ZUHAIR ARARAWI

202011380

In this robot we will discuss factorial number in two ways and finding time when given values.

1 iterative:

Code:

```
#include <iostream> #include <ctime> using namespace std; // Function to calculate factorial iteratively unsigned long long factorialIterative(int n) { unsigned long long result = 1; for (int i = 2; i <= n; ++i) { result *= i; } return result; } int main() { int n; cout << "Enter a number to calculate its factorial: "; cin >> n; clock_t start = clock(); // Start measuring time // Calculate factorial unsigned long long fact = factorialIterative(n); clock_t end = clock(); // Stop measuring time double duration = (double)(end - start) / CLOCKS_PER_SEC; cout << "Factorial of " << n << " is: " << fact << endl; cout << "Time taken for execution: " << duration << " seconds" << endl; return 0; }
```

I used the ctime library to find the time and I used the C++ programming language.

2 Recursion:

Code:

```
#include <iostream> #include <ctime> using namespace std; // Function to calculate factorial recursively unsigned long long factorialRecursive(int n) { if (n == 0 || n == 1) return 1; else return n * factorialRecursive(n - 1); } int main() { int n; cout << "Enter a number to calculate its factorial: "; cin >> n; clock_t start = clock(); // Start measuring time // Calculate factorial unsigned long long fact = factorialRecursive(n); clock_t end = clock(); // Stop measuring time double duration = (double)(end - start) / CLOCKS_PER_SEC; cout << "Factorial of " << n << " is: " << fact << endl; cout << "Time taken for execution: " << duration << " seconds" << endl; return 0; }
```

I will put several values in the two codes, and I will check the time taken, and I will explain this matter in a table.

The values I will enter are:

50

100

150

200

250

value	Time in iterative	Time in recursion
50	0.09	1
100	0.3	3
150	1	6
200	3	12
250	4	13

I have tried many numbers and it turns out that iterative is faster than recursion and recursion does not respond after a value of 150000. Stack overflow occurs.