

**POLITECNICO DI MILANO**  
**Corso di Fondamenti di Informatica**  
**Laurea On-Line**  
**Prof. Pierluigi Della Vigna**  
**Anno Accademico 2018/2019**  
**9 gennaio 2019**  
**Prima prova in presenza**

**È vietato consultare libri e appunti.**

**Tempo a disposizione: 2,5 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 scriva, in linguaggio C++, la funzione *individuaSequenza* che, avendo come parametri una matrice di interi ad una sola cifra (0-9) di dimensione MxN ed un numero intero, restituisca come valore di ritorno il numero di volte in cui ha trovato le cifre del numero passato come parametro, prese nell'ordine destrorso o sinistrorso lungo la sola direzione orizzontale; la funzione deve inoltre stampare a video e su un file appositamente creato, le coordinate della cella a partire dalla quale è stata individuata la sequenza ed il suo verso. Ogni sequenza individuata non dovrà necessariamente appartenere alla stessa riga.

Ad es. passati come parametri l'intero 4213 e la seguente matrice

7	3	5	4	2	1	3	8
9	6	5	3	4	4	2	1
3	7	8	1	5	6	0	2
2	7	6	2	4	3	1	9
0	5	1	4	5	7	9	8
6	3	1	2	4	7	7	0

si ottiene 3 come valore di ritorno e viene stampato:

*Trovate le sequenze a partire da:*

*(0 - 3) destrorso*

*(1 - 5) destrorso*

*(5 - 4) sinistrorso*

**Memento:** se a e b sono interi,  $a / b$  è un intero con troncamento del resto mentre  $a \% b$  fornisce il resto della divisione intera.

## Liste

Per una stazione di servizio per il rifornimento self-service di carburante, si vogliono implementare alcuni moduli del software C++ per la gestione delle pompe e della cassa. In particolare vi si chiede di sviluppare la funzione `attivaPompa()` che, ricevuti come parametri la lista dei rifornimenti in attesa di pagamento e la pompa che richiede l'abilitazione al rifornimento, restituisca tale abilitazione solo se nella lista di attesa è presente al più un rifornimento pendente per il pagamento relativamente alla stessa pompa. La funzione `rifornimento()`, ricevuti come parametri la lista dei rifornimenti già effettuati e non ancora pagati, il numero di pompa, il numero di litri erogati ed il prezzo unitario del carburante, dopo aver ricevuto il consenso dalla funzione `attivaPompa()`, dovrà erogare il carburante registrando in coda alla lista stessa il rifornimento, indicandone litri, numero di pompa ed importo. Infine vi si chiede di sviluppare la funzione `cassa()` che, ricevuti la lista dei rifornimenti erogati e non ancora pagati ed il numero di pompa, individui nella lista il primo rifornimento da regolarizzare presente per quella pompa, eliminandolo dalla lista stessa e stampando a video numero di pompa ed importo.