Chi ha superato PI e la prova intermedia deve risolvere gli esercizi PII.2 e PII.3. Chi deve superare l'esame integrato completo deve risolvere gli esercizi PI-PII.1,PI-PII.2,PII.1 e PII.2. Chi ha superato la prova intermedia e non PI deve risolvere tutti gli esercizi tranne PI-PII.2 e PII.1. Chi ha superato PI e non la prova intermedia deve risolvere PI-PII.2, PII.1 e PII.2.

PROGRAMMAZIONE I - PROGRAMMAZIONE II
19/6/2007
Prof. Riccardo Silvestri - Prof.ssa E. Fachini

Cognome:_____

Nome:_____

N. Matricola: _____

Esercizio PI-PII.2 (Liste concatenate)

Scrivere una funzione che, presa in input una lista concatenata di interi, la modifichi in modo tale che tutti i pari precedano i dispari, nello stesso ordine in cui compaiono nella lista data.
Ad esempio, se la lista contiene (nell'ordine) i valori:

L -> 1 -> 20 -> 35 -> 40 -> 62 -> 51 -> 66

La lista che si vuole ottenere e'

L -> 20 -> 40 -> 62 -> 66 -> 1 -> 35 -> 51

Chi ha superato PI e la prova intermedia deve risolvere gli esercizi PII.2 e PII.3. Chi deve superare l'esame integrato completo deve risolvere gli esercizi PI-PII.1,PI-PII.2,PII.1 e PII.2. Chi ha superato la prova intermedia e non PI deve risolvere tutti gli esercizi tranne PI-PII.2 e PII.1. Chi ha superato PI e non la prova intermedia deve risolvere PI-PII.2, PII.1 e PII.2.

PROGRAMMAZIONE II - 19/6/2007
Prof.ssa E. Fachini

Cognome:_____

Nome:_____

N. Matricola: _____

Esercizio PII.1 (Code e Pile)

Considerate una sequenza di interi letti da input e definite una funzione C che li stampa in modo tale che tutti i pari precedano i dispari, nello stesso ordine in cui vengono letti.

Ad esempio, se la sequenza è:

1 , 20 , 35 , 40 , 62 , 51 , 66

La stampa che si vuole ottenere e'

20 , 40 , 62 , 66 , 1 , 35 , 51

La funzione deve utilizzare come struttura dati di appoggio una pila o una coda (scegliete la struttura piu' opportuna). Si scrivano anche precondizioni e postcondizioni della funzione.
Si fornisca la specifica (prototipi, precondizioni e postcondizioni e non il codice!!) delle operazioni di gestione dell'ADT che si intende usare (se una coda: accodamento ed estrazione, controllo su coda vuota o piena, o se una pila: push, pop, controllo su pila piena o vuota), in modo che sia possibile prevedere un'implementazione dell'ADT sia con una lista concatenata che con un vettore.

Chi ha superato PI e la prova intermedia deve risolvere gli esercizi PII.2 e PII.3. Chi deve superare l'esame integrato completo deve risolvere gli esercizi PI-PII.1,PI-PII.2,PII.1 e PII.2. Chi ha superato la prova intermedia e non PI deve risolvere tutti gli esercizi tranne PI-PII.2 e PII.1. Chi ha superato PI e non la prova intermedia deve risolvere PI-PII.2, PII.1 e PII.2.

PROGRAMMAZIONE II - 19/6/2007
Prof.ssa E. Fachini

Cognome:_____

Nome:_____

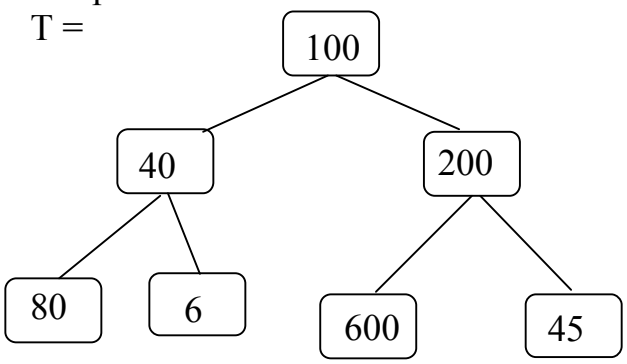
N. Matricola: _____

Esercizio PII.2 (Ricorsione su alberi)

Definire una funzione che preso in input un albero binario di interi (senza ripetizioni) restituisce 1 se la sequenza dei nodi di livello k (k >=0) letta da sinistra verso destra e' crescente e 0 altrimenti.

Esempio

su input
T =



se k=1 il risultato è 1, se k=2 il risultato è 0.

Chi ha superato PI e la prova intermedia deve risolvere gli esercizi PII.2 e PII.3. Chi deve superare l'esame integrato completo deve risolvere gli esercizi PI-PII.1,PI-PII.2,PII.1 e PII.2. Chi ha superato la prova intermedia e non PI deve risolvere tutti gli esercizi tranne PI-PII.2 e PII.1. Chi ha superato PI e non la prova intermedia deve risolvere PI-PII.2, PII.1 e PII.2.

PROGRAMMAZIONE II - 22/2/2007
prof.ssa E. Fachini

Cognome:_____

Nome:_____

N. Matricola: _____

Esercizio PII.3 (Testing)

Si individuino opportuni dati di test per la funzione sottostante evidenziando per quali dati si è seguito l'approccio a scatola nera e per quali quello a scatola trasparente. Si dia una breve motivazione della scelta.

```
ListaPtr merge (ListaPtr L1, ListaPtr L2)
/*fonde  due liste ordinate L1 e L2 (modificando entrambe)
in un'unica lista  con gli elementi di entrambe
postc: restituisce una lista ordinata
che contiene  tutti gli elementi di L1 e di L2  */
{if (!L2) return L1;
if (!L1) return L2;
/*se  L1 o L2 e' vuota l'altra  contiene gia'  tutti gli elementi  */
 if (L2 -> elem < L1 -> elem)
{L2 -> next = merge(L1,L2 ->next);
return L2;}
else
{L1 -> next = merge(L1->next,L2);
return L1;}
}
```