

Laboratorio di programmazione strutturata (STM) - A.A. 2016-2017 -

Primo appello - giugno 2017

ESERCIZI

Tempo a disposizione: 2 ore e mezza.

Esercizio 1 (15 punti)

Sia P una lista (sufficientemente lunga) che contiene i numeri primi ordinati in ordine crescente. Utilizzando P , dato un numero n si vuole generare una lista F contenenti i numeri primi divisori di n .

1. Si definisca il tipo nodo lista adatto per le liste P ed F .
2. Si scriva una funzione `fattoriPrimi` che, prende in input un numero intero n , la lista P e una lista vuota F e inserisce nell'ordine tutti i divisori primi (senza ripetizioni) di n .

Suggerimento: si scorra la lista P e per ogni elemento $x \in P$ si controlli se x divide n . In caso affermativo si inserisca x in F .

Esercizio 2

1. (5 punti) Scrivere una funzione `int InComune(int x[], int y[], int n, int m)` che prende in input 2 vettori e le loro rispettive lunghezze e stampa in output gli elementi *positivi* in comune ai due vettori. La funzione restituisce tale numero di elementi positivi in comune.
2. (5 punti) Scrivere una funzione `int InComuneOrd(int x[], int y[], int n, int m)` come al punto precedente e con l'ipotesi ulteriore che i vettori x e y siano ordinati in modo crescente.
3. (5 punti) (SOLO PER CHI SOSTIENE L'ESAME DA 6 CREDITI)
Scrivere una funzione `int InComuneOrd2(int x[], int y[], int n, int m)` come al punto precedente e con l'ipotesi ulteriore che il vettore x sia ordinato in modo crescente mentre il vettore y sia ordinato in modo decrescente.

Discutere e confrontare la complessità degli algoritmi implementati in funzione di m ed n .

Esercizio 3 (5 punti)(SOLO PER CHI SOSTIENE L'ESAME DA 8 CREDITI)

Sia `char cognomeNome[10]`; un vettore inizializzato con le prime 10 lettere del vostro cognome seguite da quelle del vostro nome. Simulate l'algoritmo insertion sort sul vettore `cognomeNome` mostrando il contenuto del vettore nei vari passi (non occorre eseguire l'algoritmo fino a completo ordinamento del vettore).