Esercitazione 2
Programmazione Avanzata ed Elementi di Ingegneria del Software

Esercizio 1:

Scrivere due file, un file `calcolo.c` e un file `calcolo.h`, che contengano:

- `calcolo.c`:
  - una funzione `check_triangle(punto x, punto y, punto z)` che restituisca `True` se i 3 punti in ingresso formano un triangolo e `False` in caso contrario.[1]
  - una funzione `get_perimeter(punto x, punto y, punto z)` che ritorni il valore del perimetro del triangolo associato ai 3 punti in ingresso o, se non ne esiste nessuno (cioè `check_triangle` ha reso `False`), ritorna 0.[2]
  - l'inizializzazione di una costante `pi = 3.14159`, che deve essere poi resa disponibile al `main`.
- `calcolo.h`:
  - la firma delle funzioni da rendere disponibili al `main`.
  - la dichiarazione della costante `pi`.
  - la dichiarazione e la definizione della struttura `punto`.

Il dato `punto` è una struttura dati definita come
`typedef struct { float x; float y } punto;`
e rappresenta un punto bidimensionale.

A scopo di prova:

I punti x = {0, 0}, y = {1, 1} e z = {2, 2} NON formano nessun triangolo.
I punti x = {0, 0}, y = {2, 0} e z = {1, 1} formano un triangolo il cui perimetro è (circa) uguale a 4.83.

---

1 Tre punti formano un triangolo se rispettano la diseguaglianza triangolare, di cui a: `https://it.wikipedia.org/wiki/Disuguaglianza_triangolare`

2 La distanza tra due punti (e quindi la lunghezza del lato associato a quei due punti) si calcola con la formula classica della distanza fra due punti di cui a: `https://www.youmath.it/formulari/formulari-di-geometria-analitica/426-distanza-tra-due-punti-nel-piano.html`

Esercizio 2:
Reimplementare il codice dell'esercizio 1 ma avvalendosi dell'ADT design pattern.

Esercizio 3:
Implementare uno stack[3] di numeri interi avvalendosi dell'ADT design pattern. Scrivere 2 file, `stack.c` e `stack.h`, che contengano:

- `stack.c`:
  - l'implementazione dello stack mediante l'uso delle liste concatenate, cioè le funzioni per creare lo stack, inserirvi un elemento oppure eliminarlo.
- `stack.h`:
  - le funzioni che devono essere rese disponibili al main

Come prova di correttezza rispetto al requisito dell'uso dell'ADT: implementare una seconda versione di `stack.c`, che contenga l'implementazione dello stesso stack ma mediante l'uso di array, e verificare che i file `main.c` e `stack.h` non cambiano.

3 Uno stack è una struttura dati, anche chiamata pila, che funziona come a: `https://it.wikipedia.org/wiki/Pila_(informatica)`