambiandolo con quello precedentemente in testa alla porzione.

Bubble sort 5/7

- Dopo il primo passo si aveva:
 - 1 5 2 3
- Dopo il secondo passo:
 - 1 2 5 3
- I primi due elementi del vettore sono necessariamente in ordine corretto (e minori di tutti i successivi).

Bubble sort 6/7

- Algoritmo completo:
 - Spostare iterativamente in testa l'elemento minimo di porzioni successive del vettore, ciascuna ottenuta dalla precedente per sottrazione del primo elemento.

Bubble sort 7/7

- Dopo il secondo passo si aveva:

- 1 2 5 3

- Dopo aver scambiato gli ultimi due elementi:

- 1 2 3 5

Ordinamento

- *ord_array.cc*
- *copia_ord_array_main.cc*
 - Mantenimento ordinamento per costruzione

Compiti per casa

- In alcuni c'è il passaggio degli *array* alle funzioni

Prova di programmazione

- *contenitore_senza_struct.cc*
 - Tempo 2h30min
 - Si tratta di nuovo di un esempio di oggetto astratto