

Recursion

→ When a function calls itself, recursion.

```
function Inception () {
    print "Dream";
    Inception ();
}
```

Factorial → ✓

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

$$5! = 120$$

$$n = 5$$

$$\text{ans} = 1$$

$$\text{for } (i=1; i < n; i++) \{$$

$$\quad \quad \quad \text{ans} *= i;$$

$$3! = 6$$

$$3 \times 2 \times 1$$

Recursive call → factorial(5)

Self work →  $24 \times 5 \rightarrow 120$

factorial(4) →  $4 \times 6$

factorial(3) →  $3 \times 2$

factorial(2) →  $2 \times 1$

factorial(1) →  $1 \times 1$

factorial(0) → [Base case] Terminating condition

```
function factorial (x) {
    if x is 0:
        return 1; // Base case
    ans = factorial (x - 1); // Recursive call
    return x * ans; // Self work
}
```

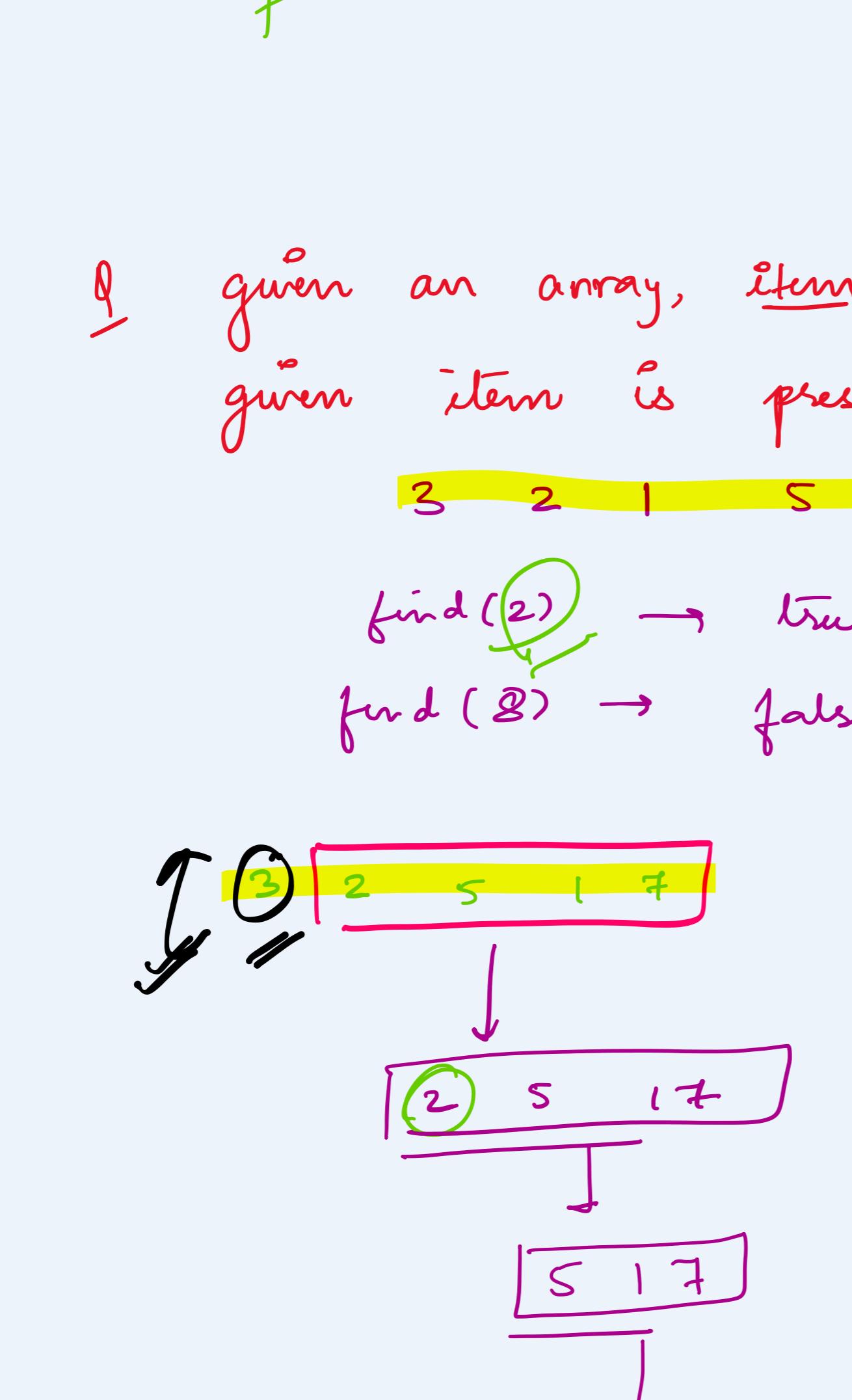
Q Given  $n^{th}$  fibonaccii no,

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

$n = 2, 1$

$n = 9, 34$

$$F_n = F_{n-1} + F_{n-2}$$



Given an array, display the array. (recursion)

- ① self work
- ② recursive call
- ③ base case

display → [3 2 1 5 +]

→ print(0, n)

→ print(1, n-1)

→ print(2, n-2)

→ print(3, n-3)

→ print(4, n-4)

→ Base case

virtual index

3 | 2 | 5 | 1 | 7 |, 0

print(3) → arr, 1

display(arr, 1) → arr, 1

print(2) → arr, 2

display(arr, 2) → arr, 2

print(1) → arr, 3

print(0) → arr, 4

print(5) → arr, 5

print(4) → arr, 6

print(3) → arr, 7

print(2) → arr, 8

print(1) → arr, 9

print(0) → arr, 10

print(5) → arr, 11

print(4) → arr, 12

print(3) → arr, 13

print(2) → arr, 14

print(1) → arr, 15

print(0) → arr, 16

print(5) → arr, 17

print(4) → arr, 18

print(3) → arr, 19

print(2) → arr, 20

print(1) → arr, 21

print(0) → arr, 22

print(5) → arr, 23

print(4) → arr, 24

print(3) → arr, 25

print(2) → arr, 26

print(1) → arr, 27

print(0) → arr, 28

print(5) → arr, 29

print(4) → arr, 30

print(3) → arr, 31

print(2) → arr, 32

print(1) → arr, 33

print(0) → arr, 34

print(5) → arr, 35

print(4) → arr, 36

print(3) → arr, 37

print(2) → arr, 38

print(1) → arr, 39

print(0) → arr, 40

print(5) → arr, 41

print(4) → arr, 42

print(3) → arr, 43

print(2) → arr, 44

print(1) → arr, 45

print(0) → arr, 46

print(5) → arr, 47

print(4) → arr, 48

print(3) → arr, 49

print(2) → arr, 50

print(1) → arr, 51

print(0) → arr, 52

print(5) → arr, 53

print(4) → arr, 54

print(3) → arr, 55

print(2) → arr, 56

print(1) → arr, 57

print(0) → arr, 58

print(5) → arr, 59

print(4) → arr, 60

print(3) → arr, 61

print(2) → arr, 62

print(1) → arr, 63

print(0) → arr, 64

print(5) → arr, 65

print(4) → arr, 66

print(3) → arr, 67

print(2) → arr, 68

print(1) → arr, 69

print(0) → arr, 70

print(5) → arr, 71

print(4) → arr, 72

print(3) → arr, 73

print(2) → arr, 74

print(1) → arr, 75

print(0) → arr, 76

print(5) → arr, 77

print(4) → arr, 78

print(3) → arr, 79

print(2) → arr, 80

print(1) → arr, 81

print(0) → arr, 82

print(5) → arr, 83

print(4) → arr, 84

print(3) → arr, 85

print(2) → arr, 86

print(1) → arr, 87

print(0) → arr, 88

print(5) → arr, 89

print(4) → arr, 90

print(3) → arr, 91

print(2) → arr, 92

print(1) → arr, 93

print(0) → arr, 94

print(5) → arr, 95

print(4) → arr, 96

print(3) → arr, 97

print(2) → arr, 98

print(1) → arr, 99

print(0) → arr, 100

print(5) → arr, 101

print(4) → arr, 102

print(3) → arr, 103

print(2) → arr, 104

print(1) → arr, 105

print(0) → arr, 106

print(5) → arr, 107

print(4) → arr, 108

print(3) → arr, 109

print(2) → arr, 110

print(1) → arr, 111

print(0) → arr, 112

print(5) → arr, 113

print(4) → arr, 114

print(3) → arr, 115

print(2) → arr, 116

print(1) → arr, 117

print(0) → arr, 118

print(5) → arr, 119

print(4) → arr, 120

print(3) → arr, 121

print(2) → arr, 122

print(1) → arr, 123

print(0) → arr, 124

print(5) → arr, 125

print(4) → arr, 126

print(3) → arr, 127

print(2) → arr, 128

print(1) → arr, 129

print(0) → arr, 130

print(5) → arr, 131

print(4) → arr, 132

print(3) → arr, 133

print(2) → arr, 134

print(1) → arr, 135

print(0) → arr, 136