NAME: Malak Ashraf Mohammad

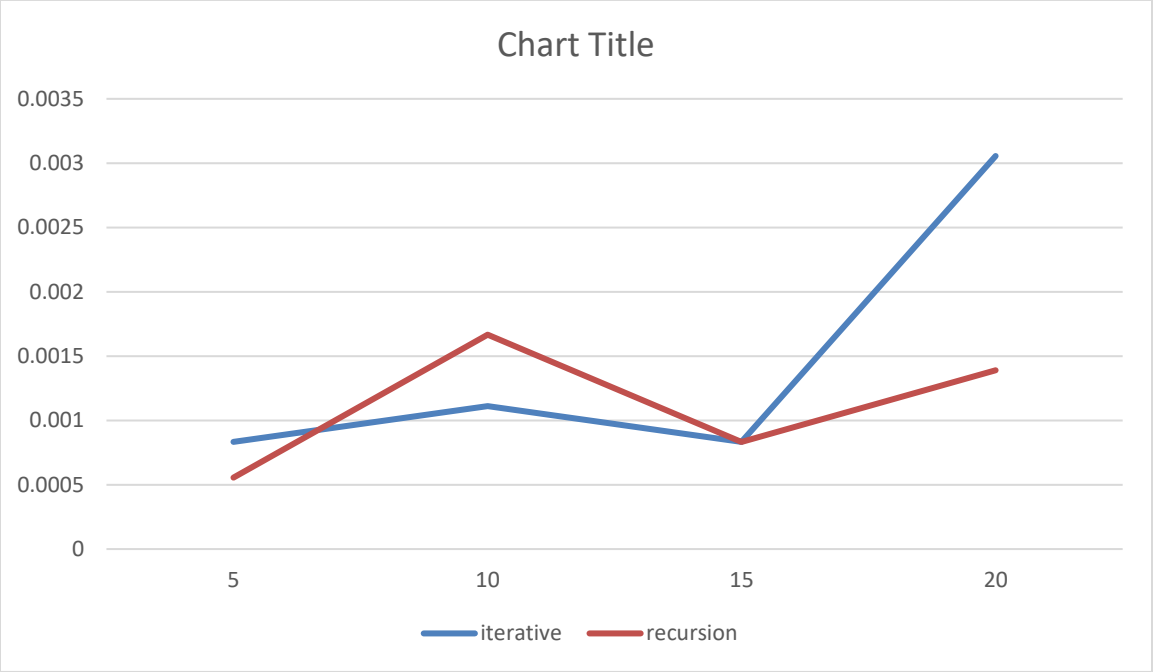
ID: 202110806

THE CODE:

```
#include<iostream>
#include<stdio.h>
#include<time.h>
#include<unistd.h>
using namespace std;




long long iterativefact(int n){
	long long fact=1;
	for(int i=1;i<=n;i++)
	{
		if(n==1)
		return 1;
		else
		fact=fact*i;

	}
	return fact;
}
```

```
long long recursionfact(int n){
        if (n==1)
        return 1;
        else

        return n*recursionfact(n-1);
}

main()

{
        time_t startTime= time(NULL);

int n;
cout<<"please enter a positive number:"<<endl;
cin>>n;

long long fact=recursionfact(n);
//long long fact=iterativefact(n);
cout<<"the factorial is:"<<fact<<endl;

sleep(1);
time_t endTime= time(NULL);
double timeTaken=endTime-startTime;
cout<<"the execution time:"<<(timeTaken)/3600<<endl; }
```

integer	factorial	iterative	recursion
5	120	.000833333	.000555556
10	3628800	.000111111	.00166667
15	2004310016	.000833333	.000833333
20	2432962008176640000	.000305556	.0013889



Stack overflow observation:

When I try the number 10000 or any large number in both methods it will not give me a true result, the stack will be overflow because the number is very large.

Discussion:

In this program I used two different method to calculate the factorial of various number, I used the iterative and recursion methods both of these two methods gave me a true result but I noticed that the iterative way is faster than recursion method when I try it with bigger numbers because the recursion method call itself many time and the time will be wasted on calling the function .