**Esercizio Ricorsione**

Realizzare un'applicazione ***DisuguaglianzaTriangolare*** definita come segue.

L’applicazione contiene una funzione **ricorsiva** ***disuguaglianzaTriangolare*** cheverifica, all’interno di un array di interi ricevuto come parametro, se esiste una terna di interi consecutivi nella sequenza che soddisfa la disuguaglianza triangolare (ovvero ciascuno dei tre interi è strettamente minore della somma degli altri due). Ad esempio, se l’array è **[2, 4, 10, 7, 1]**, la funzione deve restituire **1**, in quanto la tripla **[4, 10, 7]** è tale che **4<10+7**, **10<4+7** e **7<4+10**. Se l’array è **[1, 3, 1]**, la funzione deve restituire **0**, poiché nell’unica tripla **[1, 3, 1]** non è vero che **3<1+1**. Notare che non si richiede che i tre interi della tripla cercata siano positivi.

La funzione ***disuguaglianzaTriangolare*** ha due parametri: l’array e la sua lunghezza. Nel caso in cui si desideri utilizzare un terzo parametro per realizzare la ricorsione, deve essere definita un’ulteriore funzione ***disuguaglianzaTriangolareRic*** che ha tre parametri e che realizza la ricorsione. In tal caso la funzione ***disuguaglianzaTriangolare*** invoca la funzione ***disuguaglianzaTriangolareRic*** (fornendole opportuni parametri) per calcolare il risultato.

L’applicazione contiene una funzione ***scriviNumeri*** che scrive i valori di un array di interi su di un file. La funzione riceve come parametro l’array, la sua lunghezza ed un ulteriore intero di controllo. Il file si deve chiamare interi.dat (nel caso sia binario) oppure interi.txt (nel caso in cui sia testuale). La scelta se utilizzare un file binario oppure un file di testo è lasciata allo studente.

L’applicazione contiene una funzione ***quantiNumeri*** che determina quanti numeri interi sono memorizzati su un file dal nome interi.dat (nel caso in cui si utilizzi un file binario) oppure interi.txt (nel caso in cui si utilizzi un file testuale).

L’applicazione contiene una funzione ***leggiNumeri*** che legge una sequenza di interi da un file dal nome interi.dat (nel caso in cui si utilizzi un file binario) oppure interi.txt (nel caso in cui si utilizzi un file testuale) e li memorizza su un array che è ricevuto come parametro

Una funzione main che è già stata implementata in un file ricorsione.c (ad eccezione dei messaggi finali per l’utente) il cui codice può essere scaricato e copiato da moodle (il file contiene anche le intestazioni delle altre funzioni). La funzione chiede all’utente se vuole inserire una sequenza di interi, oppure leggere la sequenza e l’intero di controllo che sono memorizzati sul file. Dopo aver memorizzato tale sequenza in un array, viene stampato un messaggio che indica se esiste una terna di interi consecutivi nella sequenza che soddisfa la disuguaglianza triangolare. La funzione termina salvando il contenuto dell’array su di un file.

La funzione main è già parzialmente implementata in un file **Ricorsione.c**.

**Esercizio Liste**

Realizzare un'applicazione ***Fritti*** per gestire il menù di una pizzeria.

Definire due strutture, una per rappresentare un elemento della lista ed una per rappresentare un fritto che fa parte del menù. In particolare, l’applicazione deve gestire ciascun fritto come una struttura con tre campi. Il primo campo è una stringa che indica il nome del fritto (ad esempio **“fiori di zucca”**), il secondo campo è un valore reale che indica il prezzo del fritto, ed il terzo campo è un carattere **‘s’** o **‘f’** che indica se il fritto è surgelato o fresco.

Definire una funzione che visualizza il menù, stampando per ciascuna fritto il nome, il prezzo e se il fritto è surgelato oppure no.

Definire una funzione che inserisce un fritto in coda al menù.

Definire una funzione che ordina la lista così che i fritti freschi compaiano prima di quelli surgelati e così che i fritti freschi (e quelli surgelati) siano fra loro ordinati per prezzo.

Definire una funzione main che crea la lista e gestisce l’interazione con l’utente. La funzione main è già parzialmente implementata in un file **Liste.c**.