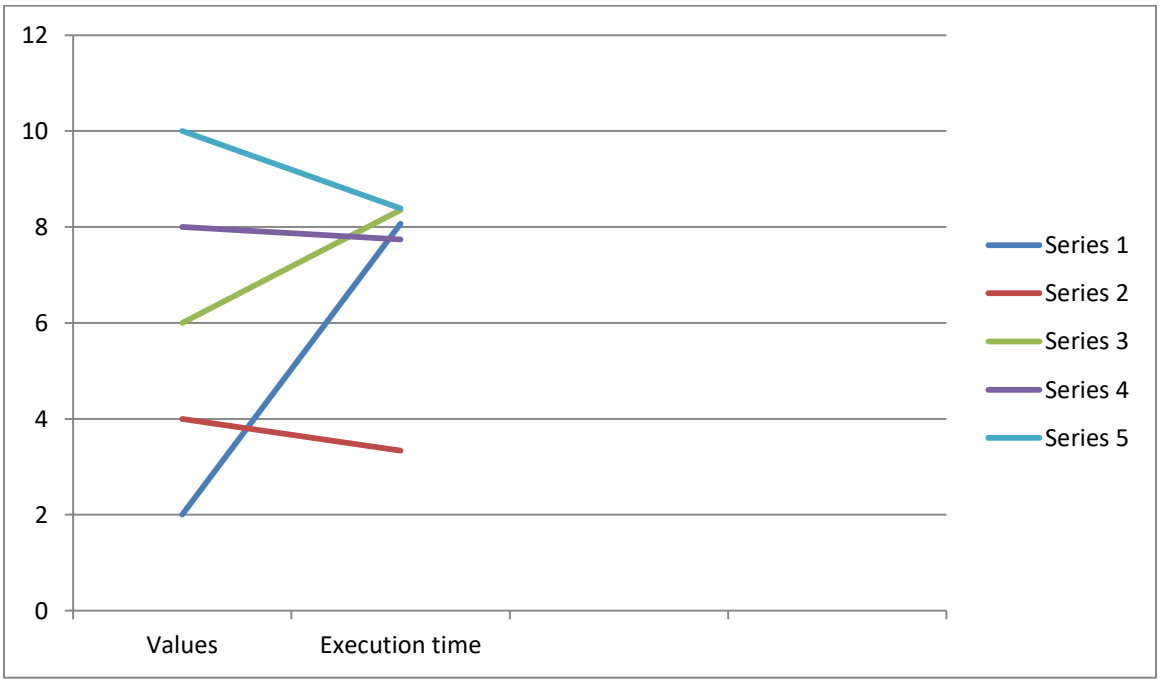
The values	Execution time
2	8.0653
4	3.33795
6	8.35671
8	7.74151
10	8.38909



We will not disagree that the recarsive function does not accommodate large numbers, while the iterative function is better and can accommodate larger numbers. The time consumed is very similar

```
#include <iostream>
 #include <chrono>
#include <vector>
#include <functional>


using namespace std;
  int fact(unsigned long long int n){
   if(n==0)
    return 1;
   else
    return n*fact(n-1);

   }



template<typename F, typename T>
double time_exec(F func, T start, T end) {
```

```
    // Execute the code.
    for(auto i = start; i != end; ++i) {
        func(*i);
    }
    // Calculate the duration in microseconds.
    long int end_time,start_time;
    //auto duration =
std::chrono::duration_cast<std::chrono::microseconds>(end_time
- start_time);
}


int main() {


    unsigned long long int m;
    cin>>m;
    int f=1;
    for(int i=m;i>=1;i--){
        f*=i;
    }
    cout<<f;
```

```
long int v;
cin>>v;
    cout<<fact(v)<<endl;


   std::vector<int> data = {1, 2, 3, 4, 5};
   auto time_taken = time_exec([](int i) {
      // Your code to execute goes here.
   }, data.begin(), data.end());


  std::cout << "Execution time: " << time_taken << " seconds" <<
std::endl;}
```