

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);
- TUTTI GLI ESERCIZI VALGONO 5 PUNTI
- ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.

1) Sia data la struttura seguente

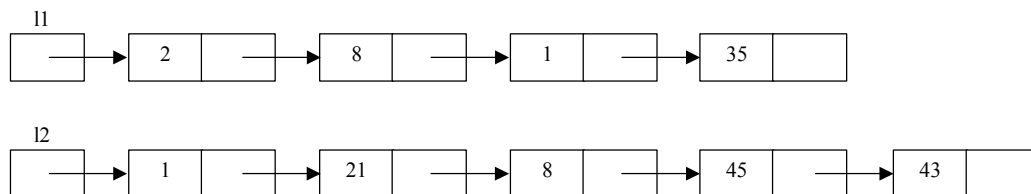
```
struct elem{int info; elem* pun;};
```

Scrivere una funzione ricorsiva che data una lista di elementi di tipo elem ed un intero x, restituisce true se x è presente nella lista; false altrimenti. La lista e l'intero x sono passati come argomento alla funzione.

2) Sia data la struttura seguente

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

Scrivere una funzione che, date due liste di elementi di tipo elem, elimina dalla prima lista tutti gli elementi della seconda. Nell'esempio seguente, siano l1 e l2 i puntatori alla testa della prima e della seconda lista, rispettivamente.



Dopo la chiamata alla funzione, la lista l1 viene modificata come segue:



- 3) Scrivere una funzione che dato un vettore di interi ordinato per valori crescenti ed un intero k, decrementa di 5 l'elemento in posizione k, mantenendo il vettore ordinato. Il vettore e l'intero k sono passati come argomento alla funzione. La funzione restituisce false se la posizione k non è valida; altrimenti restituisce true. La funzione può essere chiamata con vettori di lunghezza diversa. Se k è uguale a 0 viene decrementato il primo elemento.
- 4) Data la rappresentazione $(3EEA19)_{16}$ in base 16, trasformarla in base 2.
Data la rappresentazione in base due (100101011111110) , esprimere il numero in base 16.

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

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

class B {
protected: int b;
public:
    B(int x) { b = x; b++; ; cout << b << endl;}
    void f1() {cout << 'B' << endl;}
};

class C {
    int c;
public:
    C() { c = 10; cout << c << endl;}
};

class D: public B {
    C c;
public:
    D() : B(3) {b++; cout << b << endl;}
    virtual void f1(){ cout << 'D' << endl;}
};
```

```
class E {
    D d;
    C c;
public:
    E() {cout << 'E' << endl;}
};

int main(){
    D* pd = new D;
    B* pb = pd;
    pb->f1();
    pd->f1();
    E e;
    return 0;
}
```

6) Si mostri l'uscita a video delle esecuzioni del programma C++ seguente con rispettivamente gli ingressi c = 'a', c = 'b' e c = 'c'.

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

class A{
public:
    char c;
    A() {c = '?';}
};

class AA: public A {
public:
    AA(char a) { c = a; }
};

void f1(char c) {
    try {
        if( c=='a' )
            throw AA(c);
    }
    catch(A& obj) {
        cout << "Ecc. da f1: " << obj.c << endl;
    }
}
```

```
char f(char c) {
    try {
        f1(c);
        if (c == 'b')
            throw AA(c);
        else
            throw A();
        c = 'c';
    }
    catch(AA& obj) {
        cout << "Ecc. da f: " << obj.c << endl;
    }
    return c;
}

int main(){
    char c;
    cin >> c;
    try { cout << f(c) << endl; }
    catch (A& obj) {
        cout << "Ecc. da main: " << obj.c << endl; }
    return 0;
}
```