Ejercio 8

a) proc multiplicar (in $m1$: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in $m2$: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out res: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

Pre {

$|m1| > 0 \wedge_L$

$((\forall i : \mathbb{Z})\ 0 \leq i < |m1| \ \longrightarrow_L |m1[0]| = |m1[i]|) \wedge_L$

$((\forall i : \mathbb{Z})\ 0 \leq i < |m2| \ \longrightarrow_L |m2[0]| = |m2[i]|) \wedge_L$

$(|m1[0]| = |m2|)$

}

Post {

$|res| = |m1| \wedge_L$

$(\forall i : \mathbb{Z})\ 0 \leq i < |m1| \ \longrightarrow_L$

$(|res[i]| = |m2[0]| \wedge_L$

$(\forall j : \mathbb{Z})\ 0 \leq j < |m2[0]| \ \longrightarrow_L$

$res[i][j] = \sum_{k=0}^{|m2|-1} m1[i][k] * m2[k][j])))$

}

}

b)

```
#include <iostream>
#include "multiplicar.h"


vector<vector<int>> multiplicar(vector<vector<int>> m1, vector<vector<int>> m2) {    //|m1|=m*n, |m2|=n*l
    vector<vector<int >> result(0);                                          //1

    int i = 0;                                                                   //1
    while (i < m1.size()) {                                                      //2 + m *(2
        vector<int> aux(0);                                                  //1

        int j = 0;                                                               //1
        while (j < m2[0].size()) {                                               //3 + l *(3
            int acumulado = 0;                                                   //1
            int k = 0;                                                           //1
            while (k < m1[0].size()) {                                           //3 + n *(3
                acumulado = acumulado + (m1[i][k]) * (m2[k][j]);                 //7
                k++;                                                             //1)
            }
            aux.push_back(acumulado);                                            //1
            j++;                                                                 //1)
        }

        result.push_back(aux);                                                   //1
        i++;                                                                     //1)
    }

    return result;
} // = 4 + m *(9 + l *(10 + 11n) = O(m*l*n) sino es O(n^3)
```

c)

a) $O(filas_{m1} * columnas_{m1} * columnas_{m2})$

b) $O(N^3)$