

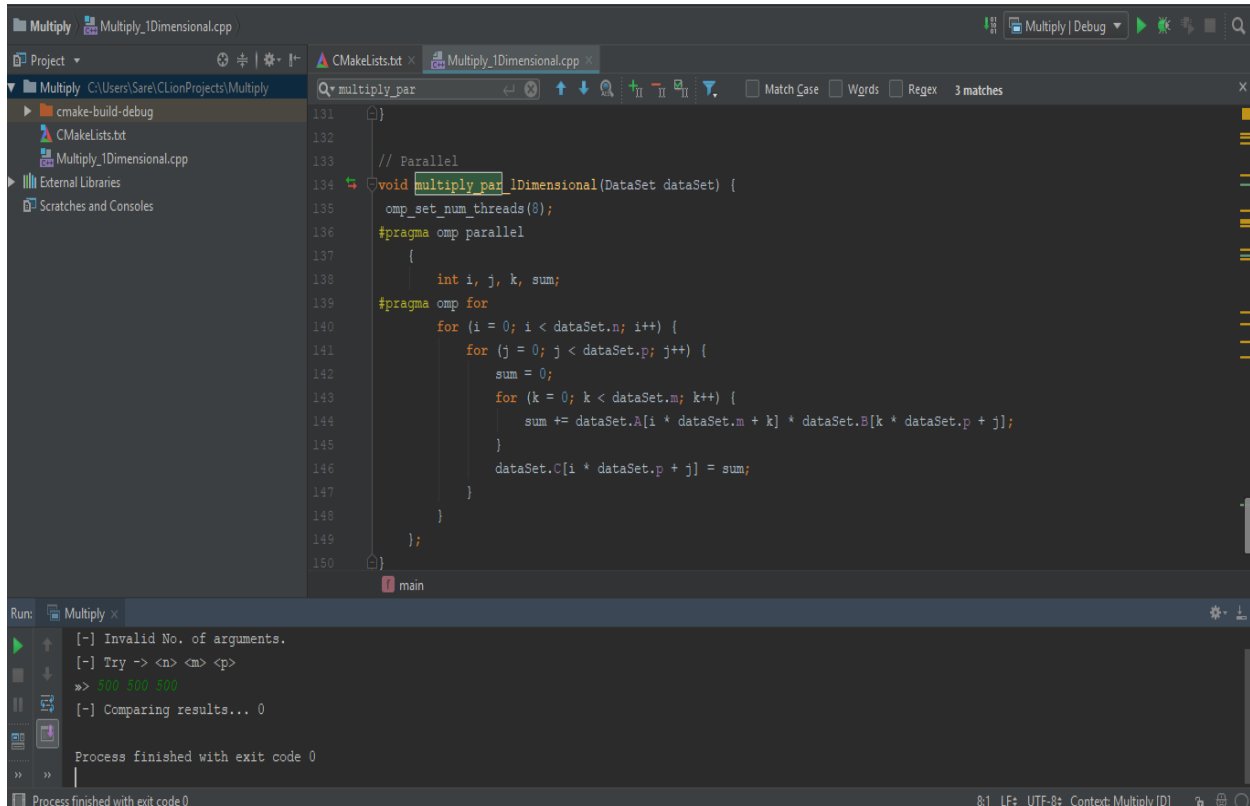
HW2

ساره سلطانی نژاد ۹۳۳۱۰۳۹

نتایج در فایل اکسل آورده شده است.

مقایسه درستی انجام عملیات ضرب

یک بعدی:

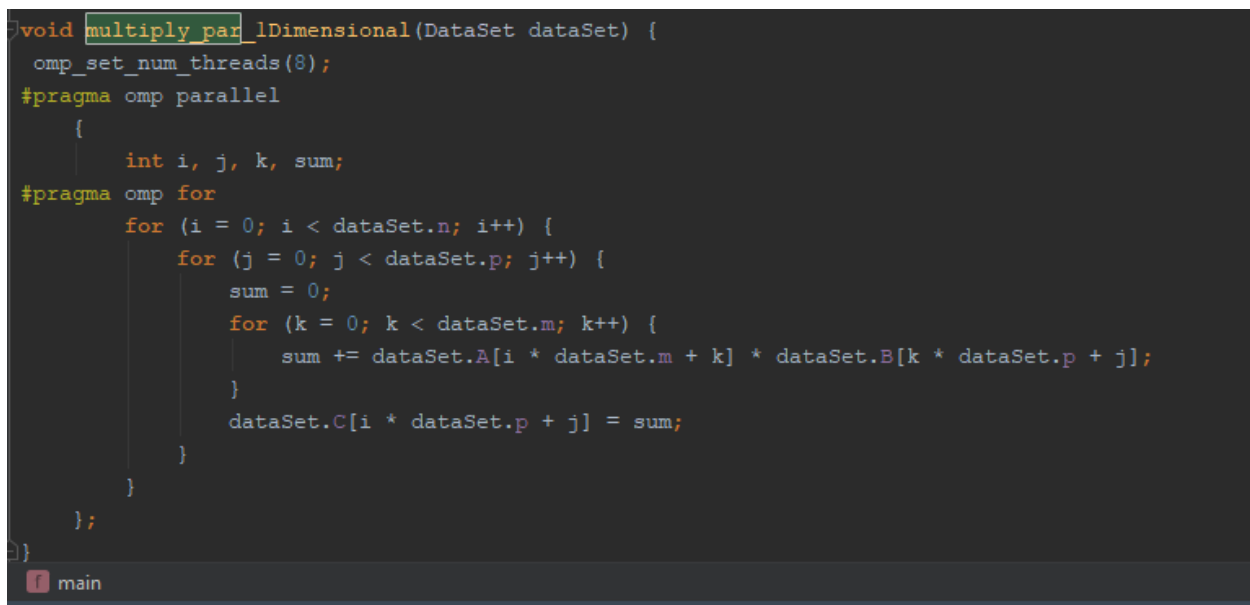


The screenshot shows a C++ IDE with a project named 'Multiply'. The file 'Multiply_1Dimensional.cpp' is open, showing a function 'multiply_parallel_1Dimensional' that uses OpenMP for parallelization. The function sets 8 threads and performs a triple loop over dimensions n, p, and m to calculate the product of two matrices A and B, storing the result in matrix C. The IDE's Run window shows the output of the program, which includes an error message about invalid arguments, a comparison of results showing 0, and a final message indicating the process finished with exit code 0.

```
131 }
132 // Parallel
133 void multiply_parallel_1Dimensional(DataSet dataSet) {
134     omp_set_num_threads(8);
135     #pragma omp parallel
136     {
137         int i, j, k, sum;
138     #pragma omp for
139         for (i = 0; i < dataSet.n; i++) {
140             for (j = 0; j < dataSet.p; j++) {
141                 sum = 0;
142                 for (k = 0; k < dataSet.m; k++) {
143                     sum += dataSet.A[i * dataSet.m + k] * dataSet.B[k * dataSet.p + j];
144                 }
145                 dataSet.C[i * dataSet.p + j] = sum;
146             }
147         }
148     }
149 };
150 }
```

Run: Multiply x

```
[!] Invalid No. of arguments.
[!] Try -> <n> <m> <p>
>> 400 400 400
[!] Comparing results... 0
Process finished with exit code 0
```



This is a close-up view of the 'multiply_parallel_1Dimensional' function. It shows the function signature, the setting of 8 threads, and the use of OpenMP's 'parallel' and 'for' pragmas to parallelize the nested loops over dimensions n, p, and m. The code calculates the product of two matrices A and B, storing the result in matrix C.

```
void multiply_parallel_1Dimensional(DataSet dataSet) {
    omp_set_num_threads(8);
    #pragma omp parallel
    {
        int i, j, k, sum;
    #pragma omp for
        for (i = 0; i < dataSet.n; i++) {
            for (j = 0; j < dataSet.p; j++) {
                sum = 0;
                for (k = 0; k < dataSet.m; k++) {
                    sum += dataSet.A[i * dataSet.m + k] * dataSet.B[k * dataSet.p + j];
                }
                dataSet.C[i * dataSet.p + j] = sum;
            }
        }
    }
};
}
```

دوبعدی

The screenshot shows a C++ IDE with the following components:

- Project Explorer:** Shows the project structure with files like `cmake-build-debug`, `CMakeLists.txt`, `Multiply_1Dimensional.cpp`, and `Multiply_2Dimensional.cpp`.
- Source Editor:** Displays the code for `Multiply_2Dimensional.cpp`. The code is as follows:


```

163 // Parallel
164 void multiply_par_2Dimensional(DataSet dataSet) {
165     omp_set_num_threads(8);
166     #pragma omp parallel
167     {
168         int sum;
169     #pragma omp for
170         for (int i = 0; i < dataSet.n; i++) {
171     #pragma omp parallel for
172         for (int j = 0; j < dataSet.p; j++) {
173             sum = 0;
174             for (int k = 0; k < dataSet.m; k++) {
175                 sum += dataSet.A[i * dataSet.m + k] * dataSet.B[k * dataSet.p + j];
176             }
177             dataSet.C[i * dataSet.p + j] = sum;
178         }
179     }
180 }
      
```
- Run Window:** Shows the execution output for `Multiply.exe`. The output includes:


```

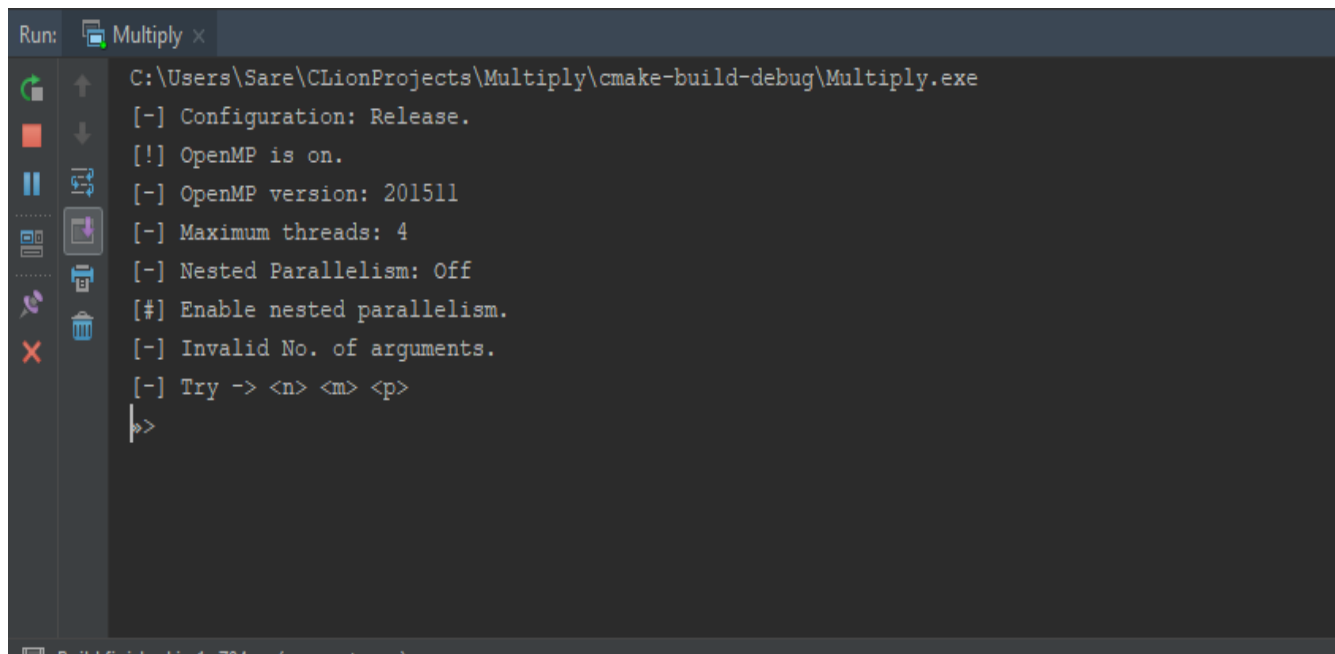
[-] Configuration: Release.
[!] OpenMP is on.
[-] OpenMP version: 201511
[-] Maximum threads: 4
[-] Nested Parallelism: Off
[#] Enable nested parallelism.
[-] Invalid No. of arguments.
[-] Try -> <n> <m> <p>
>> 800 100 100
[-] Comparing results... 0
Process finished with exit code 0
      
```

This is a close-up view of the source code from the previous screenshot, showing the `multiply_par_2Dimensional` function. The code is as follows:


```

// Parallel
void multiply_par_2Dimensional(DataSet dataSet) {
    omp_set_num_threads(8);
    #pragma omp parallel
    {
        int sum;
    #pragma omp for
        for (int i = 0; i < dataSet.n; i++) {
    #pragma omp parallel for
        for (int j = 0; j < dataSet.p; j++) {
            sum = 0;
            for (int k = 0; k < dataSet.m; k++) {
                sum += dataSet.A[i * dataSet.m + k] * dataSet.B[k * dataSet.p + j];
            }
            dataSet.C[i * dataSet.p + j] = sum;
        }
    }
}
    
```

بررسی تنظیمات OMP



The screenshot shows the 'Run' window of the CMake GUI. The title bar indicates the file 'Multiply' is open. The window displays the execution path and various OpenMP configuration options. The output text is as follows:

```
C:\Users\Sare\CLionProjects\Multiply\cmake-build-debug\Multiply.exe  
[-] Configuration: Release.  
[!] OpenMP is on.  
[-] OpenMP version: 201511  
[-] Maximum threads: 4  
[-] Nested Parallelism: Off  
[#] Enable nested parallelism.  
[-] Invalid No. of arguments.  
[-] Try -> <n> <m> <p>  
|>
```

On the left side of the window, there is a vertical toolbar with icons for running, stepping through code, and other debugging functions.