POLITECNICO DI MILANO
Corso di Fondamenti di Informatica
Laurea On-Line
Prof. Pierluigi Della Vigna
Anno Accademico 2016/2017
10 luglio 2017
Terza 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

"2048 è un videogioco libero, a giocatore singolo, pubblicato on-line il 9 marzo 2014 da Gabriele Cirulli. Si gioca su una semplice griglia di formato 4x4 in cui scorrono caselle con numeri diversi (tutti i numeri sono potenze di 2), senza intralci quando un giocatore le muove. Il gioco usa i tasti freccia della tastiera per spostare tutte le caselle a sinistra o a destra oppure in alto o in basso. Se due caselle contenenti lo stesso numero si scontrano mentre si muovono, si fondono in un'unica casella che avrà come numero la somma delle due tessere che si sono scontrate. Ad ogni turno, una nuova tessera con il valore di 2 o 4 apparirà in modo casuale in un punto vuoto sul tabellone. La partita è vinta quando, continuando a far combinare le tessere, si riesce a crearne una con il numero 2048. Se il giocatore non può più muovere le caselle (perché non ci sono più spazi vuoti o senza tessere adiacenti con lo stesso valore), la partita finisce" (da Wikipedia).
Vi si chiede di realizzare, in linguaggio C++, la funzione spostaCelle che avendo come parametri una matrice di dimensione 4x4 ed una direzione di spostamento, modifichi la matrice mettendo in atto lo spostamento richiesto.
Definiamo come "celle vuote" le celle contenenti 0 e come "celle piene" quelle contenenti un numero. Si tenga presente che bisogna effettuare lo spostamento di tutte le celle piene, nella direzione passata alla funzione, secondo le seguenti regole:

- Se una cella piena ha come adiacente nella direzione di spostamento il bordo della griglia o una cella piena di valore diverso, non si sposta.
- Una cella continua a spostarsi nella direzione di spostamento fintanto che potrà attraversare celle vuote. Non appena nel suo moto la cella dovesse incontrare il bordo della matrice o una cella di valore diverso, questo spostamento termina e la cella di partenza assumerà valore 0 e la cella di destinazione assumerà il valore originale della cella di partenza.
- Se una cella piena incontra nella direzione di spostamento una cella piena di ugual valore, le due si fondono e la cella risultante continua nello spostamento seguendo le regole indicate sopra.

Ad es. data la prima matrice e la direzione "sinistra", si ottiene la seconda matrice, partendo dalla seconda con direzione "basso" si ottiene la terza.

2		2	
	8		
2	2	2	2
	4	8	2

4			
8			
4	4		
4	8	2	

4			
8	4		
8	8	2	

Liste

Per poter effettuare visite specialistiche presso un ospedale sono necessarie due differenti fasi: *cassa*, ovvero presentarsi agli sportelli per prenotare la visita e pagare il ticket dovuto; *accodamento*, ovvero presentarsi sempre agli sportelli il giorno della visita per essere effettivamente inseriti nell'elenco ordinato del medico/ambulatorio che chiamerà il paziente per la visita, tenendo conto dell'orario assegnato in fase di prenotazione.
Per consentire tale processo quando il paziente si presenta nella struttura acquisisce, tramite l'apposito distributore, il biglietto di prenotazione per la chiamata agli sportelli scegliendo se prelevare il biglietto per operazioni di *cassa* o per operazioni di *accodamento*.
Il sistema informatico, che gestisce tali operazioni, creerà una lista unica degli utenti che hanno prelevato il ticket (di tipo C per operazioni di cassa e di tipo A per operazioni di accodamento) per la chiamata agli sportelli, organizzando poi le chiamate secondo semplici regole: si effettueranno le chiamate attingendo dalla lista in modalità FIFO (in particolare la lista è costruita in modo tale che i nuovi elementi saranno posti in coda ad essa), tenendo però conto che si vuole comunque dare precedenza alle operazioni di accodamento. Non sarà una precedenza assoluta, ma una chiamata di *cassa* si farà solo se non ci sono *accodamenti* in attesa o se sono stati eseguiti tre *accodamenti* consecutivi.
Una funzione principale avrà il compito di gestire le richieste di *cassa* o di *accodamento* avvalendosi di opportune altre funzioni.
Scrivere, in linguaggio C++, la funzione che, ricevuta in ingresso la lista delle prenotazioni per la chiamata allo sportello ed il numero di accodamenti consecutivi effettuati, restituisca l'identificativo di chi si deve presentare allo sportello, rispettando i vincoli sopra indicati, e il numero di accodamenti consecutivi aggiornato.
Scrivere una seconda funzione che, chiamata dalla funzione principale, permetta all'operatore di sportello di inserire i dati necessari (anagrafe semplificata, id ambulatorio, ora prenotazione) ed accodare il paziente nella lista dello specifico medico/ambulatorio mantenendo la lista ordinata secondo l'orario di prenotazione. Per semplicità si ipotizzi che vi siano solo tre medici/ambulatori attivi.
Non è richiesta l'implementazione della funzione che gestisce la fase di *cassa*.

Classi

Si scrivano le classi Istogramma ed Osservazione per realizzare la gestione e la stampa di una serie di dati. Un istogramma contiene una serie di osservazioni. Ogni osservazione ha un valore di ascissa e un valore di ordinata e supponiamo che siano entrambi interi. Ad ogni valore di ascissa corrisponde una sola osservazione. Le classi devono offrire i servizi seguenti:

- Creazione di un'osservazione dai parametri interi ascissa ed ordinata passati.
- Creazione di un istogramma vuoto.
- Creazione di un istogramma da un array di osservazioni.
- Aggiunta di un'osservazione mediante operatore += (rispettando le semantiche standard degli operatori) tra un oggetto di classe Istogramma e uno di classe Osservazione. Se l'istogramma contiene già un'osservazione con la stessa ascissa, l'operazione non modifica l'istogramma.
- Cancellazione di un'osservazione mediante operatore -= tra un oggetto di classe Istogramma e uno di classe Osservazione. La cancellazione va a buon fine solo se esiste un'osservazione con stesso valore di ascissa e ordinata. Altrimenti l'istogramma resta immutato.
- Cancellazione di un'osservazione mediante operatore -= tra un oggetto di classe Istogramma e un valore di ascissa. La cancellazione viene effettuata solo se esiste un'osservazione con stesso valore di ascissa. Altrimenti l'istogramma resta immutato.
- Stampa delle osservazioni, ossia stampa delle coppie di dati ascissa ed ordinata.
- Stampa dell'istogramma mediamente operatore << (rispettando le semantiche standard degli operatori) su N righe e M colonne, secondo una scala opportuna dei valori di ascisse e ordinate.