NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

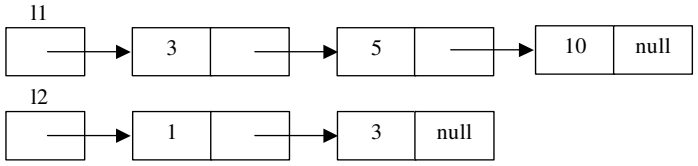
- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO 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) Una parola palindroma è una parola che, letta a rovescio, rimane identica. Le parole "esose", "ereggere", "ingegni" sono esempi di parole palindrome. Scrivere una funzione ricorsiva che, data una qualsiasi parola, restituisca true se la parola è palindroma; false altrimenti.

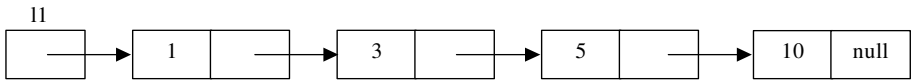
2) Sia data la struttura elem seguente:

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

Scrivere una funzione che, date due liste l1 e l2, di elementi di tipo elem, ordinate per valori crescenti del campo num e prive di elementi duplicati, aggiunga ad l1 gli elementi di l2 in modo che gli elementi di l1 risultino ordinati per valori crescenti del campo num e non vi siano elementi duplicati. Per esempio, se la funzione viene chiamata con le due liste seguenti,



la funzione modifica l1 nel modo seguente:



3) Sia data una matrice coor di dimensioni n×2, dove ogni riga della matrice contiene gli indici di riga e di colonna di una matrice mat di dimensioni q×q. Scrivere una funzione che, passati come argomenti la matrice coor e le dimensioni n e q, restituisca una matrice mat di dimensioni q×q composta da tutti 0 eccetto per gli elementi individuati dagli indici contenuti nelle righe della matrice coor. Questi elementi sono posti ad 1. Per esempio, se la funzione viene chiamata con q=3 e la matrice coor seguente,

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

la funzione restituisce la matrice mat seguente:

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

4) Data la rappresentazione $(323)_6$ in base 6, trasformarla in base 4.
Data la rappresentazione in complemento a due $(10100100)_{compl2}$, esprimere il numero in base 10.

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

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

class D{
public:
     D(){cout << "D" << endl;}
     ~D(){cout << "Via D" << endl;}
};

class A{
protected: int x;
public:
          A(int n=3) { x = n;  cout << "A::x="<< x << endl; }
          void f() {cout << "A::f() x=" << x << endl; }
          virtual ~A() {  cout << "via A" << endl; }
};

class B: public A{
    D d;
public:
          B(int k) { x = k-1; cout << "B::x="<< x << endl; }
          virtual void f() { x++; cout << "B::f() x="<< x << endl; }
          ~B() {  cout << "via B" << endl; }
};
```

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


int main()
{
   C* pc = new C(4);
   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 -2, -1 e 0:

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

int g (int n){
     cout << n << endl;
     try {
          if (n==1) throw 1;
          if (n==0) throw 'a';
          if (n==-1) throw 'b';
          }
     catch(char c) { if (c=='a') throw;
               else 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;
          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;
}
```