

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; GLI ULTIMI 3 VALGONO 4 PUNTI.**
- **ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.**

1) Scrivere una funzione ricorsiva che, data una stringa, restituisca il numero di caratteri della stringa compresi tra il carattere 'a' ed il carattere 'd' ('a' e 'd' inclusi).

Per esempio, se la stringa è "Fondamenti di Informatica", il numero restituito dalla funzione è 6.

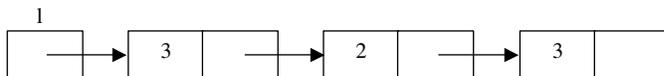
2) Sia data la struttura seguente

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

Scrivere una funzione che, data una lista di elementi di tipo elem, se due elementi consecutivi hanno lo stesso campo info, elimina uno dei due elementi. Per esempio, la lista seguente:



viene modificata dalla funzione come



3) Scrivere una funzione che, data una matrice quadrata di interi di dimensione qualsiasi, restituisca la somma di tutti gli elementi della matrice esclusi gli elementi sulla diagonale principale e sulla diagonale secondaria.

Per esempio, se la matrice è la seguente (la diagonale principale e la diagonale secondaria sono indicate con le linee tratteggiate):

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

la funzione restituisce 5.

4) Data la rappresentazione $(314)_5$ in base 5, trasformarla in base 7.

Data la rappresentazione $(-15)_{10}$ in base 10, trasformarla in complemento a 2 su 8 bit.

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

<pre>#include <iostream> using namespace std; class A{ public: int x; A() { x=10; cout << "A:x=" << x << endl;} void f() { cout << "A:f() x=" << x << endl;} virtual ~A() { cout << "via A" << endl; } }; class B: public A{ public: B(int n=7) { x=n; cout << "B:x=" << x << endl; } virtual void f() { cout << "B:f() x=" << x << endl; } ~B() { cout << "via B" << endl; } }; class C: public B{ int x; public: C() { x=2; cout << "C:x=" << x << endl; } void f() { cout << "C:f() x=" << x << endl; } ~C() { cout << "via C" << endl; } };</pre>	<pre>class D: public B { int x; public: D():B(3) { x=20; cout << "D:x=" << x << endl; } ~D() { cout << "via D" << endl; } }; int main(){ C* pc = new C; B* pb = pc; A* pa = pc; pa->f(); pb->f(); pc->f(); D* pd = new D; pd->f(); delete pa; delete pd; return 0; }</pre>
--	---

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

<pre>#include<iostream> using namespace std; class ecc { public: ecc() { cout << "nuova eccezione" << endl;} void virtuale() { cout << "tipo 0" << endl; } }; class ecc1: public ecc { public: ecc1() { cout << "ecc1" << endl; } void virtuale() { cout << "tipo 1" << endl; } }; class ecc2: public ecc { public: ecc2() { cout << "ecc2" << endl;} void virtuale() { cout << "tipo 2" << endl;} };</pre>	<pre>int f(int x, int y) { if (x==y) throw ecc1(); if (x < y) throw ecc2(); if (y==0) throw 0; return x-y; } int main(){ try{ cout << f(4,4) << endl; } catch (ecc &e){ e.virtuale(); } catch(int) {cout << "tipo int"; } try{ cout << f(3,4) << endl; } catch (ecc &e){ e.virtuale(); } catch(int) {cout << "tipo int"; } try{ cout << f(4,0) << endl; } catch (ecc &e){ e.virtuale(); } catch(int) {cout << "tipo int" << endl; } return 0; }</pre>
---	---