

## Prova scritta 2 luglio 2015

### Esercizio 1

```
void elimina (elem*& testa) {
    elem*p, *q;
    q = testa;
    while (q!=NULL) {
        if (q->info%2) {
            if (q==testa) {
                testa=testa->pun;
                delete q;
                q=testa;
            }
            else {
                p->pun =q->pun;
                delete q;
                q=p->pun;
            }
        }
        else {
            p=q;
            q=q->pun;
        }
    } // fine while
}
```

### Esercizio 2

```
int* creavettore( int* v1, int l1, int* v2, int l2) {
    int* v = new int [l1+l2];
    int i, j, k;
    i=0;
    j=0;
    k= l1+l2 -1;
    while(i < l1 && j < l2)
        v[k--]=(v[i]<v[j])?v[i++]: v[j++];
    if (i==l1)
        while (j<l2)
            v[k--]=v[j++];
    else
        while (i<l1)
            v[k--]=v[i++];

    ofstream os("output.txt");
    if (!os)
        cout << "errore nell'apertura del file \n";

    for (i=0; i< l1+l2; i++)
        os << v[i] << ' ';
    return v;
}
```

### Esercizio 3

```
int conta(int* mat, int n, int k=0) {
    if (k==n*n)
        return 0;
    return mat[k] + conta(mat, n, k+1);
}
```

### Esercizio 4

#### Soluzione 4.1

---

0.75 è uguale a  $0.5 + 0.25$ , dunque la sua rappresentazione su  $p=6$  bit di cui  $f=5$  per la parte frazionaria sarà 0.11000.

Riguardo al numero -6.75: essendo negativo  $\Rightarrow s = 1$   
Passiamo a rappresentare la parte intera 6, vedendola come naturale 6: 110  
La rappresentazione della parte frazionaria (0.75) è già stata calcolata.

A questo punto giustapponendo la rappresentazione della parte intera e quella della parte frazionaria otteniamo: 110.11  
la cui rappresentazione normalizzata è 1.1011 per due elevato alla +2.  
Dunque F sono i bit dopo la virgola, con l'aggiunta di 6 zeri:  
F = 1011000000

Per trovare la rappresentazione di E basta trovare la rappresentazione del numero +2 in una rappresentazione con BIAS su 5 bit. Poichè il bias è 01111, basta sommare due: 10001.  
E = 10001

In definitiva:  $s=1$ , F = 1011000000, E = 10001

#### Soluzione 4.2

---

Il programma calcola la distanza tra la rappresentazione ASCII di 'a' (97 in decimale) e quella del carattere 'A' (65 in decimale) che vale 32 (in base 10) e 20 in base 16.  
Pertanto la prima outbyte stamperà a video "20".

Il programma continua prelevando l'indirizzo di memoria in cui inizia la stringa ASCII str, che viene memorizzato in EBX.  
La prima istruzione output mostrerà a video il primo carattere di str, ossia 'T'.  
Il programma prosegue caricando le codifiche ASCII dei due caratteri successivi ('r' e 'e'), che vengono mostrate a video dopo essere state incrementate di 0x20.  
Pertanto il programma stamperà a video 'R' e 'E'.

Riassumendo il programma stampa a video la seguente stringa: "20TRE"