

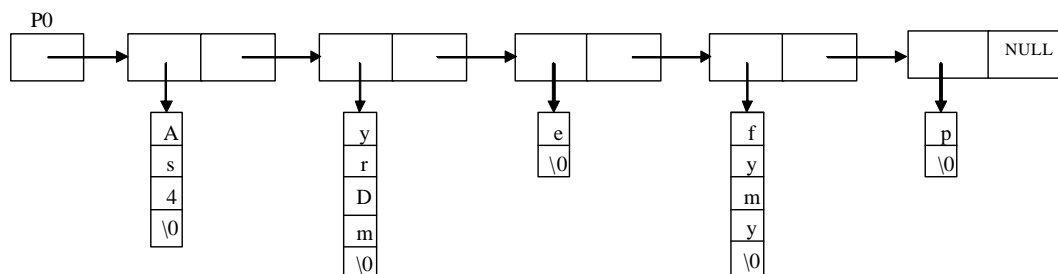
NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

- SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;
- SPEGNERE I TELEFONINI;
- È POSSIBILE CONSULTARE SOLO LA DISPENSA DI ASSEMBLER;
- NON È POSSIBILE UTILIZZARE CALCOLATRICI;
- PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO CHE SI STA RISOLVENDO. PER ESEMPIO, SCRIVERE "ESERCIZIO N. 1" QUANDO SI STA RISOLVENDO L'ESERCIZIO N. 1;
- I PRIMI DUE ESERCIZI VALGONO 10 PUNTI; GLI ULTIMI 2 VALGONO 5 PUNTI.
- ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.

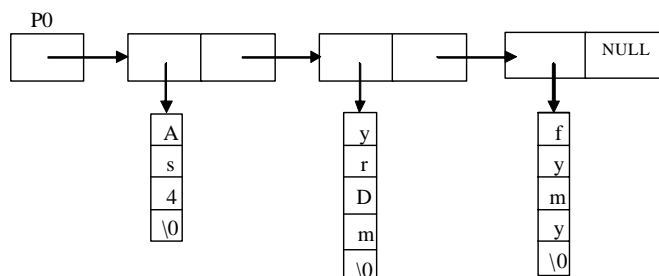
Esercizio 1

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

Scrivere una funzione che, data una lista di elementi di tipo `elem`, elimina dalla lista tutti gli elementi con campo `nome` una stringa di lunghezza minima. Una stringa ha lunghezza minima se non esiste nessuna stringa di lunghezza minore nella lista. Per esempio, se la funzione viene chiamata con la lista seguente,



La lista viene modificata come segue:



Esercizio 2

Scrivere una funzione che legge la dimensione di una matrice quadrata di interi e successivamente tutti gli elementi della matrice da file. Gli elementi sono memorizzati nel file per righe. Se ci sono errori durante la lettura la funzione restituisce `false`. Se l'operazione di lettura ha successo, la funzione restituisce `true` se tutte le righe sono formate da numeri ordinati in maniera crescente o decrescente e `false` in tutti gli altri casi. Per esempio, la funzione se legge dal file la matrice A restituisce `true`; se legge dal file la matrice B o C restituisce `false`.

	1 2 5 6	1 1 8	1 5 2
A =	7 6 4 3	B = 3 5 6	C = 4 4 1
	1 4 6 7	6 3 1	7 2 9
	3 4 8 9		

Esercizio 3

Scrivere una funzione ricorsiva che analizzi una stringa di cifre e restituisca `true` se la stringa è palindroma, `false` altrimenti. Una stringa di cifre è palindroma se il numero corrispondente rimane lo stesso se letto da sinistra verso destra e da destra verso sinistra.

Per esempio, la stringa 1421 non è palindroma. Le stringhe 121 e 537735 sono invece palindrome.

Esercizio n 4 (Anno accademico 2011-2012)

1) Data la rappresentazione $(234)_5$ in base 5, trasformare il numero in base 2.

2) Si dica cosa visualizza a video il seguente programma assembler.

```
.GLOBAL _main
.EQU N,7
vett: .ASCII "ZaNzarA"
_main:
MOV     $0,AH
MOV     $vett,EBX
MOV     $N,CL

loop:   MOV     (EBX),AL
        CMP     $64,AL
        JG      lab_1
        JMP     lab_3

lab_1:  CMP     $91,AL
        JL      lab_2
        JMP     lab_3

lab_2:  INC     AH

lab_3:  INC     EBX
        DEC     CL
        JZ      fine
        JMP     loop
```

```
fine:   MOV     AH,AL
        CALL    outbyte
        CALL    newline
        RET
```

NOTE

Nome: outbyte

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

Esercizio n 4 (Anni accademici precedenti 2011-2012)

1) Data la rappresentazione $(234)_5$ in base 5, trasformare il numero in base 2.

2) Si mostri l'uscita a video del programma C++ seguente con input 1, 0 e -1:

<pre>#include<iostream> using namespace std; class ecc { public: ecc() { cout << "nuova eccezione" << endl;} }; class ecc1: public ecc { public: ecc1() { cout << "ecc1" << endl; } }; class ecc2: public ecc { public: ecc2() { cout << "ecc2" << endl;} }; int g(int x) { if (x <=0) throw ecc1(); return x; }</pre>	<pre>int f(int x) { try { if (x==0) throw ecc(); if (x>1) throw ecc2(); if (x==1) throw 0; return x; } catch (ecc1 &e){ cout << "ecc f: " << endl;} return x ; } int main(){ int x; cin >>x; try{ int k = g(x); cout << f(k) << endl; } catch (ecc &e){ cout << "ecc main" << endl; } catch (int n){ cout << "ecc int main: " << n << endl; } catch(...){ cout << "ecc default main" << endl; } return 0; }</pre>
--	---