

## Parallel and Distributed Computing

### Lab 2

Shazain

27115

Execution Time in Seconds					
	C/C++	Java	Matlab	Python-list	Python-NumPy
Without loop $M1=M2+M3$	N/A	N/A	0.000769	0.108537	0.003045
Row Major loops (R-C)	0.00519 6	0.00874602 9	5.741612	0.000769	0.451024
Column Major loops (C-R)	0.01619 7	0.10702003	4.555673	0.149490	0.439505

C

The screenshot shows the Visual Studio Code editor with a C file named `PDCL2.c`. The code implements matrix operations: `freeMemory()`, `allocateMemory()`, `initialize()`, `clock_t` for timing, `start`, `end`, `addRowMajor()`, `addColumnMajor()`, `directAddition()`, and `main()`. The `main` function calls these functions and prints execution times. The terminal at the bottom shows the command `./PDCL2` being executed, resulting in the following output:

```
[yasq@yasq-pc lab2]$ ./PDCL2
Row Major Execution Time: 0.000196 seconds
Column Major Execution Time: 0.016197 seconds
Direct Addition Execution Time: 0.000017 seconds
[yasq@yasq-pc lab2]$ ^C
[yasq@yasq-pc lab2]$
```

Java

The screenshot shows the Programiz Online Java Compiler interface. The code in `Main.java` is as follows:

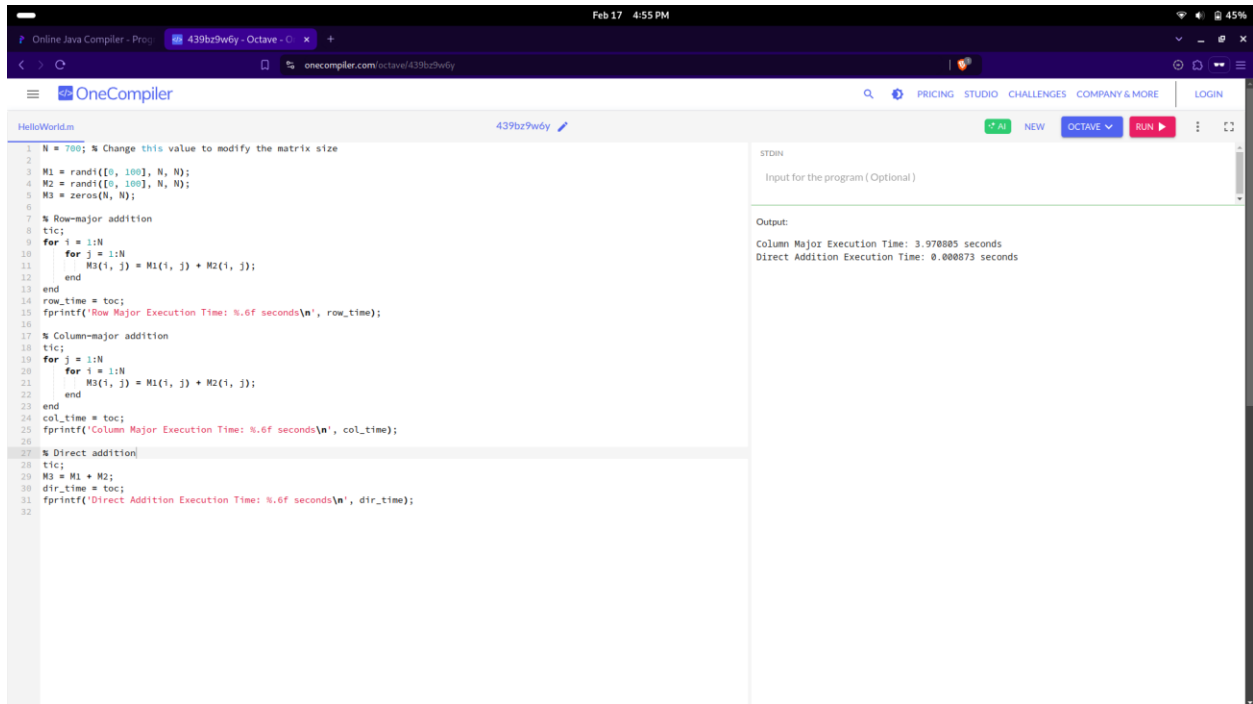
```
1- import java.util.Random;
2
3- public class Main {
4     static final int N = 1000;
5     static int[][] M1 = new int[N][N];
6     static int[][] M2 = new int[N][N];
7     static int[][] M3 = new int[N][N];
8
9     // Function to initialize matrices with random values
10    static void initialize() {
11        Random rand = new Random();
12        for (int i = 0; i < N; i++)
13            for (int j = 0; j < N; j++) {
14                M1[i][j] = rand.nextInt(100);
15                M2[i][j] = rand.nextInt(100);
16            }
17    }
18
19    // Row-major addition
20    static void addRowMajor() {
21        for (int i = 0; i < N; i++)
22            for (int j = 0; j < N; j++)
23                M3[i][j] = M1[i][j] + M2[i][j];
24    }
25
26    // Column-major addition
27    static void addColumnMajor() {
28        for (int j = 0; j < N; j++)
29            for (int i = 0; i < N; i++)
30                M3[i][j] = M1[i][j] + M2[i][j];
31    }
32
33    public static void main(String[] args) {
34        initialize();
35
36        long startTime, endTime;
37    }
```

The output on the right shows the execution times:

```
Row Major Execution Time: 8.746029 ms
Column Major Execution Time: 107.020037 ms
Direct Addition Execution Time: 101.117218 ms

=== Code Execution Successful ===
```

## Octave (Matlab)



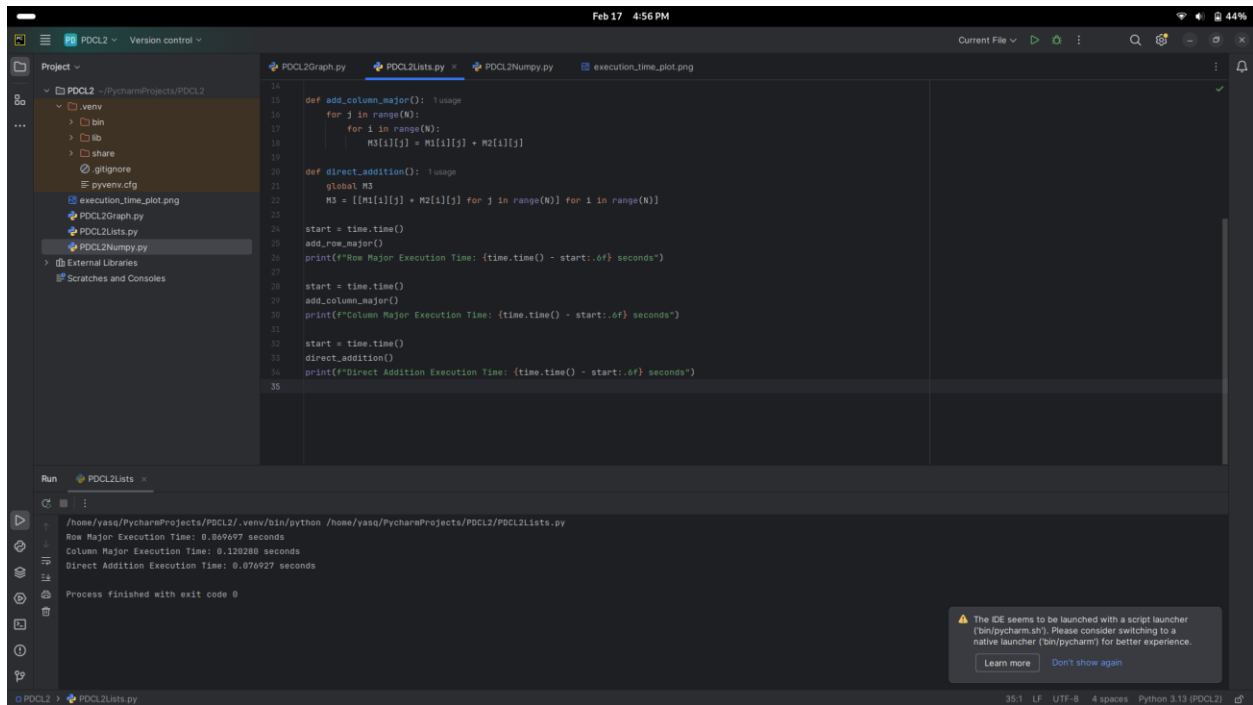
The screenshot shows the OneCompiler Octave IDE. The editor contains a script for matrix addition. The code defines a matrix size `N = 700`, generates two random matrices `M1` and `M2`, and initializes a zero matrix `M3`. It then performs three types of addition: row-major, column-major, and direct. Each type is timed and the results are printed. The output panel shows the execution times: Column Major Execution Time: 3.970885 seconds and Direct Addition Execution Time: 0.000873 seconds.

```
1 N = 700; % Change this value to modify the matrix size
2
3 M1 = randi([0, 100], N, N);
4 M2 = randi([0, 100], N, N);
5 M3 = zeros(N, N);
6
7 % Row-major addition
8 tic;
9 for i = 1:N
10     for j = 1:N
11         M3(i, j) = M1(i, j) + M2(i, j);
12     end
13 end
14 row_time = toc;
15 fprintf('Row Major Execution Time: %.6f seconds\n', row_time);
16
17 % Column-major addition
18 tic;
19 for j = 1:N
20     for i = 1:N
21         M3(i, j) = M1(i, j) + M2(i, j);
22     end
23 end
24 col_time = toc;
25 fprintf('Column Major Execution Time: %.6f seconds\n', col_time);
26
27 % Direct addition
28 tic;
29 M3 = M1 + M2;
30 dir_time = toc;
31 fprintf('Direct Addition Execution Time: %.6f seconds\n', dir_time);
32
```

Output:

Column Major Execution Time: 3.970885 seconds  
Direct Addition Execution Time: 0.000873 seconds

## Python Lists



The screenshot shows the PyCharm IDE with a project named 'POCL2'. The editor displays a Python script named 'POCL2Lists.py' that implements matrix addition using lists. The code defines functions for row-major, column-major, and direct addition, each with timing. The output panel shows the execution times: Row Major Execution Time: 0.069697 seconds, Column Major Execution Time: 0.120280 seconds, and Direct Addition Execution Time: 0.076927 seconds. A warning message at the bottom indicates that the IDE is launched with a script launcher and suggests switching to a native launcher for better experience.

```
14 def add_column_major():
15     for j in range(N):
16         for i in range(N):
17             M3[i][j] = M1[i][j] + M2[i][j]
18
19 def direct_addition():
20     global M3
21     M3 = [[M1[i][j] + M2[i][j] for j in range(N)] for i in range(N)]
22
23 start = time.time()
24 add_row_major()
25 print(f"Row Major Execution Time: {time.time() - start:.6f} seconds")
26
27 start = time.time()
28 add_column_major()
29 print(f"Column Major Execution Time: {time.time() - start:.6f} seconds")
30
31 start = time.time()
32 direct_addition()
33 print(f"Direct Addition Execution Time: {time.time() - start:.6f} seconds")
34
```

Run POCL2Lists

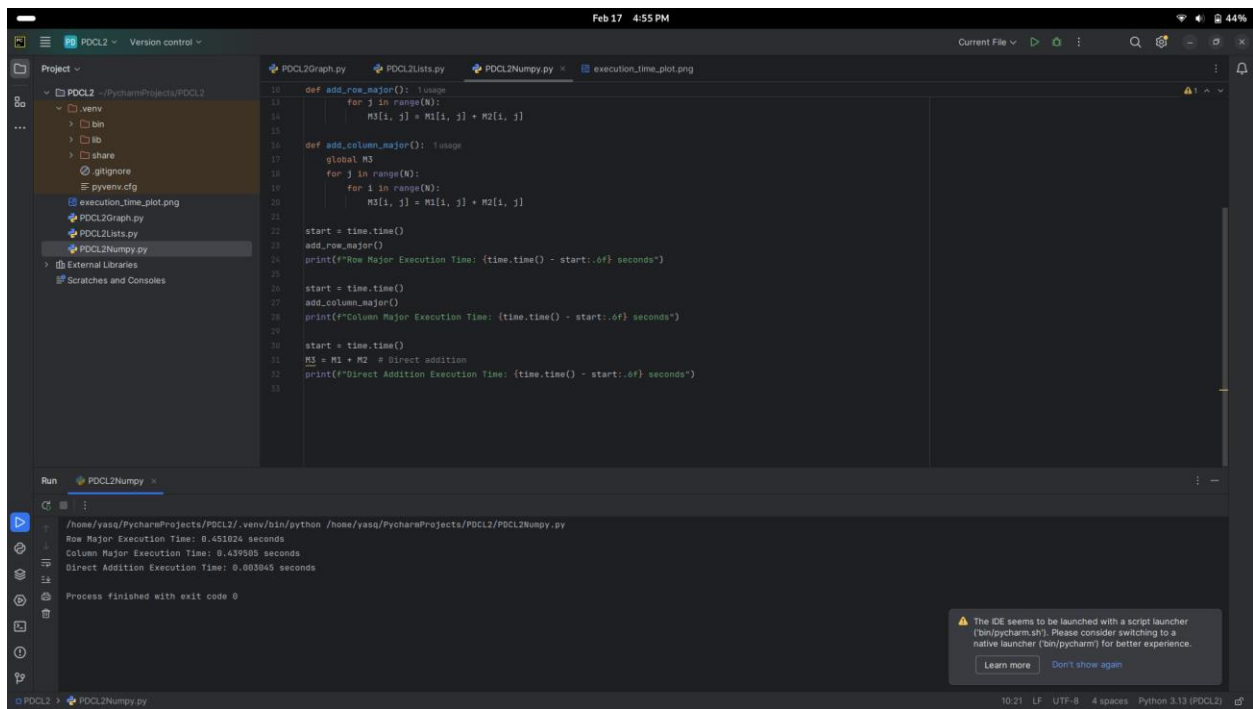
/home/yass/PycharmProjects/POCL2/.venv/bin/python /home/yass/PycharmProjects/POCL2/POCL2Lists.py

Row Major Execution Time: 0.069697 seconds  
Column Major Execution Time: 0.120280 seconds  
Direct Addition Execution Time: 0.076927 seconds

Process finished with exit code 0

The IDE seems to be launched with a script launcher ('bin/pycharm.sh'). Please consider switching to a native launcher ('bin/pycharm') for better experience.

# Python Numpy



The screenshot displays the PyCharm IDE interface. The top toolbar shows the time as Feb 17 4:55 PM. The left sidebar shows the project structure for 'PDCL2'. The main editor window displays the file 'PDCL2Numpy.py' with the following code:

```
10 def add_row_major(): usage:
11     for j in range(N):
12         M3[i, j] = M1[i, j] + M2[i, j]
13
14 def add_column_major(): usage:
15     global M3
16     for j in range(N):
17         for i in range(N):
18             M3[i, j] = M1[i, j] + M2[i, j]
19
20 start = time.time()
21 add_row_major()
22 print(f"Row Major Execution Time: {time.time() - start:.6f} seconds")
23
24 start = time.time()
25 add_column_major()
26 print(f"Column Major Execution Time: {time.time() - start:.6f} seconds")
27
28 start = time.time()
29 M3 = M1 + M2 # Direct addition
30 print(f"Direct Addition Execution Time: {time.time() - start:.6f} seconds")
31
```

The Run console at the bottom shows the output of the script:

```
/home/yass/PycharmProjects/PDCL2/.venv/bin/python /home/yass/PycharmProjects/PDCL2/PDCL2Numpy.py
Row Major Execution Time: 0.451024 seconds
Column Major Execution Time: 0.439585 seconds
Direct Addition Execution Time: 0.003845 seconds
Process finished with exit code 0
```

A warning message is displayed in the bottom right corner: "The IDE seems to be launched with a script launcher (bin/pycharm.sh). Please consider switching to a native launcher (bin/pycharm) for better experience." with buttons for "Learn more" and "Don't show again".