ALGORITHMS ANALYSIS AND DESIGN

Homework#1

Name: Ahmad Alselawi

ID: 201920086

Iterative Code:

```
<include <iostream#
include <chrono> // library to calculate the time#

;using namespace std
;using namespace chrono

factorial iteratively function  //
} unsigned long long factorialIterative(int n)
;unsigned long long result = 1
} for (int i = 2; i <= n; ++i)
;result *= i
{
;return result
{

} ()int main
enter the number n //
```

```
;int n
;" :cout << "Enter the number n
;cin >> n

To Start calculate //
;()auto startCalclate = high_resolution_clock::now

Call function //
;unsigned long long result = factorialIterative(n)

Stop the calculate //
;()auto stopCalculate = high_resolution_clock::now

calculate time//
auto duration = duration_cast<microseconds>(stopCalculate -
;startCalclate)

Display the Result and Time //
;cout << "Factorial of " << n << " = " << result << endl
cout << "Execution time  : " << duration.count() << " microseconds"
;<< endl
;return 0
{
```

Recursive Code:

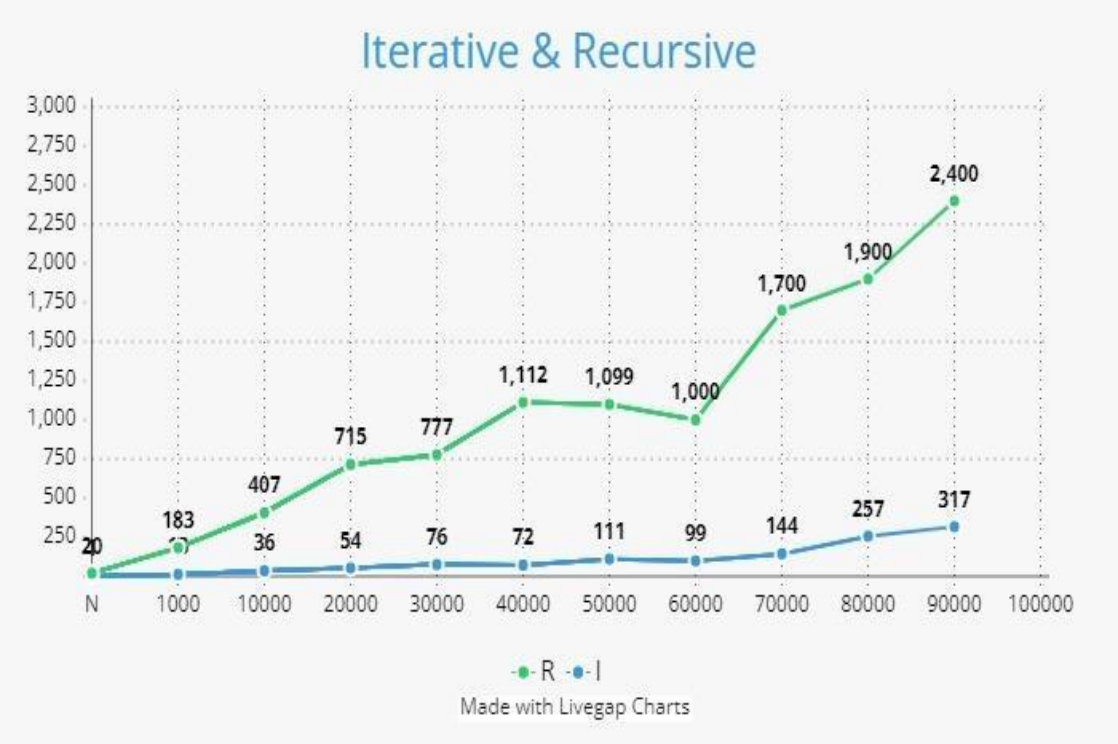
```
<include <iostream#
<include <chrono#
```

```
;using namespace std
;using namespace std::chrono

} unsigned long long factorial(int n)
} if (n <= 1)
;return 1
} else {
;return n * factorial(n - 1)
{
{

} ()int main
;int n
;" :cout << "Enter a number to calculate its factorial
;cin >> n

;()auto start = high_resolution_clock::now
;unsigned long long result = factorial(n)
;()auto stop = high_resolution_clock::now

;auto duration = duration_cast<microseconds>(stop - start)
;cout << "Factorial of " << n << " is " << result << endl
;cout << "Time taken: " << duration.count() << " microseconds" << endl

;return 0
{
```

Table of compare:

#	Recursive **Time in microsecond**	Iterative **Time in microsecond**
1000	**20**	**1**
10000	**183**	**13**
20000	**407**	**36**
30000	**715**	**54**
40000	**777**	**76**
50000	**1112**	**72**
60000	**1099**	**111**
70000	**1000**	**99**
80000	**1700**	**144**
90000	**1900**	**257**
100000	**2400**	**317**

Discussion:



Conclusion:

we made this comparison to find the execution time required to calculate the factorial of an integer N, Using two different calculation methods : Iterative and Recursive.

According to the results shown in the table and chart, we say that

- The choice between them depends on the specific problem, the characteristics of the input data and the requirements of the application. Each one has some advantages and disadvantages.
- Iterative Function : use less memory and don't need an additional space on the call stack for each recursive call. But some problems are more naturally expressed using recursion.
- Recursive Function : some problems have more natural and concise expression using recursion, leading to code that is often more readable. But
- recursive solutions can be less efficient in terms of both time and space complexity compared to their Iterative solutions, depending on the language and the implementation.