#include <cmath>

#include <cstdio>

#include <cstring>

#include <string>

#include <string.h>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

char s1[1000];

char s2[1000];

char oper[10];

double matrix1[100][100];

double matrix2[100][100];

double Result[100][100];

double a[1000];

double b[1000];

double x;

int z = 0;

char v;

int r1 = 1, c1 = 0;

int r2 = 1, c2 = 0;

cin.getline(s1, 1000, '\n');

if (s1[0] != '[') /\*if user input a scalar value at the begining\*/

{

char del[] = " ";

char\* FPtr = strtok(s1, del);

x = stod(FPtr); /\*scalar value\*/

cin.getline(oper, 10, '\n');

char\* TPtr = strtok(oper, del);

v = \*TPtr;

if (v == '\*') /\*Operates Perfectily no problem\*/

{

cin.getline(s2, 1000, '\n');

int nValues = 0;

for (int y = 0; y < strlen(s2); y++) /\*Number of Rows in Matrix\*/

{

if (s2[y] == ',')

r1++;

}

for (int y = 1; s2[y] != ','; y++) /\*Number of Columns in Matrix\*/

{

if (s2[y] <= 1000 && s2[y] != 0 && s2[y - 1] != '.' && s2[y] != ' ' && (s2[y - 1] == '[' || s2[y - 1] == ' '))

c1++;

if (s2[y] <= 1000 && s2[y + 1] == ']')

{

break;

}

}

char\* tokenptr = strtok(s2, "[,] "); /\*Matrix convertion\*/

a[0] = stod(FPtr); /\*Convert string into array of doubles\*/

matrix1[0][0] = a[0];

while (tokenptr != NULL)

{

a[nValues] = stod(tokenptr);

tokenptr = strtok(NULL, "[,] ");

nValues++;

}

z = (nValues) % c1;

if (z == 0)

{

nValues = 0;

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

matrix1[w][u] = a[nValues];/\*Convert double array into 2D array\*/

nValues++;

}

}

/\*Result of multiplying matrix by scalar value and the output matrix in its form\*/

cout << "[";

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

Result[w][u] = x \* matrix1[w][u];

if (u < c1 - 1)

{

cout << Result[w][u] << " ";

}

else

{

cout << Result[w][u];

}

}

if (w < r1 - 1)

{

cout << "," << " ";

}

else

{

cout << "]" << endl;

}

}

}

else

cout << "ERROR!" << endl;

}

/\* if user inputs any operation instead of \*/

else

cout << "ERROR!" << endl;

}

else if (s1[0] == '[') /\*If First input is matrix\*/

{

int nValues = 0;

for (int y = 0; y < strlen(s1); y++) /\*Number of Rows in Matrix\*/

{

if (s1[y] == ',')

r1++;

}

for (int y = 1; s1[y] != ','; y++) /\*Number of Columns in Matrix\*/

{

if (s1[y] <= 1000 && s1[y - 1] != '.' && s1[y] != ' ' && (s1[y - 1] == '[' || s1[y - 1] == ' '))

c1++;

if (s1[y] <= 1000 && s1[y + 1] == ']')

{

break;

}

}

char\* tokenptr = strtok(s1, "[,] "); /\*First Matrix convertion\*/

a[0] = stod(tokenptr); /\*Convert string into array of doubles\*/

matrix1[0][0] = a[0];

while (tokenptr != NULL)

{

a[nValues] = stod(tokenptr);

tokenptr = strtok(NULL, "[,] ");

nValues++;

}

z = (nValues) % c1;

if (z == 0)

{

nValues = 0;

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

matrix1[w][u] = a[nValues]; /\*Convert double array into 2D array\*/

nValues++;

}

}

char del[] = " ";

cin.getline(oper, 10, '\n');

char\* TPtr = strtok(oper, del);

v = \*TPtr;

}

else

{

cout << "ERROR!" << endl;

return 0;

}

if (v == '+') /\*Operates Perfectily no problem\*/

{

cin.getline(s2, 1000, '\n'); /\* Case of selecting Addition\*/

int nValues = 0;

for (int y = 0; y < strlen(s2); y++) /\*Number of Rows in Matrix\*/

{

if (s2[y] == ',')

r2++;

}

for (int y = 1; s2[y] != ','; y++) /\*Number of Columns in Matrix\*/

{

if (s2[y] <= 1000 && s2[y - 1] != '.' && s2[y] != ' ' && (s2[y - 1] == '[' || s2[y - 1] == ' '))

c2++;

if (s2[y] <= 1000 && s2[y + 1] == ']')

{

break;

}

}

char\* tokenptr = strtok(s2, "[,] "); /\*Second Matrix convertion\*/

b[0] = stod(tokenptr); /\*Convert string into array of doubles\*/

matrix2[0][0] = b[0];

while (tokenptr != NULL)

{

b[nValues] = stod(tokenptr);

tokenptr = strtok(NULL, "[,] ");

nValues++;

}

nValues = 0;

for (int w = 0; w <= r2 - 1; w++)

{

for (int u = 0; u <= c2 - 1; u++)

{

matrix2[w][u] = b[nValues]; /\*Convert double array into 2D array\*/

nValues++;

}

}

if (r1 == r2 && c1 == c2) /\*For two equal matrices\*/

{

for (int w = 0; w <= r1 - 1; w++) /\*Result of adding the two matrices\*/

{

for (int u = 0; u <= c1 - 1; u++)

{

Result[w][u] = matrix1[w][u] + matrix2[w][u];

}

}

cout << "["; /\*The output matrix in its form\*/

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

if (u < c1 - 1)

{

cout << Result[w][u] << " ";

}

else

{

cout << Result[w][u];

}

}

if (w < r1 - 1)

{

cout << "," << " ";

}

else

{

cout << "]" << endl;

}

}

}

else

{

cout << "ERROR!" << endl;

}

}

else if (v == '-') /\*Operates Perfectily no problem\*/

{

cin.getline(s2, 1000, '\n'); /\* Case of selecting Substraction\*/

int nValues = 0;

for (int y = 0; y < strlen(s2); y++) /\*Number of Rows in Matrix\*/

{

if (s2[y] == ',')

r2++;

}

for (int y = 1; s2[y] != ','; y++) /\*Number of Columns in Matrix\*/

{

if (s2[y] <= 1000 && s2[y - 1] != '.' && s2[y] != ' ' && (s2[y - 1] == '[' || s2[y - 1] == ' '))

c2++;

if (s2[y] <= 1000 && s2[y + 1] == ']')

{

break;

}

}

char\* tokenptr = strtok(s2, "[,] "); /\*Second Matrix convertion\*/

b[0] = stod(tokenptr); /\*Convert string into array of doubles\*/

matrix2[0][0] = b[0];

while (tokenptr != NULL)

{

b[nValues] = stod(tokenptr);

tokenptr = strtok(NULL, "[,] ");

nValues++;

}

nValues = 0;

for (int w = 0; w <= r2 - 1; w++)

{

for (int u = 0; u <= c2 - 1; u++)

{

matrix2[w][u] = b[nValues];/\*Convert double array into 2D array\*/

nValues++;

}

}

if (r1 == r2 && c1 == c2) /\*For two equal matrices\*/

{

/\*Result of subtracting the two matrices\*/

cout << "["; /\*The output matrix in its form\*/

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

Result[w][u] = matrix1[w][u] - matrix2[w][u];

if (u < c1 - 1)

{

cout << Result[w][u] << " ";

}

else

{

cout << Result[w][u];

}

}

if (w < r1 - 1)

{

cout << "," << " ";

}

else

{

cout << "]" << endl;

}

}

}

else

{

cout << "ERROR!" << endl;

}

}

else if (v == '\*') /\*Operates Perfectily no problem\*/

{

cin.getline(s2, 1000, '\n'); /\* Case of selecting Multiplication\*/

if (s2[0] != '[') /\*if user input a scalar value\*/

{

char del[] = " ";

char\* FPtr = strtok(s2, del);

x = stod(FPtr); /\*scalar value\*/

cout << "["; /\*The output matrix in its form\*/

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c1 - 1; u++)

{

Result[w][u] = x \* matrix1[w][u];

if (u < c1 - 1)

{

cout << Result[w][u] << " ";

}

else

{

cout << Result[w][u];

}

}

if (w < r1 - 1)

{

cout << "," << " ";

}

else

{

cout << "]" << endl;

}

}

}

else if (s2[0] == '[')

{

int nValues = 0;

for (int y = 0; y < strlen(s2); y++) /\*Number of Rows in Matrix\*/

{

if (s2[y] == ',')

r2++;

}

for (int y = 1; s2[y] != ','; y++) /\*Number of Columns in Matrix\*/

{

if (s2[y] <= 1000 && s2[y - 1] != '.' && s2[y] != ' ' && (s2[y - 1] == '[' || s2[y - 1] == ' '))

c2++;

if (s2[y] <= 1000 && s2[y + 1] == ']')

{

break;

}

}

char\* tokenptr = strtok(s2, "[,] "); /\*Second Matrix convertion\*/

b[0] = stod(tokenptr); /\*Convert string into array of doubles\*/

matrix2[0][0] = b[0];

while (tokenptr != NULL)

{

b[nValues] = stod(tokenptr);

tokenptr = strtok(NULL, "[,] ");

nValues++;

}

nValues = 0;

for (int w = 0; w <= r2 - 1; w++)

{

for (int u = 0; u <= c2 - 1; u++)

{

matrix2[w][u] = b[nValues];/\*Convert double array into 2D array\*/

nValues++;

}

}

if (c1 == r2) /\*Multiplication of two matrices\*/

{

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c2 - 1; u++)

{

Result[w][u] = 0;

for (int p = 0; p <= c1 - 1; p++)

{

Result[w][u] += matrix1[w][p] \* matrix2[p][u];

}

}

}

cout << "["; /\*The output matrix in its form\*/

for (int w = 0; w <= r1 - 1; w++)

{

for (int u = 0; u <= c2 - 1; u++)

{

if (u < c2 - 1)

{

cout << Result[w][u] << " ";

}

else

{

cout << Result[w][u];

}

}

if (w < r1 - 1)

{

cout << "," << " ";

}

else

{

cout << "]" << endl;

}

}

}

else

{

cout << "ERROR!" << endl;

}

}

}

else

{

cout << "ERROR!" << endl;

}

}

}