# Simple-MultiThreader

## Group Members

- **Nihal (2023345)**
- **Namit Bajaj (2023340)**

**Repo Link: [https://github.com/GolDRoger69/SimpleMultithreader](https://github.com/GolDRoger69/SimpleMultithreader)**

## Contributions

Both have the same contribution in Assignment in Implementing the logic of code code and error handling.

### Methods

#### Single-dimensional Parallel Loop

```
void parallel_for(int strt, int end, function<void(int)>&& lambda, int
num_threads)
```

- **Description**: Parallelizes a single-dimensional loop using num_threads threads.
- **Parameters**:
  - strt: Starting index of the loop.
  - end: Ending index of the loop (exclusive).
  - lambda: A lambda function that defines the loop body, accepting a single parameter (loop index).
  - num_threads: Number of threads to use for execution.

#### Nested Two-dimensional Parallel Loop

```
void parallel_for(int o_strt, int o_end, int i_strt, int i_end,
function<void(int, int)>&& lambda, int num_threads)
```

- **Description**: Parallelizes a nested two-dimensional loop using num_threads threads.
- **Parameters**:
    - o_strt and o_end: Start and end indices for the outer loop.
    - i_strt and i_end: Start and end indices for the inner loop.
    - lambda: A lambda function that defines the nested loop body, accepting two parameters (outer and inner loop indices).
    - num_threads: Number of threads to use for execution.

### Sequential Single-dimensional Loop

```
void loop(int strt, int end, function<void(int)>&& lambda_f)
```

- **Description**: Executes a single-dimensional loop sequentially.
- **Parameters**:
    - strt: Starting index of the loop.
    - end: Ending index of the loop (exclusive).
    - lambda: A lambda function that defines the loop body, accepting a single parameter (loop index).

### Sequential Nested Two-dimensional Loop

```
void n_loop(int o_strt, int o_end, int i_strt, int i_end,
function<void(int, int)>&& lambda_f)
```

- **Description**: Executes a nested two-dimensional loop sequentially.
- **Parameters**:
    - o_strt and o_end: Start and end indices for the outer loop.
    - i_strt and i_end: Start and end indices for the inner loop.
    - lambda: A lambda function that defines the nested loop body, accepting two parameters (outer and inner loop indices).

# Usage Instructions

### 1. Prerequisites

- Ensure that you have:
    - **g++** with **C++11** support.
    - **make** (optional) for building the project.

## 2. Compilation

Use the provided Makefile for easy compilation:

bash
>>>make

This will produce an executable file.

## 3. Running the Program

Run the executable file as follows:

bash
>>>./<executable_name> <Number_Of_Threads> <Size>

## 4. View Output

Observe the output in the terminal, which will include information about the execution time of parallel_for calls and the demonstration of lambda functions.