

Corso di Laurea in Fisica

Esame di Laboratorio II (I modulo)

10/09/2018

Istruzioni

Si risolva il seguente esercizio, scrivendo un programma in C++. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua le funzionalità richieste dal testo.

Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**.

Si richiede infine di mettere all'inizio del programma principale una riga di commento contenente il comando necessario per creare l'eseguibile.

Ordinamento di un insieme di dati

Si realizzi un programma che riordini un insieme di numeri dal più grande al più piccolo. In particolare il programma deve:

1. leggere dal file *unsortedData.txt* i numeri da riordinare;
2. riordinare i numeri;
3. scrivere i numeri ordinati in un nuovo file.

Il programma deve contenere le seguenti tre funzioni di cui vengono forniti i prototipi:

- `bool readData(std::string fileName, std::vector<double>& valVec)` che ha il compito di leggere i dati scritti in un file e salvarli in un vettore;
- `bool saveData(std::string fileName, std::vector<double>& valVec)` che ha il compito di salvare i dati di un vettore in un file;
- `void sortVector(std::vector<double>& valVec)` che ha il compito di riordinare i dati di un vettore dal più grande al più piccolo.

Il programma deve inoltre ricevere da riga di comando entrambi i nomi dei file, i.e. quello contenente i dati da leggere e quello che conterrà i dati ordinati.

La funzione `sortVector` deve scorrere il vettore varie volte, e ogni volta deve “portare” l’elemento più piccolo in fondo al vettore. Diciamo che il vettore possiede n elementi, alla prima passata del vettore la funzione porterà nella posizione $n - 1$ l’elemento in assoluto più piccolo. Alla seconda passata porterà il secondo elemento più piccolo nella posizione $n - 2$. Alla terza passata porterà il terzo elemento più piccolo nella posizione $n - 3$, e così via. La funzione deve quindi possedere:

1. un ciclo esterno con indice i da 1 a tutti gli elementi del vettore;
2. un ciclo interno con indice j da 0 al numero di elementi del vettore meno i ;
3. nel ciclo interno si deve implementare lo spostamento dell’elemento più piccolo.

Curiosità: l’algoritmo appena descritto è chiamato “bubble sort”, ed ha una complessità media, o equivalentemente un tempo di esecuzione medio, che scala come $O(n^2)$. Un algoritmo che ha complessità media inferiore, $O(n \log n)$, è per esempio il “quick sort”.