

# **DATA STRUCTURES AND ALGORITHMS**

---

PRACTICE No.11

# Exercises

## Exercise No.1

Multiply two matrices according to calculus rules (row x column) and using high performance way (row x row):

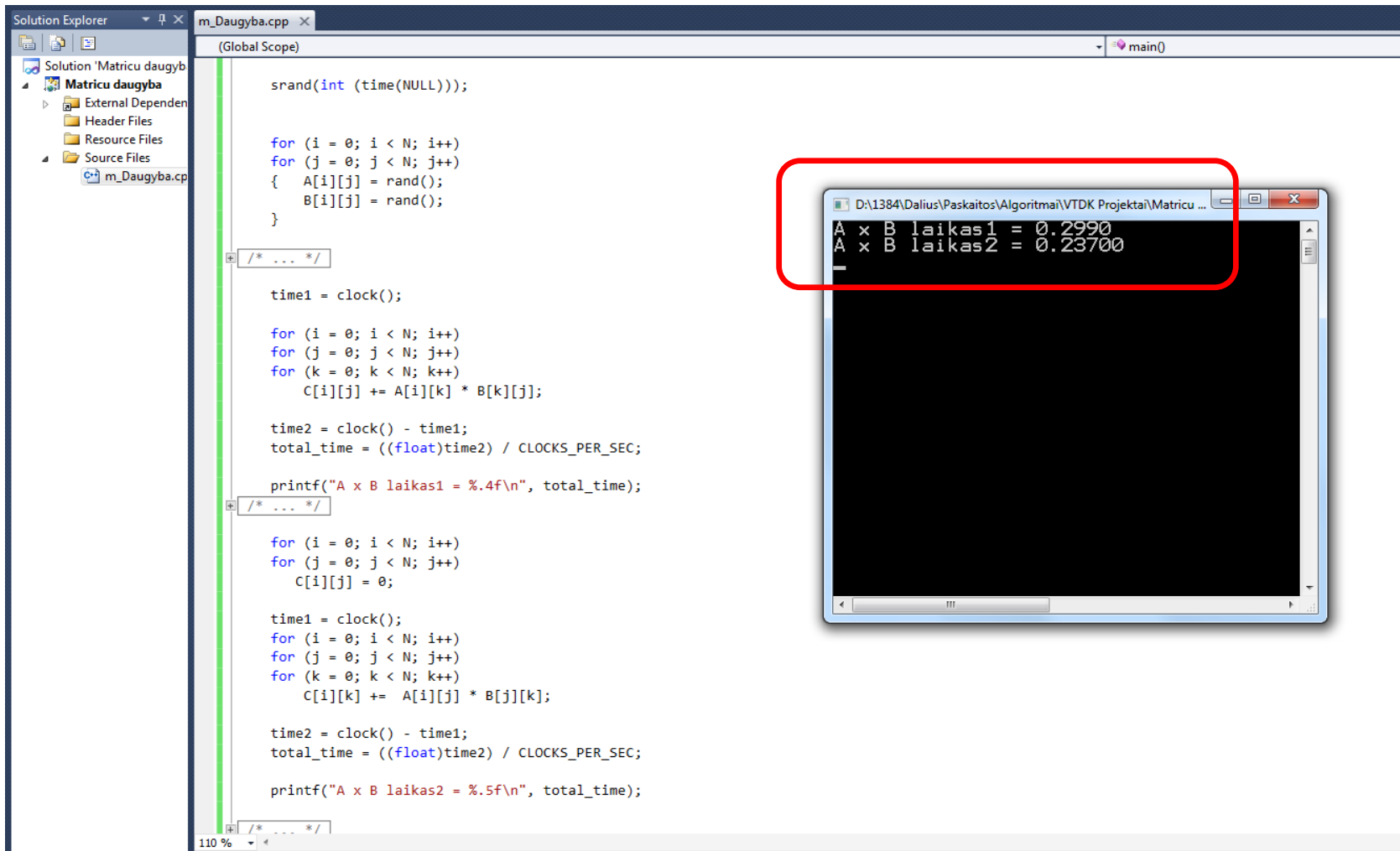
$$[C] = [A] \times [B]$$

Compare running time of these two algorithms.

**Note:**

**Matrices are stored row by row in the primary memory.**

# Results



The image shows a Visual Studio IDE window with a C++ source file named `m_Daugyba.cpp`. The code implements two matrix multiplication functions. The first function, `main()`, generates two random matrices `A` and `B` and then calls a function to multiply them. The second function performs a standard row-by-column matrix multiplication. The execution output is shown in a separate window, displaying the results of the two multiplication methods.

```
srand(int (time(NULL)));

for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
{ A[i][j] = rand();
  B[i][j] = rand();
}

/* ... */

time1 = clock();

for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
for (k = 0; k < N; k++)
    C[i][j] += A[i][k] * B[k][j];

time2 = clock() - time1;
total_time = ((float)time2) / CLOCKS_PER_SEC;

printf("A x B laikas1 = %.4f\n", total_time);

/* ... */

for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
    C[i][j] = 0;

time1 = clock();
for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
for (k = 0; k < N; k++)
    C[i][k] += A[i][j] * B[j][k];

time2 = clock() - time1;
total_time = ((float)time2) / CLOCKS_PER_SEC;

printf("A x B laikas2 = %.5f\n", total_time);

/* ... */
```

Execution Output:

```
A x B laikas1 = 0.2990
A x B laikas2 = 0.23700
```

# Exercises

## Exercise No.2

Data set consists of 100 random numbers. Select unique numbers from data set. Implement two different algorithms for problem solving and compare running time of these algorithms.

## Exercise No.3

Compare running time of the Quick sort algorithm, when pivot is:

- the leftmost element
- the rightmost element
- random element of the array
- middle element of the array

# Homework

## Arithmetic of Long numbers

### No.1

Real number has one digit in whole part and 30 decimal digits. All numbers are random.

Round decimal digits numbers till the position that are given interactively.

### No.2

Multiply long number (30 digits) by the constant number.