

Esercizio 1

```
void aggiorna(elem*& L, const char *s) {
    ifstream in(s);
    if(!in)
        return;
    int numero;
    in >> numero;
    if(!in)
        return;
    elem *p, *prec;
    while(L!=NULL)&&(L->info <=numero) {
        p=L;
        L= L->pun;
        delete p;
    }
    if (L==NULL)
        return;
    prec = L;
    p = L->pun;
    while(p!=NULL) {
        if (p->info <= numero) {
            prec->pun = p->pun;
            delete p;
            p = prec->pun;
        }
    }
    return;
}
```

Esercizio 2

```
int* creamatrice(int* mat, int r, int c) {
    int* p = new int [r*c];
    for (int i=0 ; i<r; i++) // colonne matrice nuova
        for (int j=0; j<c; j++) // righe matrice nuova
            p[j*r+i] = mat[i*c+j];
    ofstream ff("output.txt", ios::out);
    if (!ff)
        cerr << "Il file non puo' essere aperto" << endl;
    else {
        for (int h=0 ; h<c; h++){
            for (int k=0; k<r; k++)
                ff << p[h*r+k] << ' ';
            ff << endl;
        }
    }
    return p;
}
```

Esercizio 3

```
int controlla(const char* s, char c) {
    if ((*s) == '\0')
        return 0;
    if ((*s) == c)
        return 1 + controlla(s+1, c);
    return controlla(s+1, c);
}
```

Esercizio 4

4.1

La parte intera e' 1, che ha rappresentazione 1 in base due.

La parte frazionaria 0.125 ha rappresentazione $f = 0.001$

Dunque la rappresentazione in virgola fissa e': 1.001

Tenendo conto dell'uno implicito nella rappresentazione in virgola mobile, e' evidente che la parte frazionaria e': 0010000000 e l'esponente deve essere zero ($e=0$).

Pertanto $E = e + \text{bias} = 01111$

In definitiva, tenendo conto che il numero era negativo: $\{s=1, E=01111, F=0010000000\}$

4.2

Questo programma effettua il test di validita' della Legge di De Morgan fra BL e CL:

$\text{NOT}(\text{BL AND CL}) \Leftrightarrow \text{NOT}(\text{BL}) \text{ OR } \text{NOT}(\text{CL})$

Ad esempio, se il proprio numero di matricola termina per "35", in BL verra' caricato il codice ASCII di '3' (ossia 0x33) ed in CL quello di '5' (ossia 0x35).

Il loro AND logico dara':

0011-0011 (0x33) AND

0011-0101 (0x35) =

0011-0001 (0x31)

La cui negazione logica e':

1100-1110 (0xCE)

Dunque a video viene stampato "CE" tramite la prima outbyte.

Nella seconda parte si nega prima BL e CL:

$\text{NOT}(\text{BL}) = 1100-1100$ (0xCC)

$\text{NOT}(\text{CL}) = 1100-1010$ (0xCA)

e di essi viene fatto l'OR:

1100-1100 (0xCC) OR

1100-1010 (0xCA) =

1100-1110 (0xCE)

e dunque verra' stampato ancora una volta "CE" dalla seconda outbyte.

NB: Qualunque sia la coppia di cifre finali del proprio numero di matricola, le uscite delle due outbyte devono coincidere, in virtu' della Legge di De Morgan.