

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

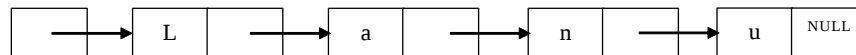
- SCRIVERE IL PROPRIO NOME, COGNOME, DATA E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO PER LO SVOLGIMENTO DELLA PROVA
- RICONSEGNARE TUTTI I FOGLI. NON SCRIVERE A MATITA.
- SPEGNERE I TELEFONINI
- NON È POSSIBILE UTILIZZARE CALCOLATRICI
- È POSSIBILE CONSULTARE SOLO LA DISPENSA SUL LINGUAGGIO ASSEMBLER DISPONIBILE SULLA CATTEDRA
- I PRIMI DUE ESERCIZI VALGONO 10 PUNTI; GLI ULTIMI 2 VALGONO 5 PUNTI
- TEMPO PER I PRIMI TRE ESERCIZI: 1 ORA E 45 MINUTI TEMPO PER L'ESERCIZIO 4: 30 MIN.

Esercizio 1

Sia data la struttura seguente

```
struct elem {char lettera; elem* pun;};
```

Scrivere una funzione che, data una stringa di lettere (maiuscole e/o minuscole) passata come argomento alla funzione, restituisce una lista di strutture di tipo `elem`, dove il campo `lettera` di ciascuna struttura contiene una lettera della stringa. La lista deve essere ordinata crescente. Per esempio, se la stringa passata alla funzione è “Luna”, la funzione restituisce la seguente lista:

**Esercizio 2**

Scrivere una funzione che, data una matrice `mat` di interi passata come argomento alla funzione, restituisca la somma degli elementi in posizione (i,j) , in cui almeno uno tra l'indice di riga i e l'indice di colonna j siano multipli di 2. La matrice `mat` ha dimensione $n \times m$, dove n e m sono variabili. Per esempio, data la matrice seguente

```

1  2  3
4  5  6
7  8  9
2  3  0
  
```

la funzione restituisce 33 ($=3+6+7+8+9+0$)

Esercizio 3

Scrivere una funzione ricorsiva che, dato un intero n positivo e dispari, stampa un triangolo di asterischi con il vertice rivolto verso il basso. La funzione può contenere istruzioni ripetitive e può avere un qualsiasi numero di argomenti. Se $n=5$, la funzione stampa:

```

*****
***
*
```

Esercizio 4

4.1

Data la rappresentazione **0.10010** in virgola fissa su 6 bit, dire a quale reale r essa corrisponde.

Inoltre trovare la rappresentazione ottale del naturale **7A** espresso in base 16.

4.2 Dire cosa stampa a video il seguente programma assembler, qualora vengano fornite in ingresso le due cifre meno significative del proprio numero di matricola.

```
_main:  CALL  inbyte
        MOV   %AL, %BL
        SHR   $1, %BL
        JC    lab
        SHR   $1, %BL
        JNC   lab
        MOV   $'S', %AL
        CALL  output
        JMP   fine
lab:    MOV   $'N', %AL
        CALL  output
fine:    RET

.INCLUDE "utility"
```

PROMEMORIA

CALL inbyte	legge da tastiera due caratteri (purché compresi tra '0' e '9' oppure tra 'A' e 'F'), li interpreta come cifre esadecimali e li utilizza per inizializzare AL. Esempio: qualora venissero inseriti '4' e 'B', in AL ci finirebbe 0100-1011 (0x4B)
SHR \$1, reg	effettua lo shift a destra del contenuto del registro <i>reg</i> di una posizione (da sinistra entra uno zero). Inoltre, il bit meno significativo finisce nel carry flag.
JC label	salta a <i>label</i> tutte le volte in cui il risultato dell'ultima operazione ha settato il carry flag
JNC label	salta a <i>label</i> tutte le volte in cui il risultato dell'ultima operazione <i>*NON*</i> ha settato il carry flag
JMP label	salta a <i>label</i> , incondizionatamente
CALL output	stampa a video il carattere ASCII la cui codifica si trova in AL. Esempio: qualora in AL si trovasse 0011-0101 stamperebbe a video il carattere '5'