

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

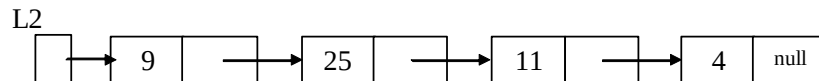
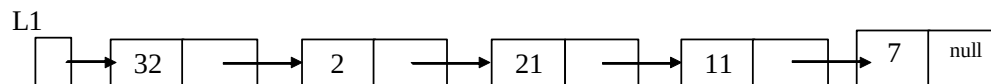
- SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO PER LO SVOLGIMENTO DELLA PROVA
- RICONSEGNARE TUTTI I FOGLI. NON SCRIVERE A MATITA.
- SPEGNERE I TELEFONINI
- NON È POSSIBILE UTILIZZARE CALCOLATRICI
- È POSSIBILE CONSULTARE SOLO LA DISPENSA SUL LINGUAGGIO ASSEMBLER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

ESERCIZIO 1

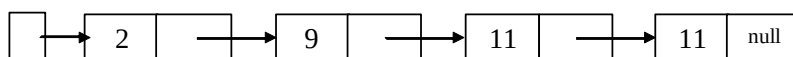
Sia data la struttura seguente `struct elem {int info; elem* pun;};`

Scrivere una funzione che date due liste di elementi di tipo `elem`, restituisce una nuova lista che contiene tutti gli elementi in posizione pari della prima lista e tutti gli elementi in posizione dispari della seconda. Il primo elemento di una lista ha posizione 1. La lista deve essere ordinata in ordine crescente. Le liste L1 ed L2 non devono essere modificate.

Per esempio, la funzione viene chiamata con le liste seguenti:



restituisce la lista



ESERCIZIO 2

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

`n = 5`

```
1 1 1 1 1
- 1 1 1 -
- - 1 - -
- 1 1 1 -
1 1 1 1 1
```

`n=4`

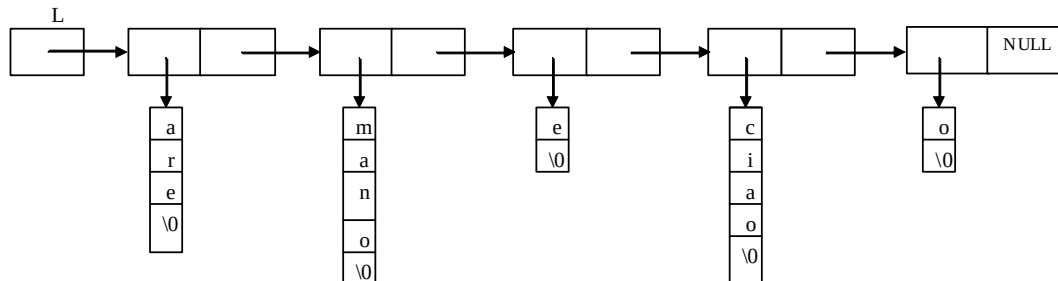
```
1 1 1 1
- 1 1 -
- 1 1 -
1 1 1 1
```

ESERCIZIO 3

Sia data la struttura seguente `struct elem {char* nome; elem* pun;};`

Scrivere una funzione ricorsiva che, data una lista di elementi di tipo `elem` ed un intero `n` restituisce il numero di stringhe di lunghezza maggiore di `n` presenti nella lista

Ad esempio, data la lista `L` seguente e l'intero 3, la funzione restituisce 2:



ESERCIZIO 4

1) Rappresentare l'intero (-73) in complemento a 2 su 8 bit.

2) Dire cosa mostra a video il seguente programma fornendo come ingresso il numero meno significativo del proprio numero di matricola, **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).

```
.INCLUDE "G:/prg/amb_MinGW/utility"
#.GLOBAL _main
# Sezione codice
_main:
```

```
        MOV  $0, %BL
        CALL input
        SUB  '$'0', %AL
        SHL  $2, %AL
        ADD  $12, %AL
        CALL outbyte
        SHR  $3, %AL
        JNC  lab1
lab1:    MOV  $1, %BL
        SHR  $1, %AL
        JNC  lab2
lab2:    ADD  $2, %BL
        SHR  $1, %AL
        JNC  lab3
lab3:    ADD  $4, %BL
        MOV  %BL, %AL
        ADD  '$'0', %AL
        CALL output
        RET
```

NOTE

Nome: input

Azione: legge la codifica ASCII di un carattere da tastiera. Immette tale codifica nel registro AL.

Nome: output

Azione: Visualizza sul monitor il carattere il cui codice ASCII è contenuto nel registro AL.

Nome: outbyte

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