

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

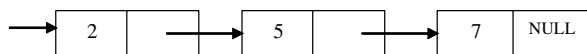
ESERCIZIO 1

Sia data la struttura seguente:

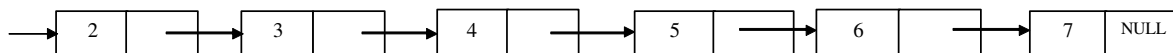
```
struct elem {int info; elem* pun;};
```

Scrivere una funzione che prende come argomento una lista ordinata e senza duplicati di elementi di tipo `elem` e la completa inserendo fra ogni coppia di elementi tutti gli eventuali elementi mancanti.

Per esempio, se la funzione viene chiamata con la lista seguente:



la funzione modifica la lista come segue:



ESERCIZIO 2

Scrivere una funzione che prende in ingresso una matrice di interi ordinata per righe per valori non decrescenti e un intero `k`. La funzione incrementa di 5 l'elemento in posizione (k,k) , e riordina gli elementi della matrice. Se la posizione `k` non è valida, la funzione restituisce `false` e lascia la matrice inalterata; altrimenti modifica la matrice e restituisce `true`.

Per esempio, se la funzione viene chiamata con la matrice seguente di dimensione 3×4 e `k=1`:

0	1	2	2
2	3	4	6
7	7	7	9

$\rightarrow 3+5 = 8$

la matrice verrà modificata come segue e la funzione restituisce `true`:

0	1	2	2
2	4	6	7
7	7	8	9

ESERCIZIO 3

Scrivere una funzione ricorsiva che prende come argomento un vettore `VETT` di stringhe ed una stringa `s`. La funzione restituisce `true` se tutti gli elementi del vettore hanno lunghezza inferiore alla stringa `s`. La funzione restituisce `false` altrimenti.

ESERCIZIO 4

1) Si consideri $Y=01100011$ se il proprio numero di matricola è pari, $Y=01011110$ altrimenti. Trovare la rappresentazione in base 4 di Y .

2) Dire cosa stampa a video il seguente programma quando in ingresso viene immesso (sotto forma delle due cifre esadecimali corrispondenti) il naturale Y del punto precedente.

```
#.GLOBAL _main

_main:  CALL    inbyte
        MOV     $255,%DL
        SUB     %AL,%DL
        MOV     %DL,%AL
        CALL    outbyte
        MOV     $1,%BL
        MOV     $8,%CL
loop:   OR      %BL,%AL
        SHL     $1,%BL
        DEC     %CL
        JNZ     loop
fine:   CALL    outbyte

.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")
- **SUB SORG, DEST** equivale all'operazione $DEST = DEST - SORG$
- **OR SORG, DEST** equivale all'operazione $DEST |= SORG$
- **SHL \$1, SORG** effettua la traslazione a sinistra di un bit del registro `SORG`.