

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

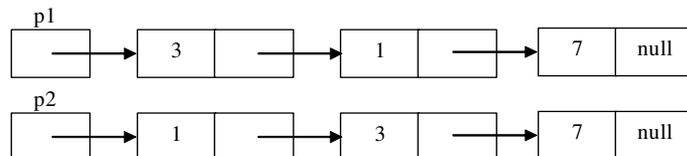
- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO CHE SI STA RISOLVENDO. PER ESEMPIO, SCRIVERE "ESERCIZIO N. 1" QUANDO SI STA RISOLVENDO L'ESERCIZIO N. 1;**
- **NON COPIARE DAL VICINO (NON È DETTO CHE IL VICINO SIA PIÙ BRAVO DI VOI);**
- **NON PERMETTETE AL VICINO DI COPIARE (È SPIACEVOLE VEDERSI ANNULLARE IL COMPITO SENZA COLPE);**
- **I PRIMI TRE ESERCIZI VALGONO 6 PUNTI. I RIMANENTI TRE VALGONO 4 PUNTI.**
- **ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.**

1) Sia data la struttura elem seguente:

```
struct elem {char inf; elem * pun;};
```

Scrivere una funzione ricorsiva che, date due liste di elementi di tipo elem passate come argomento alla funzione, restituisca true se le due liste sono uguali (cioè le liste contengono esattamente gli stessi elementi e nello stesso ordine), false altrimenti.

Per esempio, se la funzione viene chiamata con le due liste seguenti,



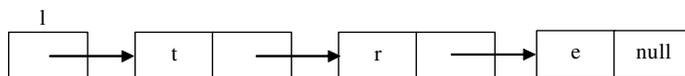
la funzione restituisce false.

2) Sia data la struttura elem seguente:

```
struct elem {char c; elem * pun;};
```

Scrivere una funzione che, data una lista di elementi di tipo elem, restituisca una stringa composta dai caratteri contenuti nel campo c degli elementi elem presi in sequenza a partire dalla testa della lista.

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



la funzione restituisce la stringa "tre".

3) Sia data la struttura complesso seguente:

```
struct complesso {double re; double im;};
```

Scrivere una funzione che, dati due vettori v1 e v2 con n > 0 elementi di tipo reale, restituisca un vettore di elementi di tipo complesso. L'elemento i-esimo del vettore restituito deve avere come parte reale re l'elemento i-esimo di v1 e come parte immaginaria im l'elemento i-esimo di v2. Ad esempio, dati i vettori v1 = [1.0 3.2] e v2 = [0.2 5.1], la funzione restituisce il vettore v3 = [(1.0,0.2) (3.2,5.1)].

4) Data la rappresentazione $(7326)_8$ in base 8, trasformarla in base 2.
Rappresentare il numero negativo -25 in complemento a due su 8 bit.

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

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

class A{
public:
    int k;
    A() { k = 5; cout << "nuovo A default " << endl; }
    A(int n) { k = n; cout << "nuovo A " << endl; }
    virtual void f() {
        k++;
        cout << "A::f(), k=" << k << endl;
    }
    ~A() { cout << "via A" << endl; }
};

class B: public A{
public:
    B(): A(4) { cout << "nuovo B default " << endl; }
    B(int n) { k = n; cout << "nuovo B" << endl; }
    virtual void f() {
        k+=10;
        cout << "B::f(), k=" << k << endl;
    }
    ~B() { cout << "via B" << endl; }
};

class C: public B{
public:
    int k;
    C() { k=8; cout << "nuovo C default " << endl; }
    C(int a) { k=a; cout << "nuovo C" << endl; }
    void f() { cout << "C::f(), k=" << k << endl; }
    ~C() { cout << "via C" << endl; }
};

int main() {
    A* v [3];
    v[0] = new C(1);
    v[1] = new B(3);
    v[2] = v[0];

    for (int i =0; i < 3; i++) v[i] ->f();

    delete v[0];
    delete v[1];

    return 0;
}
```

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

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

template<class tipo>
class A {
    int k;
    static int s;
public:
    A(int n = 10) { k=n; s++;}
    A(const A& a) { k=a.k; s++;}
    void stampa() {cout << "k=" << k << ", s=" << s << endl;}
};

template<class tipo>
int A<tipo>::s = 0;

template<class tipo>
void f(tipo obj) {
    static int w = 0;
    w++;
    cout << "w=" << w << endl;
}

int main() {
    A<int> obj1(5);
    A<char> obj2;
    A<int> obj3 = obj1;
    obj1.stampa();
    obj2.stampa();
    obj3.stampa();
    f(obj1);
    f(obj2);
    f(obj3);
    return 0;
}
```

