



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

ESAME DI FONDAMENTI DI INFORMATICA 18/02/2010

1. Sia dato un file `numeri.dat` contenente su ciascuna riga due valori interi. Si scriva una funzione `int conta_nulle()` che crea un file `righe_nulle.dat` contenente tutte la righe di `numeri.dat` tali che la somma dei due valori che le compongono è nulla. La funzione restituisce infine il numero di righe presenti nel file `righe_nulle.dat`.
2. Sia data una lista circolare `l` i cui elementi hanno un campo informativo di tipo intero maggiore di 0. Si scriva una funzione `elimina(l)` che legge il campo informativo `n` dell'elemento puntato da `l` e dealloca l'elemento di `l` che si incontra dopo aver scorso `l` di `n` passi.
3. Si scriva una funzione `bool verifica(M)` che riceve in ingresso una matrice `M` quadrata di dimensione `NxN` a valori interi e restituisce `true` se `M` ha la struttura rappresentata in figura:

1	0	0	..		
0	1	0	..		
..	0	1	0	..	
			
		..	0	1	0
		..	0	1	

4. Dati i numeri decimali $A=-122$, $B=126$ e $C=-132$, 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	11	9	3



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

1. #include <fstream.h>

```
int conta_nulle()      {
    fstream leggi, scrivi;
    int x1, x2;
    leggi.open("numeri.dat", ios::in);
    scrivi.open("righe_nulle.dat", ios::out);
    int conta=0;

    while(leggi>>x1>>x2)
    {
        If (x1+x2==0) { conta++;
                      Scrivi << x1<<" "<<x2<<"\n"; }
    }
    leggi.close();
    scrivi.close();
    return conta;
}
```

2.

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

void elimina(elem *&l)
{
    elem *aux1, *aux2;
    if(l==0) return;
    if(l->next==l) {delete l; l=0; return;}

    aux1=l; aux2=l->next;
    int n=l->info;
    for(int i=1; i<n; i++) {aux1=aux1->next; aux2=aux2->next;}

    //eliminazione della testa
    if(aux2==l) {aux1->next=aux2->next; l=l->next; delete aux2; return;}

    //caso generale
    aux1->next=aux2->next;
    delete aux2;
    return;}

```

3.

```
const int N=12;
boolean verifica (int M[N][N])
{
    for(int i=0; i<N; i++)
        for(int j=0; j<N; j++)
            {
                If(i==j && M[i][j]!=1) return false;
                If(i!=j && M[i][j]!=0) return false;
            }
    return true;}

```