

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 — Compito A del 7/2/2025

prof. Gianluca Amato

Esercizio 1A (8 punti)

Supponiamo di avere un file di testo le cui righe hanno il seguente formato:

- un carattere iniziale, che può essere solo `r` oppure `l`;
- due numeri interi che rappresentano le coordinate x ed y di un punto;
- altri due numeri interi che rappresentano le coordinate x ed y di un secondo punto.

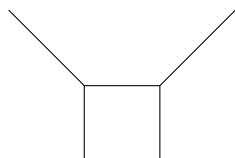
Il carattere iniziale e i numeri sono separati tra loro da caratteri spazio. Scrivere un programma che legge un file dal nome `disegno.txt` che rispetta il formato di cui sopra. Per ogni riga letta dal file, usando la libreria `ezgraphics`, il programma deve:

- se il carattere iniziale è `l`, disegnare una linea tra i due punti specificati;
- se il carattere iniziale è `r` disegnare un rettangolo con i due punti specificati come vertici opposti.

Ad esempio, se il file `disegno.txt` contiene:

```
r 100 100 200 200
l 0 0 100 100
l 200 100 300 0
```

il programma dovrà disegnare qualcosa di simile a questo:



Esercizio 2A (6 punti)

Scrivere una funzione `righe_monotone` che prende come parametro una tabella, anche frastagliata, e restituisce una lista di booleani (un booleano per ogni riga) che indica quali righe della tabella sono monotone. Una riga è monotona se tutti i suoi elementi sono in ordine strettamente crescente o decrescente.

Ad esempio, se la funzione è chiamata con la matrice

```
[[1, 2, 4, 7],
 [5, 5, 5, 5],
 [3, 2, -1],
 [1, 2, 3, 2]]
```

il risultato sarà `[True, False, True, False]` perché la prima e la terza riga sono monotone, ma la seconda e la quarta no.

Esercizio 3A (3 punti)

Utilizzare il framework `pytest` per collaudare la funzione `righe_monotone` dell'esercizio precedente. Si considerino in particolare i casi corrispondenti ai seguenti parametri in input:

- l'esempio fornito nel testo dell'esercizio;
- una tabella vuota (senza righe);
- una tabella che ha almeno due righe, una vuota e una no;
- una tabella che ha almeno due righe, tutte vuote.

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 – Compito B del 7/2/2025

prof. Gianluca Amato

Esercizio 1B (8 punti)

Supponiamo di avere un file di testo le cui righe hanno due possibili formati:

1. il carattere iniziale `c`, seguito da due numeri che rappresentano le coordinate x ed y del centro di un cerchio, seguiti da un terzo numero che rappresenta il raggio del cerchio; il carattere iniziale e i numeri sono separati tra di loro da spazi;
2. un carattere iniziale diverso da `c`, seguito da qualsiasi cosa.

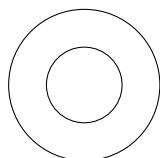
Scrivere un programma che legge un file dal nome `disegno.txt` che rispetta le specifiche di cui sopra. Per ogni riga letta dal file, usando la libreria `ezgraphics`, il programma deve:

- se il carattere iniziale è `c`, disegnare un cerchio di centro e raggio specificato;
- se il carattere iniziale è diverso da `c`, ignorare la riga.

Ad esempio, se il file `disegno.txt` contiene:

```
c 100 100 100
hello world
c 100 100 50
```

il programma dovrà disegnare qualcosa di simile a questo:



Esercizio 2B (6 punti)

Scrivere una funzione `righe_progressione` che prende in input una tabella (anche frastagliata) e restituisce una lista con i numeri delle righe che rappresentano una progressione aritmetica. Una riga è una progressione aritmetica se la differenza tra ogni elemento e il precedente è costante.

Ad esempio, se la funzione è chiamata con la matrice

```
[[1, 2, 3, 4],
 [5, 5, 5],
 [1, 2, 3, 10],
 [3, 1, -1, -3]]
```

il risultato è `[0, 1, 3]`. Infatti la riga numero 0 è una progressione aritmetica (la differenza tra un numero e il precedente è sempre 1), la riga numero 1 è una progressione aritmetica (la differenza tra un numero e il precedente è sempre 0), la riga numero 3 è una progressione aritmetica (la differenza tra un numero e il precedente è sempre -2). La riga 2, invece non è una progressione aritmetica.

Esercizio 3B (3 punti)

Utilizzare il framework `pytest` per collaudare la funzione `righe_progressione` dell'esercizio precedente. Si considerino in particolare i casi corrispondenti ai seguenti parametri in input:

- l'esempio fornito nel testo dell'esercizio;
- una tabella vuota (senza righe);
- una tabella che ha almeno due righe, una vuota e una no;
- una tabella che ha almeno due righe, tutte vuote.

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 — Compito C del 7/2/2025

prof. Gianluca Amato

Esercizio 1C (6 punti)

Scrivere un programma che utilizza la libreria **ezgraphics** per realizzare un semplice gioco. Il programma deve disegnare un quadrato rosso di dimensioni 20x20 pixel. Il quadrato deve inizialmente trovarsi in una posizione casuale all'interno della finestra grafica. Ogni volta che si preme il tasto freccia a destra (mentre è attiva la finestra grafica), il quadrato si sposta di 20 pixel verso destra; ogni volta che si preme il tasto freccia sinistra, il quadrato si sposta di 20 pixel verso sinistra. Analogamente per i tasti freccia in alto e freccia in basso.

Non appena il quadrato esce dal bordo della finestra grafica, il gioco si interrompe e compare un messaggio (sul terminale) con il numero di spostamenti effettuati prima di uscire dal bordo.

Suggerimento: per gestire gli eventi di pressione dei tasti, si può utilizzare il metodo `getKey` dell'oggetto `GraphicsWindow`. Tocca a voi scoprire come riconoscere i tasti freccia.

Esercizio 2C (6 punti)

Scrivere una funzione `genera_tabella` che prende come parametro una lista di coppie. Ogni coppia è formata da due numeri interi positivi o nulli. Ad esempio, un argomento valido per la funzione è la lista `[(2,2), (3,1), (0,4)]`. La funzione `genera_tabella` deve restituire una tabella secondo le indicazioni della lista `l` fornita come parametro. In particolare, una coppia del tipo `(n, m)` presente in `l` indica che la riga `n` della matrice deve essere lunga `m` e contenere solo zeri. Se per una riga `i` non è presente nessuna coppia `(i, j)`, allora la riga `i` sarà lunga 0. Il numero di righe della matrice risultato è il più piccolo numero possibile perché vengano rispettate le indicazioni della lista `l`.

Ad esempio, se la funzione è chiamata con l'argomento `[(2,2), (3,1), (0,4)]`, il risultato sarà la matrice

```
[
  [0, 0, 0, 0],
  [],
  [0, 0]
  [0]
]
```

Infatti la prima riga è lunga 4 perché è presente la coppia `(0,4)`. La seconda riga è lunga 0 perché non è presente nessuna coppia che inizia con 1. La terza riga è lunga 2 perché è presente la coppia `(2,2)`. La quarta riga è lunga 1 perché è presente la coppia `(3,1)`.

Esercizio 3C (3 punti)

Utilizzare il framework `pytest` per collaudare la funzione `genera_tabella` dell'esercizio precedente. Si considerino in particolare i casi corrispondenti ai seguenti parametri in input:

- l'esempio fornito nel testo dell'esercizio;
- una lista vuota;
- una lista che genera una tabella rettangolare (non frastagliata).

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 — Compito D del 7/2/2025

prof. Gianluca Amato

Esercizio 1D (6 punti)

Scrivere un programma che utilizza la libreria **ezgraphics** per realizzare un semplice gioco. Il programma deve disegnare un quadrato verde di dimensioni 20x20 pixel. Il quadrato deve inizialmente trovarsi al centro della finestra grafica. Ogni volta che si clicca in un punto qualunque della finestra, il quadrato si sposta di 20 pixel nella direzione del punto cliccato, in orizzontale, verticale o in entrambe le direzioni. Non importa quanto lontano sia il punto in cui si clicca rispetto alla posizione corrente del quadrato, conta solo la direzione, perché lo spostamento sarà sempre di 20 pixel in orizzontale e/o in verticale. Se si clicca sopra il quadrato, si esce dalla finestra grafica e il gioco si interrompe.

Attenzione. Dovete capire voi se, a seguito del click, il quadrato deve muoversi in orizzontale, verticale o in diagonale. Fate come volete, purché sia un comportamento ragionevole, e purché sia possibile effettivamente far muovere il quadrato in orizzontale e in verticale (non solo in diagonale) senza troppa difficoltà.

Esercizio 2D (6 punti)

Scrivere una funzione **genera_tabella** che prende come parametro una lista di coppie. Ogni coppia è formata da due numeri interi positivi o nulli. Ad esempio, un argomento valido per la funzione è la lista $[(2,2), (3,1), (0,4)]$. La funzione **genera_tabella** deve restituire una tabella rettangolare tutta piena di zeri. Fanno eccezione le celle di coordinate (i,j) se la coppia (i,j) è presente nel parametro passato alla funzione: queste celle devono essere riempite con 1. La dimensione della matrice deve essere quella minima necessaria perché vengano rispettate le indicazioni della lista 1

Ad esempio, se la funzione è chiamata con l'argomento $[(2,2), (3,1), (0,4)]$, il risultato sarà la matrice

```
[
  [0, 0, 0, 0, 1],
  [0, 0, 0, 0, 0],
  [0, 0, 1, 0, 0],
  [0, 1, 0, 0, 0]
]
```

Infatti, le celle poste ad 1 sono quelle in posizione (2,2) (riga 2 colonna 2, iniziando a contare da zero), (3,1) (riga 3 colonna 1) e (0,4) (riga 0 colonna 4).

Esercizio 3D (3 punti)

Utilizzare il framework `pytest` per collaudare la funzione `genera_tabella` dell'esercizio precedente. Si considerino in particolare i casi corrispondenti ai seguenti parametri in input:

- l'esempio fornito nel testo dell'esercizio;
- una lista vuota;
- una lista che genera una matrice con 2 righe e 2 colonne interamente riempita di 1.

Programmazione e Algoritmi 1 - Prova scritta

A.A. 2024/25 — Compito del 7/2/2025

prof. Gianluca Amato

Esercizio 4 (7 punti)

Eseguire passo-passo il seguente codice Python, utilizzando l'apposito modulo:

```
1  def f(l, n):
2      m = []
3      for i in range(n):
4          m.append(l)
5          l[i] = i
6      return m
7
8  x = ["a", "b", "c"]
9  y = f(x, 3)
10 print(y)
```

Disegnare inoltre un diagramma che mostri lo stato dell'heap e dello stack al termine dell'esecuzione del codice.

Esercizio 5 (5 punti)

Scrivere la funzione `swap_pair(s)` che prende come parametro una stringa `s` e restituisce la stringa ottenuta scambiando il primo e il secondo carattere di `s`, il terzo e il quarto, e così via. Se la lunghezza di `s` è dispari, l'ultimo carattere non viene scambiato. Ad esempio,

- `swap_pair("ciao")` restituisce `"icoa"`;
- `swap_pair("mario")` restituisce `"amiro"`;
- `swap_pair("")` restituisce `""`.

Esercizio 6 (5 punti)

Si descriva in dettaglio il funzionamento degli algoritmi di ricerca lineare e di ricerca binaria. Si discutano le differenze tra i due algoritmi, e si indichi in quali casi è preferibile utilizzare uno piuttosto che l'altro.

Infine, si illustri il funzionamento dei due algoritmi sull'esempio specifico in cui si intende cercare il valore 9 all'interno della lista `[1, 3, 5, 7, 9, 11, 12, 13, 15, 17, 19]`.

Programmazione e Algoritmi 1 - Prova pratica aggiuntionale

A.A. 2024/25 — Compito del 7/2/2025

prof. Gianluca Amato

Eseguire passo-passo il seguente codice Python, utilizzando l'apposito modulo:

```
1 def fun(l, n, r):
2     if n == len(l):
3         return r
4     elif n % 2 == 0:
5         return fun(l, n+1, l[n] + r)
6     else:
7         return fun(l, n+1, r)
8
9 x = fun("ciao", 0, "")
10 print(x)
```