14. Write C programs that demonstrate the mathematical analysis of non-recursive and recursive algorithms

PROGRAM:

```
def factorial_iterative(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
def fibonacci_recursive(n):
    if n <= 1:
        return n
    else:
        return fibonacci_recursive(n-1) + fibonacci_recursive(n-2)
a=10
print(factorial_iterative(a))
for i in range(1,a+1):
    print(fibonacci_recursive(i)," ",end="")
```

OUTPUT:

```
PS C:\Users\chall\OneDrive\Desktop\DAA> & C:/Users/chall/AppData/Local/Programs/Python/Python312/python.exe
"
3628800
1  1  2  3  5  8  13  21  34  55
PS C:\Users\chall\OneDrive\Desktop\DAA>
```

TIME COMPLEXITY:

Time complexity for the above code is

O(n)+O(2n)