NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

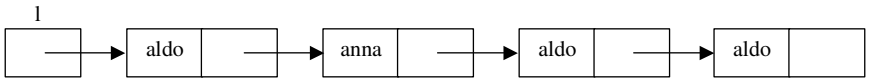
- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO 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.**
- **SI RICORDA CHE LO SCRITTO VALE PER TRE APPELLI CONSECUTIVI (COMPRESO L'APPELLO IN CUI LO SCRITTO È SOSTENUTO)**

1) Scrivere una funzione ricorsiva che, dato un numero naturale N espresso in base 10 ed una base B<10, stampi su uscita standard la codifica in base B di N. Per esempio, dato il numero N = 14 e la base B = 5 la funzione stampa 24.

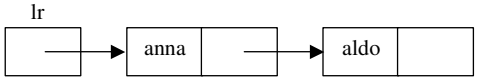
2) Sia data la struttura seguente:

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

Scrivere una funzione che, data una lista l (non ordinata) di elementi di tipo elem, restituisce una lista lr di elementi di tipo elem che contiene gli stessi campi nome di l ma ripetuti una sola volta. Ad esempio, data la lista l seguente:



la lista lr restituita dalla funzione è, per esempio,:



3) Scrivere una funzione che, dato un vettore di N stringhe, dove N è variabile, ordinato per valori alfabeticamente crescenti, applichi l'algoritmo di ricerca binaria per trovare una stringa passata come argomento alla funzione. Se la stringa è presente, la funzione restituisce la posizione della stringa nel vettore; altrimenti restituisce –1. Per esempio, dato il vettore m = $\begin{bmatrix} amico \\ casa \\ piazza \\ ponte \end{bmatrix}$ e la stringa "casa", la funzione restituisce 1.

4) Data la rappresentazione $(10100010)_2$ in base 2, trasformarla in base 8.
Data l'intero negativo –34 rappresentarlo in complemento a due su 8 bit.

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

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

class A
{ protected:
      int x;
  public:
      A(int n=0) { x=n; cout << "A : x = " << x << endl;}
      void f() { cout << "A::f() x = " << x << endl;}
};

class B: public A
{
    int x;
    A a;
  public:
      B(int n=4): A(n-2) { x=n; cout << "B: x = " << x << endl; }
      virtual void f() { cout << "B::f() x = " << x << endl;}
};

class C: public B
{
public:

     C(int n=0) : B(n) { A::x++;  cout << "C: x = " << A::x << endl; }
     void f() { cout << "C::f() x = " << A::x << endl;}
} ;
```

```
int main(){
    A* pa = new C(3);
    pa->f();
    B* pb = new C;
    pb->f();
    return 0;
}
```

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

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

template<class R>
class A {
  public:
        R g(R i)
        {
            i++;
            cout << i << endl;
            return i;
        }
};

template<class T, class R, class S>
T f (R r, S s) {
    static R x = 1;
    A<R> a;
    x+=a.g(r)+s;
    return static_cast<T>(x);
}
```

```
int main () {
        cout << f<double>(2.6, 2.4) << endl;
        cout << f<int,double,double>(1.0, 4) << endl;
        cout << f<int,int>(4.6, 4) << endl;
        return 0;
}
```