06-01-2026 - Explore data structures such as Stacks, learn about their operations, and use them to solve problems in a variety of domains.

06 January 2026 13:31

=> Iterative / Recursive

=> Recursion :

Type of a method that executes itself.

loops :
for
while
do, while

=> Factorial

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
fact (int x) {          <= Iterative
    for (i=1; i<x; i++){
        x *= x
    }
```

=> fact(n) {

if n == 0

return 1

return n * fact(n-1)

=> fact(5) => 5*24 = 120

if 5 == 0 x

return 5 * fact(4) 24

if 4 == 0 x

return 4 * fact(3) = 6

if 3 == 0 x

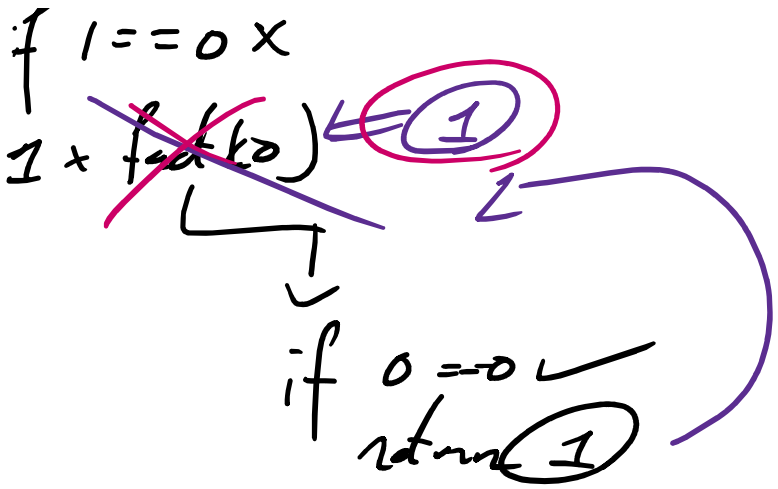
return 3 * fact(2) 2

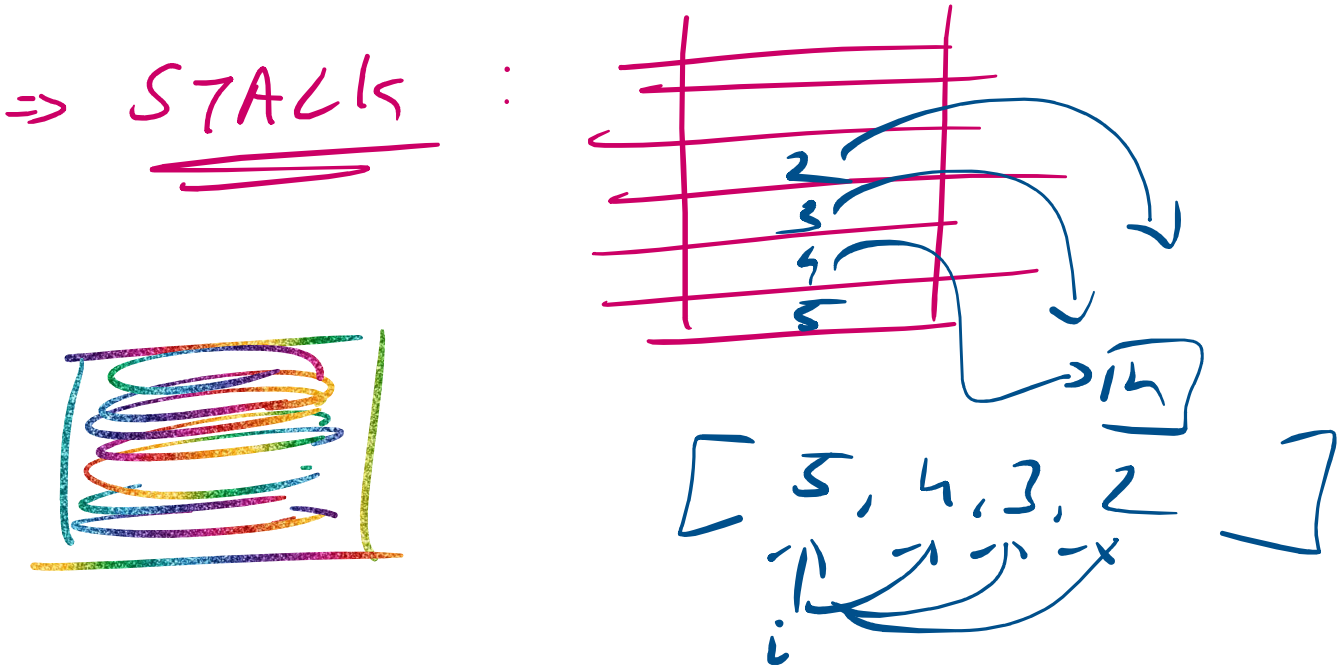
if 2 == 0 x

2 * fact(1)

1

if 1 == 0 x





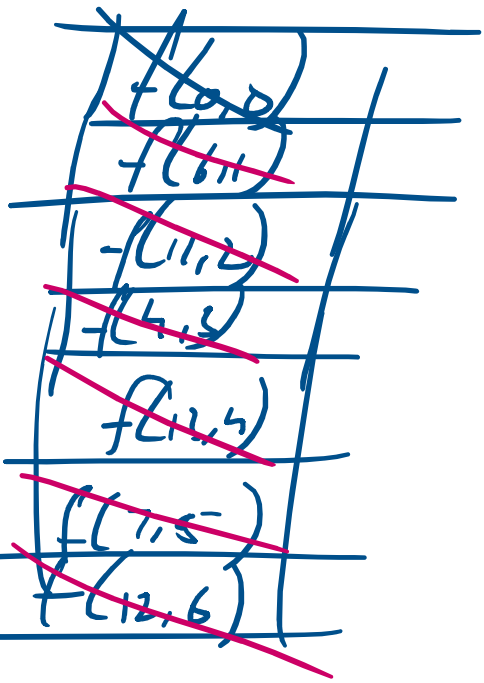
LIFO : Last In First Out

9)

```
f(*arr, n)
    if n <= 0
        return 0
    dec if (*arr % 2 == 0)
```

```
else if (*arr % 2 == 0)
    return *arr + f(*arr+1, n-1)
else
    return *arr - f(*arr+1, n-1)

main() {
    arr = {12, 7, 13, 4, 11, 6}
    n = 6
    f(*arr, 6)
```



=> f(12, 6) = 14

if 6 <= 0 ×

else if (12 % 2 == 0) ✓

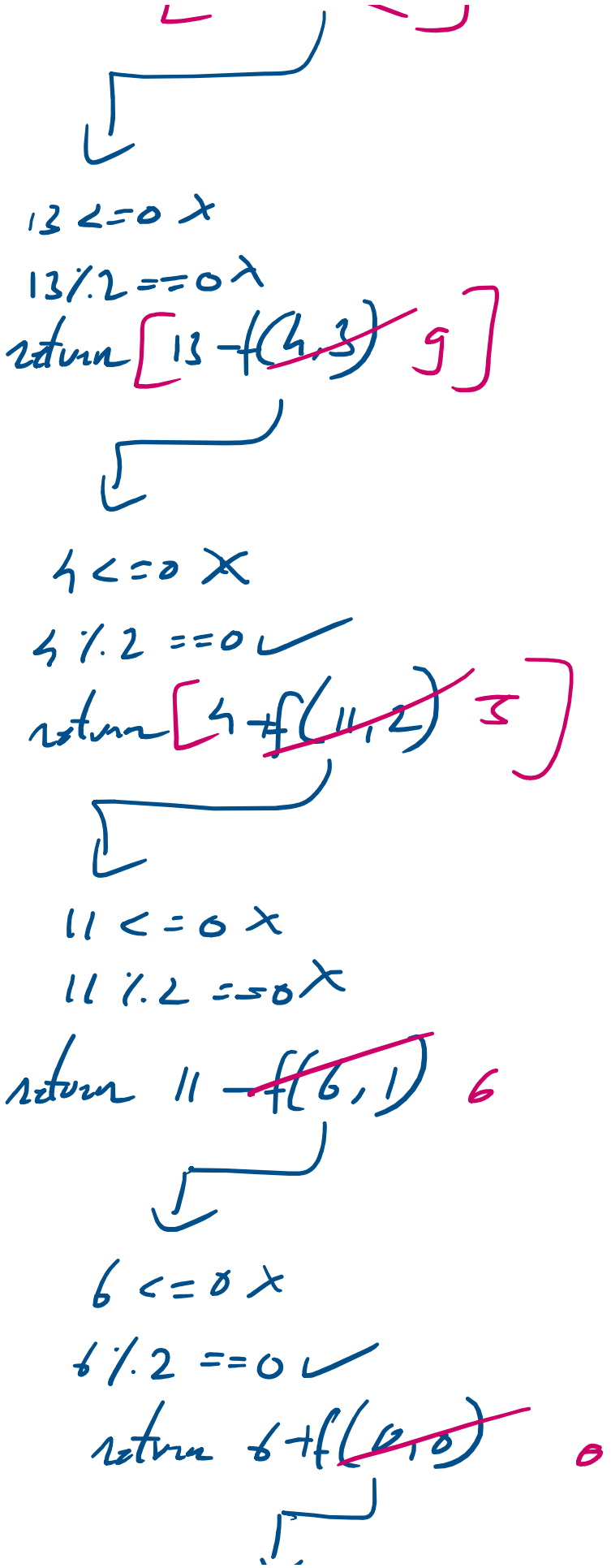
return [12 + f(7, 5)] = 15

3

if 7 <= 0 ×

7 % 2 == 0 ×

return [7 + f(13, 4)] = 4


13 <= 0 X
13 % 2 == 0 ✓
return [13 + f(4,3) 9]
4 <= 0 X
4 % 2 == 0 ✓
return [4 + f(11,2) 3]
11 <= 0 X
11 % 2 == 0 X
return 11 + f(6,1) 6
6 <= 0 X
6 % 2 == 0 ✓
return 6 + f(0,0) 0


0<=0
return 0