

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.**

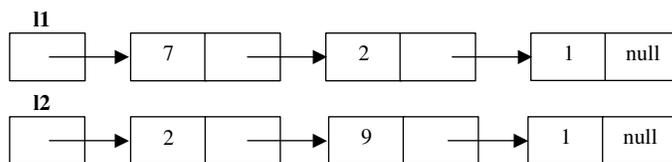
1) Scrivere una funzione ricorsiva che dato un numero naturale N dispari, stampi su uscita standard un minore (<) composto di N asterischi. Nello scrivere la funzione si supponga che N sia sempre dispari. Per esempio, dato N = 5, la funzione stampa su uscita standard

*
*
*
*
*

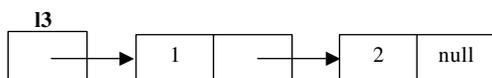
2) Sia data la struttura elem seguente:

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

Scrivere una funzione che, date due liste non ordinate l1 e l2 di elementi di tipo elem, restituisca una lista ordinata per valori crescenti del campo info contenente tutti gli elementi **comuni** alle liste l1 e l2. Per esempio, se la funzione viene chiamata con le liste l1 e l2 seguenti,



la funzione restituisce la lista l3 seguente:



3) Scrivere una funzione che, date due matrici mat1 e mat2, di r righe e c colonne, restituisca un vettore vett di r elementi. L'elemento in posizione i del vettore vett dipende dal contenuto della riga i-esima delle due matrici: l'elemento avrà valore 1 se la riga i in mat1 e la riga i in mat2 hanno almeno un elemento uguale nella stessa posizione; altrimenti avrà valore 0. Per esempio se la funzione viene

chiamata con le matrici mat1 = $\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$ e mat2 = $\begin{matrix} 2 & 2 & 3 \\ 3 & 3 & 3 \\ 4 & 8 & 9 \end{matrix}$, la funzione restituisce vett = $\begin{matrix} 1 \\ 0 \\ 1 \end{matrix}$.

4) Data la rappresentazione $(B6D)_{16}$ in base 16, trasformarla in base 8.
Rappresentare il numero negativo -24 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(int n=0) { k = n; cout << "nuovo A " << "k=" << k << endl; }
    void f() { cout << "A::f(), k=" <<k << endl; }
    ~A() { cout << "via A" << endl; }
};

class B{
    A a;
    int k;
public:
    B(int n=0) : a(4) { k = n; cout << "nuovo B " << "k=" << k << endl; }
    virtual void f() { a.f(); cout << "B::f(), k=" << k << endl; }
    virtual ~B() { cout << "via B" << endl; }
};

class C: public B{
public:
    int k;
    C(int n = 1) { k=n; cout << "nuovo C " << "k=" << k << endl; }
    void f() { cout << "C:f(), k=" << k << endl; }
    ~C() { cout << "via C" << endl; }
};

int main(){
{
    C c(2);
    B* pb = &c;
    C* pc = &c;
    pb->f();
    pc->f();
    B b;
    b=c;
    b.f();
}
return 0;
}
```

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

```
#include<iostream>
using namespace std;
template<class R>
R g (R r) {
    static R x = 0;
    x+=r;
    return x;
}

template<class T, class R>
T f (T t, R r) {
    static R x = 0;
    return t+x+g(r);
}

int main () {
    cout << f<double,int>(2.6,1) << endl;
    cout << f<int,int>(1,2) << endl;
    cout << f<double,double>(1.1,2.2) << endl;
    cout << f<int,int>(1,2) << endl;
    return 0;
}
```