

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

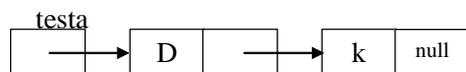
- SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;
- SPEGNERE I TELEFONINI;
- È POSSIBILE CONSULTARE SOLO LA DISPENSA DI ASSEMBLER;
- NON È POSSIBILE UTILIZZARE CALCOLATRICI;
- PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO CHE SI STA 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

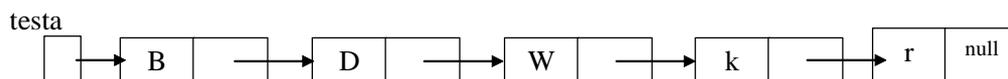
Sia data la struttura seguente `struct elem {char info; elem* pun;};`

Scrivere una funzione che, data una lista ordinata di elementi di tipo `elem`, ed un file di caratteri, legge i caratteri dal file e per ogni carattere letto che rappresenta una lettera maiuscola o minuscola dell'alfabeto inserisce un elemento nella lista con campo `info` uguale al carattere letto, mantenendo la lista ordinata.

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



Se nel file sono presenti i seguenti caratteri: `W r B eof`
la lista viene modificata come segue:



Esercizio 2

Scrivere una funzione che data una matrice di interi composta di `r` righe e `c` colonne, ed un intero `k`, restituisce una matrice di `r` righe e `c` colonne con gli elementi calcolati come segue. L'elemento in posizione (i, j) è uguale ad $A * B$, dove A è la somma dei `k` elementi che precedono e dei `k` elementi che seguono (i, j) sulla riga `i`, e B è la somma dei `k` elementi che precedono e dei `k` elementi che seguono (i, j) sulla colonna `j`. Assumere $r \geq 2$, $c \geq 2$ e $k \geq 1$.

Se la funzione viene chiamata con la matrice M seguente di dimensione 3×5 e con `k` uguale a 2, la funzione restituisce la matrice M' :

$$M = \begin{matrix} 2 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{matrix} \quad M' = \begin{matrix} 3 & 0 & 0 & 5 & 1 \\ 0 & 7 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{matrix}$$

Esercizio 3

Scrivere una funzione ricorsiva che prende come argomento un naturale dispari N maggiore di 1 e stampa a video una 'X' di asterischi con numero di asterischi sulle diagonali uguale a N.

Esempio con N=5:

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

Esercizio n 4 (Anno accademico 2011-2012)

- 1) Data la rappresentazione $(11100011)_2$ in complemento a 2, trasformare il numero in base 10.
- 2) Si dica cosa visualizza a video il seguente programma assembler.

```
.GLOBAL _main
#.GLOBAL _main

# Sezione dati
.EQU base,0xFB

# Sezione codice
_main:
    MOV $base, %BL
    MOV $3, %CL

loop:
    ADD $2, %BL
    JC lab2

lab1:
    MOV $'D', %AL
    call output
    DEC %CL
    JZ fine
    JMP loop

lab2:
    MOV $84, %AL
    call output
    DEC %CL
    JZ fine
    JMP loop

fine:
    call pause
    RET
```

NOTE

Nome: output

Azione: Visualizza sul monitor il carattere la cui codifica ASCII è contenuta nel registro AL.

Esercizio n 4 (Anni accademici precedenti 2011-2012)

- 1) Data la rappresentazione $(11100011)_2$ in complemento a 2, trasformare il numero in base 10.
- 2) Si mostri l'uscita a video del programma C++ seguente con input 1, 0 e -1:

<pre>#include<iostream> using namespace std; int g (int n){ cout << n << endl; try { if (n==1) throw 1; if (n==0) throw 'c'; cout << n << endl; } catch(char) { cout << "ecc. char da g" << endl;} cout << "fine g" << endl; return n+10; }</pre>	<pre>int f(int y) { return g(y+1); } int main(){ int x; cin >> x; try { cout << x << endl; if (x==1) throw 'c'; cout << f(x) << endl; } catch(char) {cout << "ecc. char da main" << endl;} catch(int) { cout << "int da main" << endl;} catch(...) { cout << "default da main" << endl;} cout << "fine main" << endl; return 0; }</pre>
---	--