

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
- L'ESERCIZIO 4 È DIVERSO A SECONDA DELL'A.A. IN CUI È STATO SEGUITO IL CORSO
- TEMPO PER LA PROVA 2 ORE

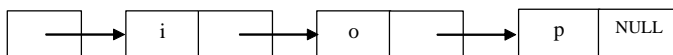
ESERCIZIO 1

Sia data la struttura seguente:

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

Scrivere una funzione che prende in ingresso una stringa `st` e restituisce una lista di elementi di tipo `elem`. La lista restituita deve contenere un elemento per ogni carattere diverso della stringa `st` (la lista non deve contenere duplicati) e deve essere ordinata in ordine crescente per il valore del campo informazione degli elementi.

Per esempio, se la funzione viene chiamata con la stringa "pippo", la lista restituita è la seguente:



ESERCIZIO 2

Scrivere una funzione che prende in ingresso il nome di un file, legge interi dal file terminati dal carattere '.' e restituisce una matrice quadrata di dimensione 8x8 costruita come segue: diagonale principale inizializzata con il massimo dei valori letti; diagonale secondaria inizializzata con il minimo dei valori letti; tutti gli altri elementi inizializzati con la somma dei numeri letti.

Se viene rilevato un errore durante la lettura dal file oppure il file ha come primo elemento il carattere '.', la matrice restituita deve essere inizializzata con tutti gli elementi uguali a 0.

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

3 5 1 .

la matrice restituita è la seguente:

5	9	9	9	9	9	9	1
9	5	9	9	9	9	1	9
9	9	5	9	9	1	9	9
9	9	9	5	1	9	9	9
9	9	9	1	5	9	9	9
9	9	1	9	9	5	9	9
9	1	9	9	9	9	5	9
1	9	9	9	9	9	9	5

ESERCIZIO 3

Scrivere una funzione ricorsiva che dato un vettore di interi, passato come argomento alla funzione, restituisce `true` se il vettore è ordinato in ordine decrescente, `false` altrimenti. La funzione deve poter essere chiamata con vettori di varie dimensioni.

ESERCIZIO 4 (Anno accademico 2011-2012)

1) Data la rappresentazione $(342)_5$ in base 5, trasformarla in base 2.

2) In relazione al seguente programma, indicare le cifre esadecimali che vengono mostrate a video dalle due chiamate alla routine 'outbyte', assumendo che venga letto il ingresso l'esadecimale 0xA6 (attraverso i due caratteri 'A' e '6' forniti da tastiera):

```
#.GLOBAL _main

# Sezione dati
.EQU mask1,0xB8
.EQU mask2,0x9B
.EQU mask3,0x8A
.EQU mask4,0x3D

# Sezione codice
_main:    call inbyte
         OR    $mask1, AL
         AND   $mask2, AL
         call outbyte

         SHR   $4, AL
         JC    L1
         OR    $mask3, AL
L1:       OR    $mask4, AL
         call outbyte
         call pause
         RET
```

NOTE

Nome: inbyte

Azione: Attende che dalla tastiera arrivino 2 (codifiche ASCII di) cifre esadecimali. Quando ciò avviene deposita nel registro AL il byte espresso in forma compatta dalle due cifre. Oltre a ciò fa l'eco sul monitor (delle codifiche ASCII) delle 2 cifre esadecimali prelevate

Nome: outbyte

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

ESERCIZIO 4 (Anni accademici precedenti al 2011-2012)

1) Data la rappresentazione $(342)_5$ in base 5, trasformarla in base 2.

2) Si mostri l'uscita a video del programma C++ seguente:

```
#include <iostream>
using namespace std;
class A{
public:
    int x;
    A(int n=0) { x=n; cout << "A:x=" << x << endl;}
    void g() { cout << "A:g() x=" << x << endl;}
    virtual ~A() { cout << "via A" << endl; }
};

class B: public A{
public:
    B(int n=7) { x++; cout << "B:x=" << x << endl; }
    virtual void g() { cout << "B:g() x=" << x << endl;}
    ~B() { cout << "via B" << endl; }
};
```

```
class C: public B{
    int x;
public:
    C(int n=1) :B(n) { x=n; cout << "C:x=" << x << endl; }
    void g() { cout << "C:g() x=" << x << endl; }
    ~C() { cout << "via C" << endl; }
};

int main(){
    C* p = new C(3);
    A* pa = p;
    B* pb = p;
    pa->g();
    pb->g();
    delete pb;
    return 0;
}
```

Soluzione:

Quando viene letto 0xA6, viene stampato a video dalla prima outbyte l'esadecimale 0x9A e successivamente l'esadecimale 0x3D.

Svolgimento:

```
inbyte 1010-0110 (A6) | AL: 1010-0110 (A6)
OR 1011-1000 (B8) | AL: 1011-1110 (BE)
AND 1001-1011 (9B) | AL: 1001-1010 (9A)
outbyte AL | -> deve stampare 9A
SHR,4 | AL: 0000-1001 (09)
JC label1
OR 1000-1010 (8A) | AL: 1000-1011 (8B)
label1: OR 0011-1101 (3D) | AL: 0011-1101 (3D)
outbyte AL -> deve stampare 3D e non 8B,
perchè la shr attiva il carry
```

A:x=0
B:x=1
C:x=3
A:g() x=1
C:g() x=3
via C
via B
via A