

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

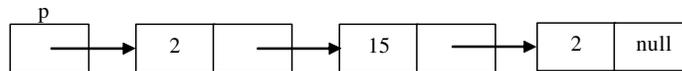
- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO 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 vettore `vett` di elementi di tipo intero passato come argomento alla funzione, ordina il vettore in ordine crescente.

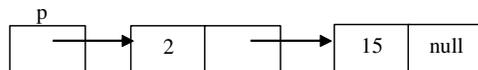
2) `struct elem {int n; elem * pun;};`

Scrivere una funzione che, data una lista di elementi di tipo `elem`, elimina dalla lista tutti i duplicati.

Se chiamata con argomento la lista puntata da `p`,



la funzione modificherà la lista come segue:



3) Scrivere una funzione che, dati due vettori `v1` e `v2` di elementi di tipo intero ordinati in ordine crescente, restituisca un vettore con tutti gli elementi di `v1` e tutti gli elementi di `v2`, ordinato in ordine decrescente. I vettori `v1` e `v2` possono contenere duplicati e possono avere dimensione diversa.

4) Data la rappresentazione $(126)_8$ in base 8, trasformarla in base 16.

Indicare la rappresentazione in complemento a 2 su 8 bit del numero intero $(-34)_{10}$ 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\t" << "x=" << x << endl; }
void f() { cout << "A:f()\t" << "x=" << x << endl; }
};

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

class C: public B{
public:
C(int k = 2) { x=k; cout << "nuovo C\t" << "x=" << x << endl; }
void f() { cout << "C:f()\t" << "x=" << x << endl; }
};
```

```
int main(){
    C obj(10) ;
    B obj1(3) ;
    obj1 =obj ;
    B* pb = &obj ;
    A* pa = pb;

    obj.f() ;
    obj1.f() ;
    pa->f();
    pb->f();
    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){
    try {
        if (n==1) throw 1;
        if (n==0) throw 'c';
        cout << n << endl;
    }
    catch(int) { cout << "ecc. int da g" << endl;}

    cout << "fine g" << endl;
    return 10;
};

int f(int y) {
    try {
        cout << g(y+1) << endl;
    }
    catch(char) { cout << "ecc. char da f" << endl;}

    cout << "fine f" << endl;
    return 7;
};
```

```
int main(){

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

    cout << "fine main" << endl;
    return 0;
}
```