



Esercitazione 4 (10-22)



4.1 Lista rovesciata

- Scrivere una funzione ricorsiva per rovesciare una lista
 - Parametro: `list`
 - Risultato: `list`
- L'output contiene gli stessi elementi dell'input, però in ordine opposto [2, 3, 1, 4] -> [4, 1, 3, 2]

Saranno accettate esclusivamente le soluzioni ricorsive

Se lunghezza < 2: la lista resta uguale

Altrimenti: la lista di output è ottenuta rovesciando la lista di input, ad esclusione del primo elemento (ricorsione); al risultato si concatena infine il primo elemento della lista di input



4.2 Tabelline in file

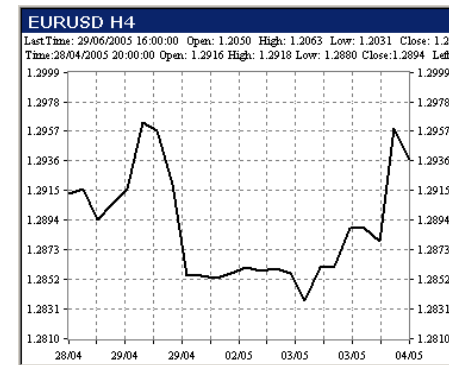
- Chiedere all'utente un numero N
- Stampare a video la tavola pitagorica, fino ad N
 - In ciascuna riga, valori separati da virgola
- Salvare la stessa tavola in un file

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100



4.3 Valori in file

- Si supponga, per esempio, che una misura venga ripetuta due volte al giorno, per più giorni
 - Le misure vengono riportate in un file
 - Una riga del file per ogni giornata
 - Quindi, due misure per ogni riga
- Leggere tutti i valori contenuti in un file di questo tipo
 - Ogni riga riporta due valori reali, separati da virgola
- Visualizzare massimo, minimo e media di tutti i valori letti



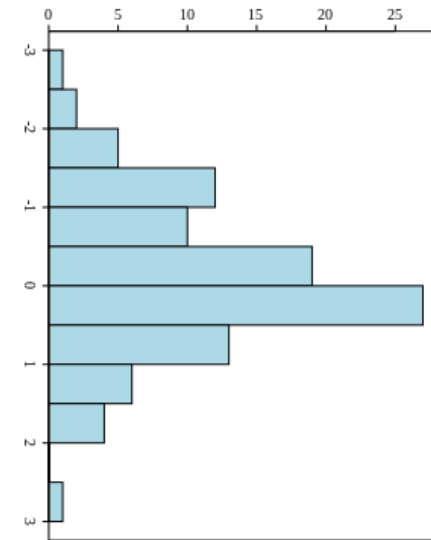
Usare l'operazione `split` su stringa

Se necessario, provare prima ad elaborare un file con un solo valore per riga



4.4 Istogramma con barre orizzontali

- Leggere da un file una sequenza di valori (positivi)
- Mostrare un istogramma, in un canvas 600x600
 - Larghezza di ciascuna barra proporzionale al valore corrispondente
 - Rosso di ciascuna barra proporzionale al valore corrispondente
 - Verde e blu invece sempre al massimo
 - La barra più lunga occupa tutto lo spazio disponibile
 - L'altezza del canvas è divisa equamente per le barre



Memorizzare i valori in una lista



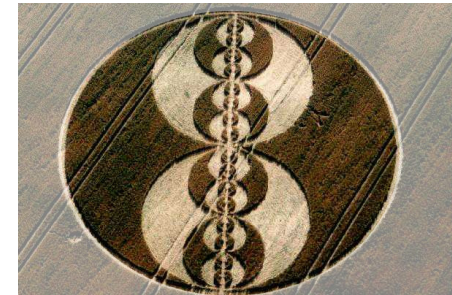
4.5 Lista di ambulanze

- Leggere da un file i dati di una lista di ambulanze (es. 3.2)
 - In ogni riga ci sono i dati completi di una ambulanza: nome e posizione
- Poi, in un ciclo, chiedere all'utente il punto in cui intervenire
- Il sistema sceglie l'ambulanza più rapida per l'intervento
- Comunica all'utente:
 - Il nome dell'ambulanza
 - La stima della distanza di Manhattan tra ambulanza e punto di intervento



4.6 Circular infinity

- In un riquadro assegnato, disegnare ricorsivamente dei cerchi, con i centri allineati in verticale
- Tracciare un cerchio contenuto nel riquadro (x , y , w , h)
 - $r = h / 2$, $xc = x + w / 2$, $yc = y + h / 2$
- Dividere il riquadro in due metà
- Applicare a ciascuno dei due riquadri il processo, ricorsivamente
- Ad ogni livello, cambiare o alternare il colore del cerchio disegnato
- Non disegnare cerchi più piccoli di un pixel



Saranno accettate esclusivamente le soluzioni ricorsive



4.7 Frogger, automobili

- Modificare la classe `Ball` dell'esempio `bounce`
- Il movimento è solo orizzontale
- Quando un oggetto esce da un bordo, riappare dal bordo opposto
- Dare però al movimento uno spazio più ampio dell'arena visibile
 - Ammesse coordinate negative o maggiori del bordo destro
 - In questo modo, l'oggetto riappare con un certo ritardo
- Inserire la classe nel gioco, come `Vehicle`
- Rimuovere la rana quando urta un veicolo

