

# Laboratorio di programmazione strutturata (STM) - A.A. 2015-2016 -

Primo appello - giugno 2016

## ESERCIZI

Tempo a disposizione: 2 ore e mezza.

### Esercizio 1 (15 punti)

Una frazione si dice *irriducibile* se il numeratore e il denominatore non hanno divisori comuni cioè sono coprimi. Inoltre, si assuma già definita una funzione `int MCD(int x, int y)` per il calcolo del Massimo Comune Divisore.

Sia  $L$  una lista di frazioni rappresentate come coppie di interi corrispondenti al numeratore e al denominatore.

1. Si definisca il tipo nodo lista che contiene i due campi per memorizzare rispettivamente numeratore e denominatore suddetti.
2. Si scriva una funzione `semplifica` che, prende in input una lista  $L$  di frazioni e ne produce/restituisce un'altra che contiene tutte le frazioni irriducibili equivalenti a quelle della lista  $L$ . Più precisamente: se una frazione  $f$  di  $L$  è irriducibile andrà inserita direttamente nella nuova lista, se invece  $f$  può essere semplificata in una frazione irriducibile equivalente  $f'$ , allora nella nuova lista bisognerà inserire  $f'$ .

NOTA: Gli elementi della nuova lista potranno trovarsi anche in un diverso ordine rispetto a quello che hanno in  $L$ .

### Esercizio 2 (7 punti)

Scrivere una funzione `int MCD(int x, int y)` per il calcolo del Massimo Comune Divisore.

### Esercizio 3 (8 punti)

Si scriva la funzione `void ordina(int vet[], int s)` che ordina in modo *decrescente* un vettore `vet` di dimensione `s`, utilizzando l'algoritmo di ordinamento *selectionSort*.