FONDAMENTI DI INFORMATICA E PROGRAMMAZIONE A OGGETTI **16 SETTEMBRE 2009**
FONDAMENTI DI INFORMATICA I

NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

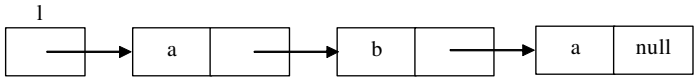
- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO 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, verifica se il vettore è simmetrico. La funzione restituisce true se il vettore è simmetrico; altrimenti restituisce false.
   Per esempio, la funzione restituisce true sia per il vettore [1 2 3 3 2 1] che per il vettore [1 2 3 2 1], mentre restituisce false per il vettore [1 2 3 4 2 1].

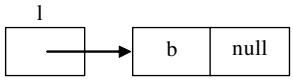
2) Sia data la struttura elem seguente:

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

   Scrivere una funzione che, data una stringa s ed una lista l di elementi di tipo elem, elimini dalla lista l tutti gli elementi il cui campo let coincide con un carattere della stringa s.
   Per esempio, se la funzione viene chiamata con la stringa "cane" e la lista seguente



   la funzione modifica la lista nel modo seguente:



3) Scrivere una funzione che, data una matrice n×m di interi, con n e m variabili, restituisca true se la matrice contiene almeno due interi uguali; false altrimenti.
   Per esempio, data la matrice seguente

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 2 & 9 & 0 & 10 \end{bmatrix}$$

   la funzione restituisce true.

4) Data la rappresentazione $(345)_8$ in base 8, trasformarla in base 16.
   Rappresentare il numero intero $(-48)_{10}$ in complemento a 2 su 8 bit.

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;
                            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;
                            cout << "B::x="<< x << endl; }
        void f() { x++; cout << "B::f() x="<< x << endl; }
        ~B() {  cout << "via B" << endl; }
};

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

```
int main()
{
   B* pb = new C;
   A* pa = pb;
   pa->f();
   pb->f();
   C c(5);
   c.f();
   A a = c;
   a.f();
   delete pa;
   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) {
   cout << g(y+1) << endl;
   cout << "fine f" << endl;
        return 7;
}
```

```
int main(){
    int x;
    cin >> x;
    try { 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;
}
```