



UNIVERSITÀ DEGLI STUDI DI PISA  
FACOLTÀ DI INGEGNERIA  
CORSO di LAUREA in INGEGNERIA delle TELECOMUNICAZIONI

**ESAME DI FONDAMENTI DI INFORMATICA - SIMONCINI – 21/09/2006 - C++**

1. Siano dati due file `F1.dat` e `F2.dat` ciascuno dei quali è composto da una sequenza di numeri interi alternativamente pari e dispari. La lunghezza di `F1.dat` può essere diversa da quella di `F2.dat`. Si scriva una funzione `raggruppa()` che costruisce un nuovo file `F3.dat` che contiene tutti i valori pari contenuti in `F1.dat` e `F2.dat` e, a seguire, tutti i valori dispari.
  
2. Dati due array `B1` e `B2` di dimensione `N` a valori booleani, si scriva una funzione booleana `unione(B1, B2)` che riceve in ingresso i due array e costruisce un nuovo array `B3` di dimensione `N` il cui generico elemento `B3[i]` è dato dal risultato dell'operazione logica `B1[i] or B2[i]`. La funzione restituisce `true` se `B3` risulta composto da tutti valori uguali a `true`,  
`false` altrimenti.
  
3. Sia data una lista `l` semplice a valori interi non vuota composta da elementi diversi fra loro. Si scriva una funzione `elimina_testa(l, n)` che riceve in ingresso la lista `l` e un valore intero `n`. La funzione cerca in `l` un elemento con campo informativo uguale a `n` e, se esiste, elimina dalla lista tutti gli elementi che lo precedono.
  
4. Dati i numeri decimali `A=115`, `B=42` e `C=-87`, si determini il minimo numero di bit necessari per rappresentare contemporaneamente i suddetti numeri in complemento a 2 e se ne dia la rappresentazione. Utilizzando poi lo stesso numero di bit, si esegua l'operazione `D=B-C` e si discuta se il risultato ottenuto è o no significativo.

Punteggio:

Es.1	Es.2	Es.3	Es.4
10	10	10	3



UNIVERSITÀ DEGLI STUDI DI PISA  
FACOLTÀ DI INGEGNERIA  
CORSO di LAUREA in INGEGNERIA ELETTRONICA

1.

```
#include <fstream.h>

void raggruppa() {
    fstream f1, f2, f3;
    f1.open("F1.dat", ios::in);
    f2.open("F2.dat", ios::in);
    f3.open("F3.dat", ios::out);
    int n1, n2, n3;

    while (f1>>n1) {
        f3<<n1<<"\n";
        f1>>n1; }
    while (f2>>n2) {
        f3<<n2<<"\n";
        f2>>n2; }

    f1.close(); f2.close();
    f1.open("F1.dat", ios::in);
    f2.open("F2.dat", ios::in);

    f1>>n1;
    while (f1>>n1) {
        f3<<n1<<"\n";
        f1>>n1; }
    f2>>n2;
    while (f2>>n2) {
        f3<<n2<<"\n";
        f2>>n2; }

    f1.close(); f2.close(); f3.close();
}
```

2.

```
const int N=32;

boolean unione(boolean B1[N], Boolean B2[N]) {
    {   boolean B3[N];
        boolean risultato=true;
        for(int i=0; i<N; i++)
            if (B1[i] || B2[i]) B3[i]=1;
            else {   risultato=false;
                    B3[i]=0; }
        return risultato; }
}
```

3.

```
void elimina_testa(int n, elem *&l)
{
    for(aux=l->next; aux!=0; aux=aux->next)
        if(aux->info==n) break;
    if(aux==0) return;
    aux=l;
    while(aux->info!=n)
    {   l=l->next;
        delete aux;
        aux=l; }
}
```