# Esercizi parte A

**Esercizio 1:**

Per la realizzazione del programma non utilizzare array o matrici. Il programma dovrà essere di supporto alla cucina di un grande ristorante. In particolare si è interessati ad accedere elettronicamente alla disponibilità dei prodotti presenti nella dispensa. Inoltre il cuoco deve poter accedere all'elenco delle ricette di tutti i piatti offerti, per sapere se si dispone di tutti gli ingredienti necessari. Un file dispensa.txt contiene l'elenco di tutte le confezioni dei prodotti presenti nella dispensa, nel seguente formato:

1A3F;Pomodori;23

il primo campo rappresenta il codice univoco del prodotto (sempre quattro cifre alfa-numeriche); il secondo campo è la descrizione; il terzo campo è la quantità disponibile in termini di unità di prodotto. Per uno stesso prodotto possono essere presenti più righe, che differiscono per quantità. In un altro file ricette.txt sono elencate le ricette con i rispettivi ingredienti, si supponga che ciascuna ricetta sia costituita da due ingredienti, nel seguente formato:

4;Salsiccia e polenta;4J78;2;4DF4;2

il primo campo numerico rappresenta il numero del piatto (come presente sul menu per le ordinazioni); il secondo è la descrizione; il terzo campo è il codice del primo ingrediente; il quarto campo è la quantità necessaria del primo ingrediente, il quinto campo è il codice del secondo ingrediente; il sesto campo è la quantità necessaria del secondo ingrediente. Le operazioni da svolgere sono:

1. Leggere il file dei prodotti dispensa.txt e memorizzare le informazioni in una lista singolarmente concatenata. In tale struttura ogni prodotto dovrà comparire una sola volta, e nel caso comparisse più volte nel file, andranno sommate le rispettive quantità. Stampare a video la lista.

2. Leggere il file delle ricette ricette.txt e memorizzare le informazioni in una lista singolarmente concatenata ordinandole per numero del piatto crescente. Stampare a video la lista.

3. Dopo aver richiesto al cuoco il numero di un piatto controllare se sono disponibili tutti gli ingredienti necessari e comunicarlo al cuoco. Se entrambe le quantità degli ingredienti sono sufficienti il cuoco decide di cucinare tale piatto. Decrementare le relative quantità di prodotti presenti in dispensa.

**Esercizio 2:**

Per la realizzazione del programma non utilizzare array o matrici. Si realizzi un programma che legga una espressione matematica in notazione infissa e la trasformi in notazione polacca inversa. A tal fine si ipotizzi che l'espressione in ingresso possa contenere solo numeri interi (ad 1 cifra), parentesi tonde aperte o chiuse e gli operatori +, -, *, /.
Un algoritmo semplificato per la trasformazione da notazione infissa a notazione polacca inversa è realizzabile mediante l'uso combinato di uno stack (LIFO) e di una coda (FIFO).
L'espressione in ingresso viene valutata da sinistra a destra e:
1) se si incontra un numero questo viene messo nella coda
2) se si incontra un operatore o una parentesi aperta il carattere viene messo nello stack

3) se si incontra una parentesi chiusa si tolgono gli elementi dallo stack uno ad uno e li si inseriscono nella coda fino a che non si incontra la corrispondente parentesi aperta che viene ignorata
4) Al termine gli elementi eventualmente ancora presenti nello stack vengono prelevati e inseriti nella coda. La coda viene poi svuotata e stampata.

**Esercizio 3:**

Si consideri la sequenza numerica così ottenuta a partire da un numero $n_0$ iniziale:

$$n_{i+1} = \begin{cases} n_i \times 3 + 1 & per\ n_i\ dispari \\ n_i / 2 & per\ n_i\ pari \end{cases} \quad (1)$$

la sequenza termina quando $n_{i+1}$ raggiunge il valore 1.
Scrivere un programma che calcoli e memorizzi tutte le sequenze possibili per $n_0$ nell'intervallo [1,100] utilizzando un array di liste concatenate. In particolare:
1. Si crei e si inizializzi l'array di liste
2. Per $n_0$ fatto variare tra 1 e 100 con passo 1 si calcolino tutti i numeri della sequenza ottenibile Mediante la formula (1) e li si memorizzino nella lista lista[$n_0$-1]
3. si indichi la sequenza piú lunga ottenuta stampandone la lunghezza e il valore n0 che la genera
4. si salvi in un file di testo la sequenza identificata al punto precedente nell'ordine con cui è stata ottenuta
5. ordinare ciascuna lista ottenuta in ordine crescente utilizzando un algoritmo di ordinamento a piacere

**Esercizio 4:**

Un grande magazzino intende gestire in modo automatizzato la redazione di alcuni report riguardo ai costi dei suoi reparti. Questa operazione viene eseguita mediante la registrazione su file di ogni spesa effettuata dal grande magazzino. Spetta all'IT (Information Technology) del grande magazzino il compito di sviluppare il programma che elabori le informazioni su questi files.
Il programma riceve in input un file spese.dat che contiene per riga: mese iniziale spesa, mese finale spesa, ammontare della spesa. Ad esempio
1 3 90
2 6 50
1 1 20
5 7 30
Il programma deve:
1.leggere il contenuto di tale file e memorizzarlo in una lista singolarmente concatenata, non ordinata
2.partendo dalla prima lista e mediante un attraversamento, generare una seconda lista che, nodo per nodo, contenga la spesa complessiva di un singolo mese. A tal fine si ipotizzi che la spesa sia uniformemente suddivisa nei mesi di competenza. Ad esempio la prima riga del file di esempio contribuirà per 30 alla spesa dei mesi 1, 2, 3. Tale passo di programma deve essere svolto nel seguente modo:
i.si attraversi la prima lista e per ogni nodo
ii.calcolare il contributo per ciascun mese come contributo=ammontare/(mese finale – mese iniziale +1)
iii.si realizzi un ciclo da mese iniziale e mese finale
iv.per ogni passo del ciclo si inserisca mese corrente e contributo nella seconda lista se non esiste il nodo relativo oppure si aggiorni l'ammontare del nodo relativo se già esistente.

3.ordinare la seconda lista in base all'ammontare

Esempio, stampa del punto 3:
1 50
2 40
3 40
5 20
6 20
4 10
7 10

**Esercizio 5:**

Si realizzi un programma per la gestione dei tempi per gare di nuoto. Ciascun partecipante viene registrato con le seguenti informazioni: nome, tempo effettuato in gara. Il file
"50rn.txt" contiene tali informazioni secondo il seguente formato:
NOME;MINUTI SECONDI CENTESIMI_DI_SECONDO
Es:
Rossi Paolo;0 31 99
Il programma dovrà:
1. creare una classe Item per una lista singolarmente concatenata nella quale l'informazione utile per ciascun nodo sarà costituita dai seguenti campi:
    - nome (variabile string)
    - minuti (variabile int)
    - secondi (variabile int)
    - tempo totale (variabile float) espresso in secondi in formato virgola mobile. Quest'ultimo può essere calcolato secondo la seguente espressione:

$$tempo\,totale = \frac{(minuti * 60 * 100) + (secondi * 100) + centesimi\,di\,secondi}{100}$$

2. leggere il file 50rn.txt e memorizzare le informazioni di ciascun partecipante in una lista ordinata in ordine crescente secondo il tempo totale in secondi;
3. stampare a video i primi 10 migliori tempi;
4. riportare le informazioni, visualizzate al punto 3, nel file "top_ten.txt" memorizzando le seguenti informazioni: nome, tempo totale in secondi. La formattazione del file è a discrezione del candidato;
5. Eseguire una "clear" della lista prima di terminare il programma.

**Esercizio 6:**

Si intende sviluppare un simulatore di una coda di stampa, in particolare si ipotizzi di avere una sola coda di stampa e che la stampante stampi 1 pagina al secondo. Ogni lavoro (job) mandato in stampa viene etichettato con il timestamp in secondi di inizio job di stampa, il numero di pagine ancora da stampare, l'ID dell'utente di chi lo ha mandato in stampa (da 1 a 10) e un numero sequenziale che identifica il lavoro in stampa (JOB, da 1 a crescere). La coda di stampa è gestita con modalità FIFO. Date queste informazioni si sviluppi il simulatore secondo le seguenti direttive:

1.creare la coda di stampa. Inizialmente vuota e inizializzare il numero sequenziale JOB a 1

2.Iniziare iterativamente un ciclo per contare i secondi ipotizzando una giornata lavorativa di 8 ore. Ad ogni iterazione (secondo):

a.generare un numero casuale per determinare se un lavoro è stato mandato in stampa. Tale condizione si verifica se tale numero casuale è divisibile per 180. Qualora questa condizione si verifichi generare, sempre casualmente, l'ID di chi ha mandato il lavoro in stampa (1-10) e il numero di pagine (1-20), inserire queste informazioni, il timestamp attuale e il numero di JOB nella coda di stampa e stampare opportuno messaggio a video con tutte le informazioni possibili.

b.aggiornare il numero di pagine ancora da stampare del lavoro in testa alla coda di stampa, se presente (suggerimento: utilizzare un contatore che viene inizializzato al numero di pagine da stampare del job in testa alla coda e, successivamente, viene fatto decrementare di una unità ad ogni iterazione di simulazione). Se tale numero diviene 0 rimuovere tale lavoro dalla coda e stampare a video opportuno messaggio di segnalazione fine stampa con tutte le informazioni possibili del job.

3.Stampare a video il numero totale di pagine stampate

4.(Esercizio aggiuntivo) Stampare a video indicazioni su quanti documenti e quante pagine sono state stampate per ciascun ID utente (per questo punto è ammissibile l'uso degli array).

**Esercizio 7:**

Il programma riceve in input il file "giro-italia.txt" in cui è riportato l'albo d'oro del Giro d'Italia di ciclismo. Il file contiene N righe che rappresentano per ogni anno il vincitore del giro. Ogni informazione nella riga è separata dal carattere ";" (punto e virgola), come illustrato dall'esempio sottostante:

1938 ; VALETTI Giovanni ; ita

1937 ; BARTALI Gino ; ita

Le informazioni sono rispettivamente:

ANNO ; COGNOME Nome ; nazione

Il programma da realizzare deve:

1. Memorizzare in una lista concatenata l'albo d'oro.

2. Creare una seconda lista concatena che memorizzi anche il numero di vittorie di ogni ciclista

3. Ordinare la seconda lista in ordine di vittorie decrescente per ciclista

Esempio:

5 BARTALI Gino

3 VALLETTI Giovanni

Ecc..

4. Aiutandosi con eventuali modifiche delle strutture dati precedenti, visualizzare il nome del ciclista/dei ciclisti con il massimo numero di anni trascorsi tra la prima e l’ultima vittoria.