

**Esame di Fondamenti di Programmazione (Canale P-Z)**  
**15 luglio 2010**  
**ore 10:00 AULA V MAT**

**Nome                                  Cognome                                  Matricola**

**Problema 1(iterazione 14/30)**

In un vettore di interi T definiamo una **sequenza crescente ordinata** come segue:

1.  $T[0]$  è il primo elemento della sequenza
2. Se  $T[i]$  è il k-esimo elemento della sequenza il seguente elemento nella sequenza crescente, cioè il (k+1)-esimo, è l'elemento  $T[j]$  di T per  $j > i$ , tale che  $T[j] > T[i]$ .

**Esempio**

Supponiamo che

$T = [2, 1, 4, 5, 3, 6, 7, 4, 2, 9, 2, 10, 8]$

Allora la sequenza crescente ordinata in T è la sequenza 2,4,5,6,7,9,10.

Progettare un **programma iterativo** che dato un vettore di interi T e la sua dimensione N, ordini T in modo tale che prima appaia la sequenza crescente ordinata di T e poi tutti gli altri elementi in un ordine qualsiasi e non necessariamente ordinati.

**Esempio.**

Se T è il vettore dell'esempio precedente, allora l'output della funzione deve essere il vettore  $T = [2, 4, 5, 6, 7, 9, 10, 1, 3, 4, 2, 8]$ .

Il progetto della funzione deve prevedere:

- spiegazione di alto livello dell'idea della soluzione (3-4 righe al massimo);
- pre e post condizione della funzione;
- variabili usate e descrizione del loro significato;
- condizione di terminazione del ciclo o dei cicli se più d'uno;
- verifica (anche informale) della terminazione del ciclo o dei cicli;
- corpo del ciclo e spiegazione del proseguimento del ciclo o dei cicli;
- implementazione in C della funzione.

## Problema 2 (ricorsione12/30)

Si progetti **una (sola) funzione ricorsiva** che dato un intero  $n > 0$  calcoli il reale

$\frac{n!}{\sqrt{2^{n(n+1)}}$ . Nella funzione si può usare senza implementarla una funzione

```
int exp (int n)
/* Pre n>0 */
/* Post exp(n) = 2^n */
```

Si ricordino e si usino le seguenti uguaglianze nel progetto della funzione ricorsiva.

1.  $2^n * 2^m = 2^{n+m}$ ;
2.  $\sqrt{n} = n^{1/2}$
3.  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$
4.  $(2^n)^m = 2^{n*m}$

Il progetto della funzioni ricorsive deve prevedere:

- pre e post-condizione;
- analisi del caso base della ricorsione;
- analisi del caso ricorsivo;
- verifica (anche informale) della terminazione della ricorsione.

**Problema 3 (6/30)**

Siano L1 e L2 due liste concatenate semplici e ordinate. Scrivere una funzione che in input L1 e L2 produca una terza lista L3 ottenuta dalla fusione ordinata di L1 e L2. Il programma deve essere efficiente.

**Spazio per Soluzioni**