

## Compito d'esame -- 14 febbraio 2022 -- Compito C

### Istruzioni (leggere attentamente)

**Nota importante:** la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione della prova d'esame.

**Registrazione dei dati dello studente:** PRIMA DI INIZIARE, eseguite (con un doppio click sull'icona) il programma `RegistraStudente` che si trova nella cartella `Esame`. Inserite (separatamente) *Numero di Matricola*, *Cognome* e *Nome* seguendo le istruzioni che compaiono sul terminale, e confermate i dati che avete inserito. Il programma genera il file `studente.txt` che contiene *Matricola*, *Cognome* e *Nome* su tre righe separate (nell'ordine indicato). Il file `studente.txt` non deve essere modificato manualmente. Verificate che i dati nel file `studente.txt` siano corretti. In caso di errore potete rieseguire il programma `RegistraStudente`.

**Tempo a disposizione:** 1 ora e 30 minuti

Per risolvere gli esercizi in modo che possano essere successivamente corretti è **necessario scrivere la soluzione di ogni esercizio nel file .py relativo**, che trovate nella cartella dell'esercitazione (ad esempio, per l'esercizio 1 scrivete il vostro programma nel file `Ex1.py`, per l'esercizio 2, nel file `Ex2.py`, e così via). Notate che ogni file incorpora del codice python per eseguire alcuni test sulla funzione. **NON** modificate questo codice, ma **SCRIVETE SOLO il contenuto della funzione**. Non spostate i file dalla loro posizione e non create nuovi file. **Si noti che per la correzione verranno usati insieme di dati di test diversi**.

In ogni esercizio, se non diversamente richiesto, potete sempre assumere che gli input forniti siano coerenti con la traccia (ad esempio, se l'esercizio chiede di dare in input alla funzione una lista non vuota di stringhe, potete sempre assumere l'input sia in tale forma e non è necessario nel codice effettuare controlli per gestire casi diversi da questo, considerando, ad esempio, il caso di lista vuota).

Per gli esercizi relativi a lettura da file, la stringa in input che identifica il file è sempre comprensiva anche della sua estensione e il file risiede sempre nella stessa directory dell'esercizio.

### Esercizi

- **Ex1(s)** Scrivere una funzione che riceve in ingresso una stringa `s` e calcola la lunghezza della più lunga sottostringa consecutiva di `s` in cui tutti i caratteri sono diversi tra di loro. Ad esempio, se la stringa `s` vale `'cavallo pazzo'`, la funzione deve restituire 6 poiché `'lo paz'` non contiene caratteri ripetuti e qualunque sequenza consecutiva più lunga contiene ripetizioni.
- **Ex2(m)** Scrivere una funzione che riceve in ingresso una matrice `m` rappresentata come lista di liste e contenente numeri interi. Chiamiamo *posizione massima-a-destra* per un elemento di `m` una posizione (coppia di coordinate (indice-riga, indice-colonna) ) in cui l'elemento è seguito nella stessa riga solo da elementi più piccoli, e che non sia l'ultima posizione nella riga. Ad esempio, sia `[7, 8, 7, 6]` la riga 2 di `m`, la posizione `(2, 1)` è una posizione massima-a-destra per l'elemento 8 e la posizione `(2, 2)` è una posizione massima-a-destra per l'elemento 7, mentre la posizione `(2, 0)` di 7 non è massima-a-destra perché questa occorrenza del 7 è seguita da un 8 e la posizione `(2, 3)` non è massima-a-destra per 6 perché è l'ultima posizione della riga. La funzione deve restituire un dizionario le cui chiavi sono gli interi di `m` per i quali esiste almeno una posizione massima-a-destra in `m`, e per ogni chiave `k` il valore associato

è un insieme contenente tutte le posizioni massime-a-destra di  $k$  in  $\mathbf{m}$ , dove la posizione è una tupla di dimensione due della forma (indice-riga,indice-colonna). Ad esempio, se  $\mathbf{m} = [[1, 10, 4, 5], [10, 7, 8, 9], [7, 8, 7, 6]]$ , la funzione deve restituire  $\{10: \{(0, 1), (1, 0)\}, 8: \{(2, 1)\}, 7: \{(2, 2)\}\}$  (notate che 8 compare in due posizioni, ma solo una è massima-a-destra, e che due delle tre posizioni in cui compare 7 non sono massime-a-destra).

- **Ex3(dispatch,file)** Scrivere una funzione che gestisce le richieste di prestito erogate da una banca in un periodo di 5 anni. Questa funzione prende in ingresso un numero intero **disp** che rappresenta la disponibilità iniziale della banca ed un file csv **file** che contiene (in ordine crescente di mese di richiesta) le informazioni sui prestiti richiesti dai clienti (e che la banca deve decidere se approvare) nel seguente formato:

MeseRichiesta, NomeCliente, Importo, NumRate, ImportoRata

dove MeseRichiesta è un numero intero  $> 0$  e  $< 60$  (5 anni) che denota il mese in cui è stata fatta la richiesta, NomeCliente identifica il richiedente (potete assumere che nessun cliente richieda più di un prestito), Importo è l'ammontare (in euro) che la banca (se approva la richiesta) deve versare al cliente all'inizio del mese della richiesta, NumRate è il numero di rate mensili (pagate all'inizio di ogni mese, a partire da quello successivo alla richiesta e che non supereranno mai il mese 60, cioè potete assumere che tutti i finanziamenti si concluderanno entro il mese 60) che il cliente deve rimborsare e ImportoRata è l'ammontare (in euro) che il cliente verserà alla banca (fino a completamento del numero di rate).

L'obiettivo è di calcolare, per ogni cliente, se la richiesta di prestito viene approvata. La banca approva le richieste nell'ordine in cui sono presenti nel file e se, all'inizio del mese della richiesta, ha abbastanza soldi disponibili, cioè la sua disponibilità è maggiore dell'importo richiesto dal cliente. La disponibilità all'inizio di un mese è data dalla disponibilità iniziale, meno i prestiti già approvati e più le rate già incassate dei prestiti approvati (*incluse le rate incassate nel mese corrente, cioè la banca prima incassa le rate e poi decide sui prestiti da erogare*). Ad esempio, se la disponibilità iniziale **disp** è 10000 ed il file **file** contiene:

```
1, Paola, 1000, 10, 105
1, Flavio, 3000, 20, 155
2, Gianna, 7000, 30, 250
10, Francesco, 7000, 30, 250
```

allora la funzione deve restituire `{'Paola': 'SI', 'Flavio': 'SI', 'Gianna': 'NO', 'Francesco': 'SI'}` poiché all'inizio del mese 1 la banca ha la disponibilità di 10000, eroga i due prestiti di 1000 e 3000 poi all'inizio del mese 2 incassa 105 come rata del primo prestito e 155 come rata del secondo. All'inizio del secondo mese, la disponibilità della banca è quindi 6260 ( $10000 - 1000 - 3000 + 105 + 155$ ) e quindi non può concedere il prestito a Gianna. All'inizio del mese 10, invece, la disponibilità della banca è salita a 7440 euro, grazie alle rate incassate nei mesi precedenti, e quindi la banca può concedere il prestito a Francesco.

**Nota:** Potete assumere che le ore e i costi nel file **file** siano sempre convertibili in numeri interi.