

# 1 Luglio – Fondamenti di Informatica – Seconda Parte – 120 minuti

## COMPITO A – Esercizi di Programmazione (24 punti)

### Esercizio A1 (9 punti): Stringhe

Realizzare un'applicazione che consiste di due funzioni.

**(6pt)** Una funzione *treSoloSeConti* riceve come parametro una stringa e la modifica eliminando ogni '3' (il carattere che rappresenta il numero tre) che non è preceduto da una stringa "12". Ad esempio, se *s* = "33abc1233de3f33", la stringa deve essere modificata in "abc123def"; infatti il terzo '3' della stringa originaria è l'unico che deve rimanere a seguito della modifica, in quanto preceduto dalla sottostringa "12".

**(3pt)** Una funzione *main* che gestisce l'interazione con l'utente. La funzione *main* deve:

- chiedere all'utente se vuole introdurre una stringa, oppure recuperare una stringa da un file
- nel caso in cui l'utente voglia introdurre una stringa, la funzione *main* chiede all'utente di inserirla e la memorizza all'interno di un array di 50 caratteri e su di un file "stringhe.txt". La funzione *main* invoca la funzione *treSoloSeConti* fornendole come parametro la stringa letta; la funzione *main* stampa quindi la stringa modificata.
- nel caso in cui l'utente voglia recuperare una stringa da file, la funzione *main* legge una stringa da un file "stringhe.txt" e la memorizza all'interno di un array di 50 caratteri (stampando opportuni messaggi nel caso in cui il file non possa essere aperto, oppure non contenga alcuna stringa). Se ha effettivamente recuperato una stringa, la funzione *main* invoca la funzione *treSoloSeConti* fornendole come parametro la stringa recuperata; la funzione *main* stampa quindi la stringa modificata.

### Esercizio A2 (5 punti): Ricorsione

Realizzare un'applicazione *DueSingolaCifra* definita come segue.

**(4pt)** L'applicazione contiene una funzione *ricorsiva dueSingolaCifra* che verifica, all'interno di un array di interi ricevuto come parametro, se ogni tre interi adiacenti nella sequenza contengono almeno due interi compresi fra -9 e 9. Ad esempio, se l'array è [9, 11, 7, -2, -6], la funzione deve restituire 1, in quanto ogni tripla contiene almeno due interi maggiori o uguali a -9 e minori o uguali a 9. Infatti, nella tripla [9, 11, 7] i due interi compresi -9 e 9 sono 9 e 7, nella tripla [11, 7, -2] i due interi compresi -9 e 9 sono 7 e -2 e nella tripla [7, -2, -6] tutti gli interi sono compresi fra -9 e 9.

La funzione *dueSingolaCifra* 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 *dueSingolaCifraRic* che ha tre parametri e che realizza la ricorsione. In tal caso la funzione *dueSingolaCifra* invoca la funzione *dueSingolaCifraRic* (fornendole opportuni parametri) per calcolare il risultato.

**(1pt)** In un commento che precede la funzione *dueSingolaCifra* descrivere la specifica della funzione *dueSingolaCifra* (espressa come Input-Precondizione-Output-Postcondizione).

**(0pt)** L'applicazione contiene una funzione *main* che gestisce l'interazione con l'utente. La funzione *main* è implementata (a meno dei messaggi finali per l'utente) in un file *ricorsione.c* il cui codice può essere scaricato e copiato da moodle.

### Esercizio A3 (10 punti): Liste

Realizzare un'applicazione *Quadrati* per gestire una lista di quadrati con lati paralleli agli assi coordinati.

**(1pt)** Definire tre strutture, una per rappresentare un punto del piano, una per rappresentare un quadrato, ed una per rappresentare un elemento della lista. In particolare, l'applicazione deve gestire ciascun punto come una struttura con due campi, che rappresentano le coordinate del punto. Inoltre, l'applicazione deve gestire ciascun quadrato come una struttura con due campi, che rappresentano il punto che definisce il vertice in basso a sinistra e la lunghezza del lato.

**(1.5pt)** Definire una funzione che visualizza la lista, stampando per ciascun quadrato le informazioni ad esso associate.

**(2.5pt)** Definire una funzione che inserisce un quadrato in coda alla lista.

**(5pt)** Definire una funzione che rimuove quadrati "duplicati" dalla lista. Ovvero, dopo l'invocazione di questa funzione, la lista non deve contenere due quadrati che hanno lo stesso vertice in basso a sinistra ed il lato della stessa lunghezza.

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** che può essere scaricato da Moodle.