

# 14BHD INFORMATICA, A.A. 2021/2022

## Esercitazione di Laboratorio 9

---

### Obiettivi dell'esercitazione

Ripassare e consolidare:

- L'utilizzo dei costrutti condizionali per prendere decisioni all'interno di un programma
- L'utilizzo dei cicli per l'esecuzione ripetuta di istruzioni
- Definizione di liste e tabelle
- Manipolazione ed esecuzione di calcoli su liste e tabelle

### Contenuti tecnici

- Definizione di liste e operazioni sugli elementi
  - Definizione di tabelle e manipolazione dei suoi elementi
  - Enunciati while e for per la realizzazione di cicli
- 

### Da risolvere in laboratorio

- Esercizio 1. Scrivete **funzioni** che risolvano i problemi seguenti per liste di numeri interi, fornendo un programma di collaudo per ciascuna funzione.
- a. Scambiare tra loro il primo e l'ultimo elemento della lista.
  - b. Far scorrere tutti gli elementi di una posizione "verso destra", spostando l'ultimo elemento nella prima posizione. Ad esempio, la lista 1 4 9 16 25 deve diventare 25 1 4 9 16.
  - c. Sostituire con 0 tutti gli elementi di valore pari.
  - d. Sostituire ciascun elemento, tranne il primo e l'ultimo, con il più grande dei due elementi ad esso adiacenti. È possibile usare liste di appoggio.
  - e. Eliminare l'elemento centrale della lista se questa ha dimensione dispari, altrimenti eliminare i due elementi centrali.
  - f. Spostare tutti gli elementi pari all'inizio della lista (lasciando quelli dispari in coda), preservando però l'ordinamento relativo tra gli elementi.
  - g. Restituire il secondo valore maggiore della lista (senza pari meriti).
  - h. Restituire True se e solo se la lista è ordinata in senso crescente.
  - i. Restituire True se e solo se la lista contiene due elementi adiacenti duplicati.
  - j. Restituire True se e solo se la lista contiene elementi duplicati (non necessariamente adiacenti). [P6.4]

Esercizio 2. Lo schema dei posti a teatro è una tabella con i prezzi dei biglietti per ciascun posto, come questa. [P6.27]

10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
10	10	20	20	20	20	20	20	10	10
10	10	20	20	20	20	20	20	10	10
10	10	20	20	20	20	20	20	10	10
20	20	30	30	40	40	30	30	20	20
20	30	30	40	50	50	40	30	30	20
30	40	50	50	50	50	50	50	40	30

Scrivete un programma che gestisca un menù che chieda all'utente di scegliere un posto, un prezzo o l'uscita dal programma. Contrassegnate con un prezzo uguale a 0 i posti già venduti. Quando l'utente specifica un posto, accertatevi che sia libero e che le coordinate siano all'interno della tabella. Quando, invece, specifica un prezzo, assegnategli un posto qualsiasi tra quelli disponibili a quel prezzo.

Esercizio 3. Nei lunghi viaggi in auto, per ingannare il tempo, si può fare il gioco delle “parole concatenate”. Il primo giocatore dice una parola iniziale, poi a turno ciascun giocatore dovrà dire una nuova parola (ossia mai detta prima) la cui *sillaba iniziale* sia uguale alla *sillaba finale* della parola precedente. (NOTA: Per semplicità, ipotizziamo che tutte le sillabe siano lunghe esattamente 2 caratteri, quindi per “figli” la ‘sillaba’ finale sarà “li” e non “gli”).

Ad esempio: gatto - torino - notte - tela - lana ...

Scrivere un programma per permettere di gestire una o più partite del gioco.

Ciascuna partita termina quando un giocatore inserisce una parola già detta nella stessa partita, quando inserisce una parola non correttamente concatenata, oppure quando non riesce a proseguire (per abbandonare, inserisce \*).

---

### Da risolvere a casa

Esercizio 4. Un supermercato vuole ricompensare il proprio miglior cliente del giorno, mostrandone il nome su uno schermo all'interno del negozio. A questo scopo, vengono memorizzati in una lista (customers) i nomi di tutti i clienti del giorno e, in

un'altra lista (sales), il corrispondente importo della spesa effettuata.

Scrivete la funzione `nameOfBestCustomer(sales, customers)` che restituisca il nome del cliente che ha speso la cifra più alta.

Poi, scrivete un programma che chieda al cassiere di digitare tutti gli importi spesi e i nomi dei relativi clienti, aggiungendoli via via a due liste distinte, per poi invocare la funzione che avete progettato e visualizzare il risultato. Usate il prezzo 0 come sentinella. [P6.33]

Esercizio 5. Scrivere un programma che acquisisca dall'utente un elenco di numeri interi positivi, scritti su un'unica riga e separati dal carattere ':'.

Esempio: 3:12:21:8:4:7

Il programma dovrà stampare, sempre nello stesso formato:

- gli stessi numeri, ad eccezione del minimo e del massimo (es. 12:8:4:7)

- i soli numeri pari (es. 12:8:4)

- i soli numeri di 2 cifre (es. 12:21)

Si suggerisce di lavorare costruendo una lista di numeri interi.

Esercizio 6. Scrivere un programma che generi una "spirale" di numeri interi inserita in una matrice di  $N \times N$  elementi, con  $N$  inserito dall'utente. I numeri saranno i valori compresi tra 1 e  $N^2$ .

Suggerimento: costruire prima la spirale numerica in una tabella  $N \times N$ , e solo successivamente stamparla.

Ad esempio, se  $N=4$ , il programma dovrà stampare:

1	2	3	4	⊙	→	→	↘
12	13	14	5	↗	→	↘	↓
11	16	15	6	↑	⊗	↙	↓
10	9	8	7	↑	←	←	↙