UNIVERSITÀ DEGLI STUDI DI PISA

FACOLTÀ DI INGEGNERIA

CORSO di LAUREA in INGEGNERIA delle TELECOMUNICAZIONI

ESAME DI FONDAMENTI DI INFORMATICA 29/01/2010

1. **Sia dato un file** `numeri.dat` **contenente un elenco non vuoto di numeri interi. Si scriva una funzione** `crea_file()` **che legge l'ultimo valore del file** `numeri.dat` **e costruisce un nuovo file** `solo_uguali.dat` **contenente i soli elementi di** `numeri.dat` **uguali a tale numero.**

2. **Sia data una lista semplice** `l` **i cui elementi hanno due campi informativi di tipo intero. Si scriva una funzione** `modifica(l)` **che elimina dalla lista** `l` **quegli elementi che hanno i campi informativi la cui somma è nulla.**

3. **Dati due array** `B1` **e** `B2` **di dimensione** `N` **a valori booleani, si scriva una funzione booleana** `intersezione(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] AND B2[i]`**. La funzione restituisce** `true` **se** `B3` **risulta composto da tutti valori uguali a** `false`**, altrimenti restituisce** `false`**.**

4. **Dati i numeri decimali** `A=116`**,** `B=-128` **e** `C=-143`**, 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=A-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

Risultati e Soluzione del compito: http://www.isti.cnr.it/~glami/

UNIVERSITÀ DEGLI STUDI DI PISA

FACOLTÀ DI INGEGNERIA

CORSO di LAUREA in INGEGNERIA ELETTRONICA

1.

```
#include <fstream.h>

void crea_file()
{
fstream leggi, scrivi;
int last, x;
leggi.open("numeri.dat", ios::in);

while(leggi>>last)
        ;

leggi.close();
scrivi.open("solo_uguali.dat", ios::out);
leggi.open("numeri.dat", ios::in);

while((leggi>>x)
        if(x==last)
                scrivi<<x<<" ";

scrivi.close(),
leggi.close(); }
```

2.

```
struct elem { int info1;
        int info2;
        elem *next; };

void elimina(elem *&l)
{
elem *aux1, *aux2;
if(l==0) return;
aux1=l;

//  eliminazione dalla testa della lista
while(l!=0 && l->info1 + l->info2==0)
{   l=l->next;
    delete aux1;
    aux1=l;
}
if(l==0) return;
aux2=l->next;
while(aux2!=0)
{   if(aux2->info1+ aux2->info2==0)
    {       aux1->next=aux2->next;
            delete aux2;
            aux2=aux1->next;
    }
    else
    {       aux1=aux1->next;
            aux2=aux2->next;
    }
} }
```

3.

```
const int N=32;
boolean intersezione (boolean B1[N], Boolean B2[N])
        { boolean B3[N];
        boolean risultato=true;
        for(int i=0; i<N; i++)
        if (B1[i] && B2[i]) {       risultato=false;
                                    B3[i]=1; }
        else B3[i]=0;

        return risultato; }
```