

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 info; elem * pun;};
```

Scrivere una funzione ricorsiva che, data una lista di elementi di tipo elem passata come argomento alla funzione, restituisce true se i campi info di tutti gli elementi sono lettere maiuscole dell'alfabeto oppure la lista è vuota, false altrimenti. Per esempio, se la funzione viene chiamata con la lista p1 restituirà false, se invece viene chiamata con la lista p2 restituirà true.

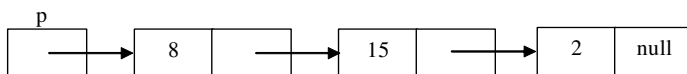


2) Sia data la struttura elem seguente:

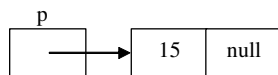
```
struct elem {int info; elem * pun;};
```

Scrivere una funzione che, data una lista di elementi di tipo elem ed un intero n, elimina dalla lista tutti gli elementi con campo informazione minore o uguale a n.

Per esempio, se la funzione viene chiamata con la lista seguente e l'intero n=10,



la funzione modificherà la lista come segue:



3) Scrivere una funzione che, data una matrice quadrata $n \times n$ di interi ed un intero val, restituisca il numero totale di righe e di colonne la cui somma degli elementi è uguale a val. La funzione deve poter essere chiamata con matrici di dimensione $n \times n$ qualsiasi.

Per esempio, se la funzione viene chiamata con la matrice mat seguente e l'intero val = 3, la funzione restituirà 3.

$$\text{mat} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- 4) Data la rappresentazione $(126)_9$, in base 9, trasformarla in base 7.
 Data la rappresentazione in complemento a due $(10111100)_{\text{compl2}}$, 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=1) { x = n; cout << "nuovo A\n" << "x=" << x << endl; }
    void f() { cout << "A:f()\n" << "x=" << x << endl; }
    ~A() { cout << "via A" << endl; }
};

class B: public A{
public:
    int x;
    B(): A(5) { x = 7; cout << "nuovo B\n" << "x=" << x << endl; }
    virtual void f() { cout << "B:f()\n" << "x=" << x << endl; }
    ~B() { cout << "via B" << endl; }
};

class C: public B{
    A obj;
public:
    C(int k = 2) { x=k; cout << "nuovo C\n" << "x=" << x << endl; }
    void f() { cout << "C:f()\n" << "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 pb;
    return 0;
}
```

- 6) Si mostri l'uscita a video del programma C++ seguente con input -1, 0 e 1:

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

int g (int n){
    cout << n << endl;
    try {
        if (n==1) throw 1;
        if (n==0) throw 'c';
        cout << n << endl;
    }
    catch(char) { cout << "ecc. char da g" << endl;}
    cout << "fine g" << endl;
    return n+10;
}

int f(int y) { return g(y+1); }
```

```
int main(){
    int x;
    cin >> x;
    try {
        cout << x << endl;
        if (x==1) throw 'c';
        cout << f(x) << endl;
    }
    catch(char) {cout << "ecc. char da main" << endl;}
    catch(int) { cout << "int da main" << endl;}
    catch(...) { cout << "default da main" << endl;}
    cout << "fine main" << endl;
    return 0;
}
```