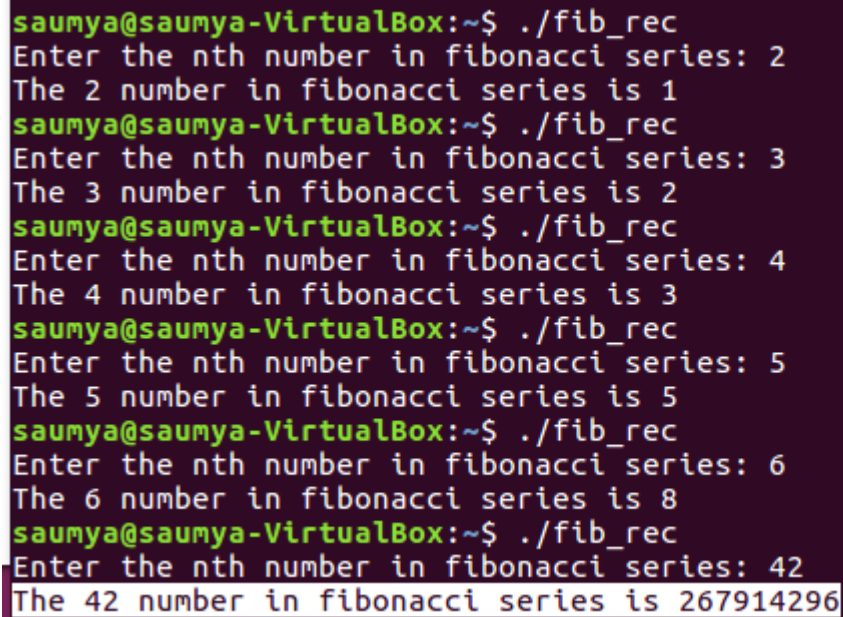


The C function for fibonacci series using recursive calls has been attached as "**fib_rec.c**"

The 42th number in Fibonacci series?

The 42th number in Fibonacci series is 267914296

PFB the screenshot of the result achieved:



```
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 2
The 2 number in fibonacci series is 1
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 3
The 3 number in fibonacci series is 2
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 4
The 4 number in fibonacci series is 3
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 5
The 5 number in fibonacci series is 5
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 6
The 6 number in fibonacci series is 8
saumya@saumya-VirtualBox:~$ ./fib_rec
Enter the nth number in fibonacci series: 42
The 42 number in fibonacci series is 267914296
```

Why does your program take as long as it does?

In the recursive method we have:

$$T(1) = 2 T(0)$$

$$T(2) = (2^2) T(0)$$

$$T(3) = (2^3) T(0)$$

.

.

$$T(n) = (2^n) T(0)$$

=> Increasing the input by 1 is enough to multiply the algorithm's running time (by 2). So, the larger the input value is, the more number of multiplications will increase the algorithm's running time

What is the order of the algorithm (in k)?

The order of the algorithm is $O(2^n)$

The C function for fibonacci series using recursive calls has been attached as "**fib_big_O.c**"

PFB the result:

```
saumya@saumya-VirtualBox:~$ ./fib_big_0
Enter the nth number in fibonacci series: 3
The 3 number in fibonacci series is 2
saumya@saumya-VirtualBox:~$ ./fib_big_0
Enter the nth number in fibonacci series: 5
The 5 number in fibonacci series is 5
saumya@saumya-VirtualBox:~$ ./fib_big_0
Enter the nth number in fibonacci series: 42
The 42 number in fibonacci series is 267914296
saumya@saumya-VirtualBox:~$
```