FONDAMENTI DI INFORM. & PROGR. A OGGETTI **SCRITTO 7 GIUGNO 2012**
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

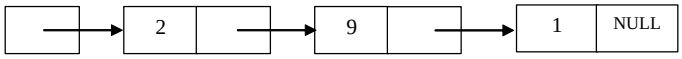
NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

- 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

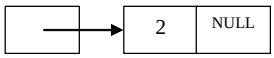
ESERCIZIO 1
Sia data la struttura seguente:

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

Scrivere una funzione che presi come argomenti una lista di elementi di tipo `elem` e due interi `i` e `j`, modifica la lista eliminando tutti gli elementi compresi fra la posizione `i` e la posizione `j` (`i` e `j` inclusi).
Assumere che il primo elemento della lista abbia posizione 1, il secondo elemento della lista abbia posizione 2 e cosi' di seguito. La funzione deve gestire le eventuali situazioni di errore.
Per esempio, se la funzione viene chiamata con i=2 e j=3, e con la lista seguente:



la lista viene modificata come segue:



ESERCIZIO 2
Sia dato il tipo seguente:

```
enum stato {L, O};
```

Scrivere una funzione che prende in ingresso una matrice di dimensione `nxm` di elementi di tipo `stato` che rappresenta lo stato dei posti di un teatro, ed un intero `k`. La funzione occupa un quadrato di dimensione `kxk` di posti liberi del teatro. Assumere k>0, k<=n e k<=m. La funzione restituisce `true` se l'operazione ha avuto successo, `false` altrimenti. Quando l'operazione fallisce, la matrice non deve essere modificata, altrimenti i posti devono essere occupati.
Per esempio, se la funzione viene chiamata con `k=3` e con la matrice seguente di dimensione 5x8:

```
L O O O O L L L
O L L L O L L O
O L L L O L O O
O L L L L L O O
O O O L L O O O
```

la matrice viene modificata come segue e la funzione restituisce `true`.

```
L O O O O L L L
O O O O O L L O
O O O O O L O O
O O O O L L O O
O O O L L O O O
```

ESERCIZIO 3
Scrivere una funzione ricorsiva che dati due vettori V1 e V2 di interi entrambi di lunghezza `n` restituisce `true` se i vettori differiscono di al più `k` elementi, `false` altrimenti. Assumere `k>0 e k<=n`. La funzione può avere argomenti aggiuntivi. Indicare esplicitamente la chiamata alla funzione.

ESERCIZIO 4 (Anno accademico 2011-2012)

1) Data la rappresentazione $(11100011)_2$ in complemento a 2, trasformarla in base 10.

2) In relazione al seguente programma, indicare le cifre esadecimali che vengono mostrate a video dalle tre chiamate alla routine 'outbyte', assumendo che venga letto il ingresso l'esadecimale corrispondente alla cifra meno significativa del proprio numero di matricola. Ad esempio, se tale cifra è 6, inserire 0x06 digitando da tastiera 0 e 6:

```
#.GLOBAL _main
# Sezione dati
.EQU mask,0x05
# Sezione codice
_main:
          CALL   inbyte
          CALL   newline
          CALL   outbyte
          CALL   newline
          MOV    (AL), (BL)
          ADD    $8, (BL)
          MOV    $mask, (CL)
loop:
          SHR    $1, (CL)
          JC     op
          CMP    $0, (CL)
          JZ     fine
          JMP    loop
op:
          SHL    $1, (BL)
          MOV    (BL), (AL)
          CALL   outbyte
          CALL   newline
          JMP    loop
fine:
          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 $(11100011)_2$ in complemento a 2, trasformarla in base 10.
2) Si mostri l'uscita a video del programma C++ seguente:

```
#include<iostream>
using namespace std;

template<class tipo>
class uno {
static int a;
public:
  int cambia(){ return  a+=10; }
};

template<class tipo>
int uno<tipo>::a = 8;

template<class T>
T fun(T k){ return k + 2;}

template<class T1, class T2>
void fun1(T1 y, T2 x){
   cout << y << endl;
   cout << x << endl;
   cout << fun(y) + fun(x) << endl;
}

int main() {
uno<int> obj1, obj2;
uno<char> obj3;

cout << obj1.cambia() << endl;
cout << obj2.cambia() << endl;
cout << obj3.cambia() << endl;
fun1<int>(1.2, 2.5);
return 0;
}
```