

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 ESERCIZI 4.1 E 4.2: 30 MIN.

ESERCIZIO 1

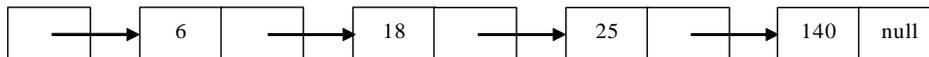
Sia data la struttura seguente

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

Scrivere una funzione che costruisce una lista di elementi di tipo elem, ordinata per valori crescenti del campo info. I valori degli elementi della lista vengono letti da un file il cui nome è passato come argomento alla funzione. L'operazione ha successo se il file contiene solo interi terminati dal carattere '.' In caso di errore nella lettura dei dati in ingresso, la funzione deve restituire una lista vuota. Per esempio, se il file contiene:

25 140 6 18 .

la funzione restituisce la lista seguente:



ESERCIZIO 2

Scrivere una funzione che, data una matrice mat di $n \times m$ elementi di tipo intero ed un intero p, restituisce true se esistono almeno **3 elementi adiacenti** uguali a p in una riga e in una colonna della matrice, false altrimenti.

Per esempio, data la matrice mat (4×6), la funzione restituisce true se $p=1$ oppure $p=0$; la funzione restituisce false altrimenti:

```
mat = 0 1 1 1 7 0
      0 0 0 0 1 0
      2 0 1 0 1 3
      1 6 2 0 1 8
```

ESERCIZIO 3

Scrivere una funzione ricorsiva che dato un intero k, stampi su video k "ciao" seguiti da k "mondo" senza utilizzare istruzioni iterative. Per esempio, se $k = 4$, la funzione stampa su uscita standard

ciao ciao ciao ciao mondo mondo mondo mondo

ESERCIZIO 4

4.1

Data il numero 56 in base 9, trovarne la rappresentazione in base 10 e quella in base 5.

4.2 Dire cosa stampa a video il seguente programma assembler, qualora vengano fornite in ingresso la terzultima e l'ultima cifra del proprio numero di matricola.

```
_main:  MOV    $4, %DH
        MOV    $0, %AH
        CALL  inbyte
        AND    $0x0F, %AL
        CALL  outbyte
lab1:   SHR    $1, %AL
        JNC   lab2
        INC   %AH
lab2:   DEC    %DH
        JZ    lab3
        JMP   lab1
lab3:   CMP    $1, %AH
        JE    lab4
        MOV   '$N', %AL
        JMP   fine
lab4:   MOV    '$S', %AL
fine:   CALL  output
        RET
```

```
.INCLUDE "utility"
```

PROMEMORIA

CALL inbyte	legge da tastiera due caratteri (purché compresi fra '0' e '9' oppure fra '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).
CALL outbyte	stampa a video le due cifre esadecimali associate al contenuto del registro AL. Esempio: Qualora AL contenesse 0100-1011, stamperebbe a video "4B".
SHR \$1, REG	effettua lo shift a destra del contenuto del registro REG di 1 posto. Il bit meno significativo finisce nel Carry Flag. Da sinistra entra un bit a zero.
JNC label	Salta a <i>label</i> tutte le volte in cui l'ultima operazione NON ha settato il carry flag.
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'.
CMP IMM,REG	confronta l'operando immediato IMM con il contenuto del registro REG
JE label	salta a <i>label</i> solo nel caso in cui i due operandi confrontati nella precedente istruzione CMP erano uguali