

**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 stampi su video k "hip" seguiti da k "hurra" senza utilizzare istruzioni ripetitive. Per esempio, dato k = 4, la funzione stampa su uscita standard

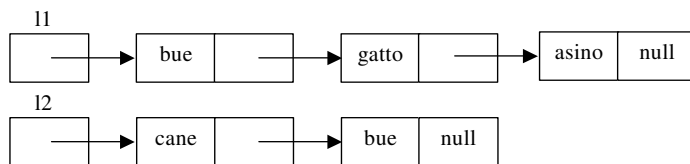
hip hip hip hip hurra hurra hurra hurra

2) Sia data la struttura elem seguente:

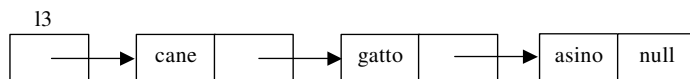
```
struct elem {char* nome; elem * pun; };
```

Scrivere una funzione che, date due liste di elementi di tipo elem, restituisca una nuova lista composta da una copia degli elementi che sono presenti soltanto in una delle due liste, ma non in entrambe.

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



la funzione restituirà la lista seguente:



L'ordine degli elementi nella lista restituita dalla funzione può essere qualunque.

3) Scrivere una funzione che, data una matrice n×m contenente lettere minuscole dell'alfabeto inglese, con n e m variabili, restituisca un vettore di 26 interi, dove l'intero in posizione 0 indica il numero di occorrenze della lettera 'a' nella matrice, l'intero in posizione 1 il numero di occorrenze della lettera 'b' nella matrice, ecc..

Per esempio, se la funzione viene chiamata con la matrice mat seguente,

$$\text{mat} = \begin{bmatrix} a & a & b & f \\ c & a & d & f \\ a & c & e & f \end{bmatrix}$$

la funzione restituisce il vettore [4 1 2 1 1 3 0]

4) Data la rappresentazione  $(234)_5$  in base 5, trasformarla in base 8.

Data la rappresentazione in complemento a due  $(10001101)_{\text{compl}2}$ , esprimere il numero in base 10.

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

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

class A{
public:
    int x;
    A(int n=3) { x = n;
                f();
                cout << "A::x=" << x << endl; }
    virtual void f() { x++; cout << "A::f() x=" << x << endl; }
    virtual ~A() { cout << "via A" << endl; }
};

class B: public A{
public:
    B(): A(4) { x = 7;
                f();
                cout << "B::x=" << x << endl; }
    void f() { x++; cout << "B::f() x=" << x << endl; }
    ~B() { cout << "via B" << endl; }
};

class C: public B{
public:
    int x;
    C(int k = 2) { x=k;
                  f();
                  cout << "C::x=" << x << endl; }
    void f() { cout << "C::f() x=" << x << endl; }
    ~C() { cout << "via C" << endl; }
};

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

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

```
#include <iostream>
using namespace std;
template<class R>
class A {
    R rr;
public:
    A(R r) { rr=r;}
    R g() {
        rr++;
        cout << rr << endl;
        return rr;
    }
};

template<class T, class R>
T f (R r) {
    static R x = 1;
    A<R> a(r);
    x+=a.g();
    return static_cast<T>(x);
}

int main () {
    cout << f<double>(2.6) << endl;
    cout << f<double,double>(1.0) << endl;
    cout << f<int,int>(4.6) << endl;
    return 0;
}
```