

Soluzione prova scritta 27 giugno 2012
Esercizio 1

```
void elimina(elem*& testa) {
    if (testa==NULL) return;
    int L = strlen(testa->nome);
    elem*p, *q;
    for( q=testa; q!= NULL; q=q->pun)
        if (strlen(q->nome) < L)
            L= strlen(q->nome);
    q= testa;
    while (q!= NULL) {
        if (strlen(q->nome) == L) { //elemento deve essere eliminato
            if (q==testa) {
                testa=testa->pun;
                delete[]q->nome;
                delete q;
                q=testa;
            }
            else {
                p->pun = q->pun;
                delete[]q->nome;
                delete q;
                q=p->pun;
            }
        }
        else { // elemento non deve essere eliminato
            p=q;
            q=q->pun;
        }
    }
}
```

Esercizio 3

Esempio di chiamata:

controlla_se_palindroma("1221", 4, 0);

```
bool controlla_se_palindroma (const char* s, int n, int k=0){
    int medio = n/2;
    if (k >= medio)
        return true;
    if (s[k] != s[n-1-k])
        return false;
    return controlla_se_palindroma (s, n, k+1);
}
```

Esercizio 2

```
bool riga_ordinata(int* mat, int dim, int i) {
    bool ord_cresc= true, ord_decresc= true;
    for (int j=0; j<dim-1; j++) {
        if (! (mat[i*dim +j] < mat[i*dim +j +1]))
            ord_cresc = false;
        if (! (mat[i*dim +j] > mat[i*dim +j +1]))
            ord_decresc = false;
    }
    return (ord_cresc || ord_decresc);
}
```

```
bool controlla_ordinamento(const char* s) {
    ifstream in;
    in.open(s);
    if(!in) return false;
    int m;
    in >>m; //dimensione della matrice
    if (!in || m<=0) return false;
    int *mat = new int [m*m];
    for (int k=0; k< m*m; k++) {
        in >> mat[k];
        if (!in) return false;
    }
    in.close();

    for (int i=0; i<m; i++) {
        if( !riga_ordinata(mat,m,i)) {
            delete[] mat;
            return false;
        }
    }

    delete[] mat;
    return true;
}
```

Esercizio 4

a) $(234)_5 = 50 + 15 + 4 = 69 \rightarrow 01000101$

b) La funzione assembler conta il numero di lettere maiuscole presenti nel vettore di caratteri puntato da vett, che è pari a tre. Pertanto l'uscita a video del programma è il numero tre in esadecimale, ossia stampa:
03

b) Esercizio C++

input -1

-1

nuova eccezione

ecc1

ecc main

input 0

0

nuova eccezione

ecc1

ecc main

input 1

1

ecc int main: 0