
Matrix commands**X40782_en**

You have to complete the program given below. It provides functions to read matrices from the input, to compute the sum of two matrices and to compute the product of two matrices. The program reads commands from the input. Each command consists of a character to indicate which operation is requested (sum of two matrices, product of two matrices), followed by the two matrices in the format assumed by the function `read_matrix`; then the program outputs the sought answer.

You can add code only in the parts marked with ..., respecting the rest and making an adequate use of it with the appropriate calls and declarations.

You can use functions and procedures from the previous problems.

```
#include <iostream>
#include <vector>
using namespace std;

typedef vector<int> Row;
typedef vector<Row> Matrix;

// Reads a matrix with n rows and m columns
// from the input and returns it. Assumes
// that the input format is
// a00 ... a0(m-1) a10 ... a1(m-1) ... a(n-1)0 ... a(n-1)(m-1)
Matrix read_matrix(int n, int m)
{
    Matrix M(n, Row(m));
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j)
            cin >> M[i][j];
    return M;
}

// prints the given non-empty matrix into the cout; the first line
// gives the number of rows followed by the number
// of columns, then each successive row is printed
// in a different line, with each element of the row
// separated from the next by a blank space,
// and an end-of-line is printed at the end, after the
// last row of the matrix
void print_matrix(const Matrix& M) {
    ...
}

// Returns the sum of two matrices of correct dimensions
Matrix matrix_sum(const Matrix& A, const Matrix& B) {
    int n = A.size(); int m = A[0].size();
    Matrix C(n, Row(m));
```

```

    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j)
            C[i][j] = A[i][j] + B[i][j];
    return C;
}

// Returns the product of two matrices of correct
// dimensions
Matrix matrix_product(const Matrix& A, const Matrix& B) {
    int n = A.size(); int p = A[0].size();
    int m = B[0].size();
    Matrix C(n, Row(m,0));
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j)
            for (int k = 0; k < p; ++k)
                C[i][j] += A[i][k]*B[k][j];
    return C;
}

// The main program reads commands from the input and
// writes the appropriate answers in the output; the
// commands are either
// "+ <n> <m> <elements of A> <n'> <m'> <elements of B>",
// which writes the sum of the two matrices if the
// dimensions of the matrices are correct, and an error
// message otherwise; or
// "* <n> <m> <elements of A> <n'> <m'> <elements of B>",
// which writes the product of the two matrices if the
// dimensions of matrices are correct, and an error
// message otherwise.
int main() {
    char s;
    while (cin >> s) {
        ...
        if (s == '+') {
            ...
        } else if (s == '*') {
            ...
        }
    }
}

```

Exam score: 2.5 Automatic part: 20%

Input

The input has several cases. Each case starts with a character '+' or '*' followed by a line break. Next, we have the description of two matrices. For every matrix we have its dimensions $n, m \geq 1$, followed by its elements in row order. Every integer is separated by the next

one by a blank space and each row is ended with a line break. Two consecutive matrices or operators are separated by a blank line.

Output

For each case print the corresponding matrix result or notify that such an operation cannot be performed. Look at the sample tests to see the expected format. Note that the result for each case is followed by a blank line.

Sample input

```
+  
  
2 2  
1 0  
0 1  
  
2 2  
0 -1  
-1 1  
  
*  
  
2 2  
1 0  
0 1  
  
2 3  
-1 -2 -3  
1 2 3  
  
*  
  
3 2  
1 2  
3 4  
5 6  
  
2 4  
-1 -2 -3 -4  
-5 -6 -7 -8  
  
+  
  
1 1  
2  
  
2 2  
1 0  
0 1  
  
*  
  
2 1  
3  
2  
  
2 2  
1 0  
0 1
```

Sample output

```
2 2  
1 -1  
-1 2  
  
2 3  
-1 -2 -3  
1 2 3  
  
3 4  
-11 -14 -17 -20  
-23 -30 -37 -44  
-35 -46 -57 -68  
  
The two matrices cannot be added.  
  
The two matrices cannot be multiplied.
```

Problem information

Author : Professorat de PRO1

Generation : 2015-12-12 21:51:20

© *Jutge.org*, 2006–2015.

<http://www.jutge.org>