

Esercizio 1

```

void elimina(elem*& p0, int n, int m) {
if (p0==NULL) return;
int min = (n<=m)?n:m;
int max = (n>=m)?n:m;

elem*p=p0;
elem*q;
while(p!=NULL){
if ((p->info>=min) && (p->info <=max)) {
    if(p==p0){
        p0=p->pun;
        delete p;
        p=p0;
    }
    else {
        q->pun = p->pun;
        delete p;
        p=q->pun;
    }
} // fine cancellazione
else {
    q=p;
    p = p->pun;
}
}
}

```

Esercizio 2

```

int* build(){
    int* m=new int[9];
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            if (j == (2-i))
                m[i*3+j]=1;
            else
                m[i*3+j]=0;
    return m;
}

int* leggiMatrice(const char* s){
    ifstream in(s);
    if(!in)
        return build();
    int dim, n;
    in >> dim;
    if((!in) || dim <=0)
        return build();

    int* mat=new int [dim*dim];
    for (inti=0; i< dim*dim; i++) {
        in >> n;
        if(!in) {
            delete[]mat;
            return build();
        }
        else {
            mat[i] = n;
        }
    }
}

```

```

char c;
in >>c;
if(!in || (c != '\!')) {
    delete[]mat;
    return build();
}
return mat;
}

```

Esercizio 3

```

int conta(elem* L) {
    if (L==NULL) return 0;
    if ((L->info >= 'A') && (L->info <= 'Z'))
        return 1 + conta(L->pun);
    return  conta(L->pun);
}

```

Esercizio 4

4.1)

a) La rappresentazione e' 9E. Si ottiene facilmente raggruppando le cifre in gruppi di 5: 01001-01110 e trovando la cifra associata a ciascun gruppo di cinque nella base 32. La controprova e' la seguente: 0100101110 corrisponde a $256+32+8+4+2 = 302$ che e' anche uguale a $9*32+14$

b) Si, perchè nella half precision per la parte frazionaria della mantissa F viene utilizzato esattamente un naturale su 10 bit (nello specifico la mantissa m varrebbe 1.0100101110)

4.2)

All'inizio il programma stampa a video i caratteri "=>". Successivamente ripete un ciclo per 8 volte. All'interno di ogni ciclo, legge il bit piu' a sinistra del byte corrispondente alle proprie due cifre meno significative. Se tale bit vale 1, l'istruzione ROL setta il carry flag e si passa a stampare a video un uno, altrimenti, si stampa uno zero. Poiche' il contenuto di BL e' stato ruotato a sinistra, nel ciclo successivo si passa ad analizzare la seconda cifra del byte iniziale. Questo e' ripetuto fino a quando non si e' stampata la quarta cifra: in tal caso viene stampato il carattere di separazione '-', per poi riprendere con la stampa dei rimanenti bit meno significativi.

Ecco alcuni esempi di cosa stampa il programma.

Cifre meno significative: 28
 stampa "=>0010-1000"

Cifre meno significative: 97
 stampa "=>1001-0111"

e cosi' via.