

**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:

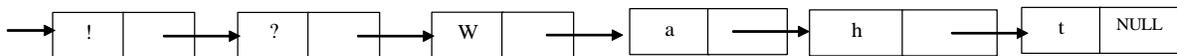
```
struct elem {char info; elem* pun;};
```

Scrivere una funzione che prende come argomento un file di caratteri e costruisce una lista di elementi di tipo elem. La lista ha un elemento per ogni carattere letto dal file, senza duplicati e deve essere ordinata in ordine crescente secondo la codifica ASCII dei caratteri. Le spaziature non vengono considerate.

Per esempio, se la funzione viene chiamata con il file seguente:

```
What !? eof
```

la funzione restituisce la lista:



**ESERCIZIO 2**

Sia data la struttura seguente:

```
struct punto {double x; double y;}
```

Scrivere una funzione che, dati due vettori v1 e v2 di lunghezza  $n > 0$  di elementi di tipo reale, restituisca un vettore di elementi di tipo punto di dimensione uguale alla dimensione massima fra quella di v1 e v2.

L'elemento i-esimo del vettore restituito deve avere come parte x l'elemento i-esimo di v1 e come parte y l'elemento i-esimo di v2. Se i-esimo elemento del vettore non esiste, assumere come valore 0.

Ad esempio, dati i vettori  $v1 = [1.0, 3.2]$  e  $v2 = [0.2, 5.1, 7.3]$ , la funzione restituirà il vettore  $[(1.0, 0.2), (3.2, 5.1), (0.0, 7.3)]$ .

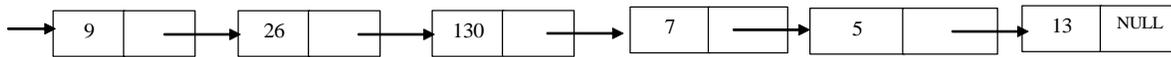
### ESERCIZIO 3

Sia data la struttura seguente:

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

Scrivere una funzione ricorsiva che prende come argomento una lista L di elementi di tipo `elem` e restituisce il numero di elementi con valore del campo informazione multiplo di 13.

Per esempio se la funzione è chiamata con la lista seguente, la funzione restituisce 3.



### ESERCIZIO 4

#### 4.1) ESERCIZIO 4.1

Trovare la rappresentazione in base nove del naturale 2210 espresso in base tre.

4.2) Dire cosa stampa a video il seguente programma quando in ingresso viene immesso (sotto forma delle due cifre esadecimali associate) la terzultima e l'ultima cifra del proprio numero di matricola.

```
#.GLOBAL _main
```

```
_main:  NOP
        CALL  inbyte
        AND   $0xFC,%AL
        OR    $0x0C,%AL
        CALL  outbyte
        CALL  subr
        CALL  outbyte
        RET
```

```
subr:   ADD   $3,%AL
        ROL  $4,%AL
        RET
```

```
.INCLUDE "utility"
```

#### NOTE

- **inbyte** attende che vengano immesse da tastiera due cifre esadecimali e le copia nella parte alta e bassa di AL, rispettivamente (esempio: "2F" => AL=00101111).
- **outbyte** stampa le codifiche ASCII corrispondenti alle due cifre esadecimali che compongono il registro AL (esempio AL=00101111 => "2F")
- **ADD SORG, DEST** equivale all'operazione  $DEST \leftarrow DEST + SORG$
- **ROL \$X, %SORG** effettua la rotazione circolare a sinistra di X passi del registro SORG