

POLITECNICO DI MILANO
Corso di Fondamenti di Informatica
Laurea On-Line
Prof. Pierluigi Della Vigna
Anno Accademico 2016/2017
4 settembre 2017
Quarta prova in presenza

È vietato consultare libri e appunti.

Tempo a disposizione: 3 ore.

Si prega di salvare tutti gli esercizi in un unico file con nome *Cognome.cpp*, dove *Cognome* indica il cognome del candidato.

Si raccomanda di salvare frequentemente il lavoro svolto.

All'inizio del vostro file apponete un commento del tipo

```
//Cognome:  
//Nome:  
//Matricola:  
//Classe Virtuale:
```

Array

Si realizzi, in linguaggio C++, una funzione che ha come parametri una matrice quadrata di dimensione N (con N dispari e maggiore di 3) di interi positivi ed un array di M interi positivi. Il numero M di celle dell'array corrisponde al numero di cornici concentriche presenti nella matrice ed è quindi pari a $N/2 + 1$. La funzione deve sommare i numeri presenti in ogni singola cornice concentrica ed assegnare il totale alla corrispondente cella dell'array: il totale dei valori della cornice più esterna andrà assegnato alla prima cella dell'array e così via.

Ad esempio se alla funzione viene passata la seguente matrice di dimensione N pari a 5 ed un array di M elementi (quindi con M pari a 3), si ottiene il seguente risultato.

0	1	0	1	0
0	0	1	0	1
2	0	1	1	3
0	1	0	0	0
1	0	2	0	0

11	3	1
----	---	---

Liste

In un grande magazzino che vende mobili e accessori per la casa, i clienti scelgono la merce da acquistare e si fanno fare da un commesso un preventivo con l'elenco di tutti gli articoli. Una volta pagato il preventivo in cassa, l'ordine verrà inviato al magazzino dove gli addetti selezioneranno gli articoli da consegnare al cliente in base all'elenco ricevuto. Ogni cliente pagante verrà accodato ai precedenti, in attesa che il proprio ordine venga evaso.

Si chiede di implementare una funzione C++ che gestisca la fase di preventivo. La funzione riceverà la lista dei preventivi e permetterà di accodarne uno nuovo. In particolare ogni preventivo sarà caratterizzato da un codice identificativo, nome, cognome, un campo che indichi l'avvenuto pagamento (che sarà contrassegnato al momento del pagamento) e ovviamente dalla lista degli articoli, anch'essi identificati da un codice.

Successivamente sviluppare un'altra funzione C++ che ricevuta in ingresso la lista precedentemente implementata, individui il primo preventivo pagato, secondo una logica FIFO, e aiuti i magazzinieri a gestire l'ordine di ciascun cliente.

Questa seconda funzione dovrà quindi individuare il cliente da servire, stamperà a video sul terminale del magazziniere l'elenco degli articoli richiesti che saranno identificati dal codice dell'articolo. Ogni volta che verrà preso un prodotto dallo scaffale il magazziniere lo spunterà dall'elenco con un lettore di codice a barre e la funzione aggiornerà la lista degli articoli ancora da reperire rivisualizzando l'elenco fino a che lo stesso non sarà esaurito.

Classi

In Formula 1, nei due giorni precedenti la gara, si svolgono le sessioni di prove libere, per mettere a punto le macchine, e le sessioni di prove di qualificazione per determinare l'ordine di disposizione dei piloti sulla griglia di partenza. Nell'intervallo di tempo in cui è aperta una sessione ogni pilota può eseguire quanti giri di pista vuole, ognuno dei quali viene cronometrato; man mano viene anche stilata la classifica delle migliori prestazioni realizzate da ogni pilota nella sessione in corso.

Vi si chiede di realizzare, in linguaggio C++, le classi GiroDiPista e Sessione per gestire appunto tali sessioni di prove.

La classe GiroDiPista permette di gestire i singoli giri di pista compiuti da ogni pilota ed è quindi caratterizzata da cognome e nome del pilota, dal suo numero di gara, dalla macchina su cui corre e dal tempo impiegato per eseguire il giro (tempo espresso in minuti, secondi e millesimi di secondo).

Si chiede di implementare almeno i seguenti metodi:

- il costruttore a cui vengono passati tutti i vari elementi caratterizzanti un GiroDiPista;
- un metodo per confrontare due oggetti di tale classe e verificare se sono o no relativi ad uno stesso pilota;
- un metodo per verificare se un oggetto presenta un valore di tempo impiegato a percorrere il giro minore di un altro oggetto.

La classe Sessione permette di gestire la singola sessione di prove ed è caratterizzata dal nome del circuito, dal giorno in cui si esegue la prova, dal tipo di sessione di prove (libere o di qualificazione Q1, Q2 e Q3) e dalla classifica delle prestazioni realizzate dai piloti. Si chiede di implementare almeno i seguenti metodi:

- il costruttore a cui vengono passati nome del circuito, data e tipo di sessione;
- un metodo per aggiungere un nuovo giro di pista alla sessione aggiornandone la classifica (tenendo conto che ogni pilota compare con la sua prestazione migliore);
- un metodo per stampare tutte le informazioni relative ad una sessione di prove.