

**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* puni;};`

Scrivere una funzione che prende in ingresso una stringa di caratteri e restituisce una lista con un elemento per ogni lettera dell'alfabeto presente nella stringa. *La funzione non deve far uso di funzioni ausiliarie.* La lista deve essere ordinata e non deve contenere duplicati. Le lettere maiuscole e minuscole sono considerate separatamente. Per esempio, se la stringa passata alla funzione è "oho ABRA2 oho" la lista restituita è:



**Esercizio 2**

Scrivere una funzione che, dato in ingresso un intero N ed un nome di file, legga da tale file una matrice NxN di naturali, eventualmente riempiendo di zeri i valori mancanti.

La funzione deve restituire `true` se tutte le colonne in posizione pari contengono solo elementi pari e tutte le colonne in posizione dispari contengono solo elementi dispari. La funzione restituisce `false` altrimenti. Le colonne sono numerate a partire da 0 e lo 0 un numero pari).

Se la funzione viene chiamata con l'intero 5 e legge dal file la matrice seguente, la funzione restituisce `true`:

2	1	4	1
4	3	0	7
0	1	0	1
0	5	2	1

**Esercizio 3**

Scrivere una funzione ricorsiva che analizzi una stringa di cifre e, assumendo che sia "palindroma", ritorni `true` se la stringa è "piramidale" e `false` altrimenti.

Ricordiamo che una stringa è palindroma se, leggendo la stringa da destra oppure da sinistra, le cifre rimangono le stesse. Una stringa di cifre piramidale se le cifre della prima metà sono ordinate in ordine strettamente crescente e quelle della seconda metà in ordine strettamente decrescente.

Per esempio, se la funzione è chiamata con la stringa 12321 restituisce `true`; se la funzione è chiamata con le stringhe 12021, 45554 e 3113 restituisce `false`.

#### Esercizio n 4 (Anno accademico 2011-2012)

- 1) Data la rappresentazione (223)<sub>5</sub> in base 5, trasformarla in base 7.
- 2) Si dica cosa visualizza a video il seguente programma assembler.

```
#.GLOBAL _main
.EQU N,8                                fine:
                                         MOV     %AH,%AL
                                         call    outbyte
                                         call    newline

# Sezione dati
vett: .ASCII "ZaNzARA"

_main:  MOV     $0,%AH                   ADD     $48,%AL
        MOV     $vett,%EBX              call    output
        MOV     $N,%CL                  call    newline

loop:   MOV     (%EBX),%AL               call    pause
        CMP     $0x61,%AL               RET
        JE      lab_2
        JMP     lab_1

lab_1:  CMP     $0x41,%AL
        JE      lab_2
        JMP     lab_3

lab_2:  INC     %AH
lab_3:  INC     %EBX
        DEC     %CL
        JZ      fine
        JMP     loop
```

#### NOTE

##### Nome: outbyte

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

##### 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 (223)<sub>5</sub> in base 5, trasformarla in base 7.
- 2) Si mostri l'uscita a video del programma C++ seguente:

<pre>#include &lt;iostream&gt; using namespace std; class A{ public: int x; A(int n=1) { x = n; cout &lt;&lt; "nuovo A " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; } void f() { cout &lt;&lt; "A:f() " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; } virtual A() { cout &lt;&lt; "via A" &lt;&lt; endl; } }; class B: public A{ public: int x; B(): A(5) { x = 7; cout &lt;&lt; "nuovo B " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; } virtual void f() { cout &lt;&lt; "B:f() " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; }     B() { cout &lt;&lt; "via B" &lt;&lt; endl; } }; class C: public B{ A obj; public: C(int k = 2) { x=k; cout &lt;&lt; "nuovo C " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; } void f() { cout &lt;&lt; "C:f() " &lt;&lt; "x=" &lt;&lt; x &lt;&lt; endl; }     C() { cout &lt;&lt; "via C" &lt;&lt; endl; } };</pre>	<pre>int main(){ C* pc = new C; A* pa = pc; B* pb = pc; pa-&gt;f(); pb-&gt;f(); pc-&gt;f(); delete pb; return 0; }</pre>
---	--