

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

- SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO PER LO SVOLGIMENTO DELLA PROVA
- RICONSEGNARE TUTTI I FOGLI. NON SCRIVERE A MATITA.
- SPEGNERE I TELEFONINI
- NON È POSSIBILE UTILIZZARE CALCOLATRICI
- È POSSIBILE CONSULTARE SOLO LA DISPENSA SUL LINGUAGGIO ASSEMBLER DISPONIBILE SULLA CATTEDRA
- I PRIMI DUE ESERCIZI VALGONO 10 PUNTI; GLI ULTIMI 2 VALGONO 5 PUNTI
- TEMPO PER LA PROVA 2 ORE

ESERCIZIO 1

2) Sia data la struttura seguente:

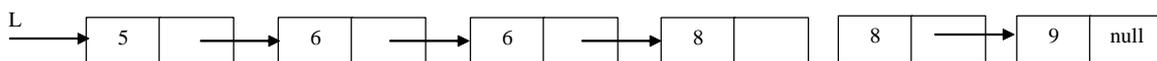
```
struct elem{int info; elem* pun;};
```

Scrivere una funzione che, data una lista ordinata L di elementi di tipo elem passata come argomento alla funzione, modifica la lista duplicando tutti gli elementi che hanno nel campo informazione un numero pari. La lista L deve essere mantenuta ordinata.

Ad esempio, data la lista L seguente:



la lista L viene modificata come segue:



ESERCIZIO 2

Scrivere una funzione che legge un intero n da un file, il cui nome e' passato come argomento alla funzione, e restituisce una matrice di dimensione nxn inizializzata come segue:

```
1 1 1 1 1 1 1
1 2 2 2 2 2 1
1 2 3 3 3 2 1
1 2 3 4 3 2 1
1 2 3 3 3 2 1
1 2 2 2 2 2 1
1 1 1 1 1 1 1
```

ESERCIZIO 3

Sia data la struttura seguente:

```
struct elem{int info; elem* pun;};
```

Scrivere una funzione ricorsiva che calcola la somma dei soli elementi multipli di 3 che si trovano in posizione pari di una lista di elementi di tipo elem. Si assuma che il primo elemento della lista sia in posizione 1. Mostrare anche l'esempio di chiamata della funzione.

Nell'esempio seguente la funzione restituisce 15.



ESERCIZIO 4

- 1) Rappresentare gli interi -12 e -5 in complemento a due su 5 bit. Rappresentare inoltre la loro somma, nella stessa base e numero di bit.
- 2) Dire cosa mostra a video il seguente programma, **riportando sul foglio anche i passaggi salienti** (non occorre mostrare il contenuto di tutti i registri per ogni linea di codice, basta mostrare qualche contenuto di registro nei momenti chiave).

```
#.GLOBAL _main
.EQU N,5

# Sezione dati
vett: .BYTE 0x05,0x04,0x03,0x10,0x0A

_main: MOV $0, %DL
      MOV $N, %DH
      MOV $vett, %EBX
loop: MOV (%EBX), %CH
      MOV %CH, %CL
      SHR %CL
      JC label
      ADD %CH, %DL
label: INC %EBX
      DEC %DH
      JZ fine
      JMP loop
fine: MOV %DL, %AL
      call outbyte
      RET
```

NOTE

Nome: outbyte

Azione: Visualizza sul monitor il contenuto del registro AL sottoforma di due cifre esadecimali.