# Basics of Recursion

↳ A function calling itself

```
psv print1() {
    syso(1);
    print2();
}
psv print2() {
    syso(2);
    print3();
}
psv print3() {
    syso(3);
    print4();
}
psv print4() {
    syso(4);
}

p s v main () {
    int a = 10;
    print1();
    print4();
}
```

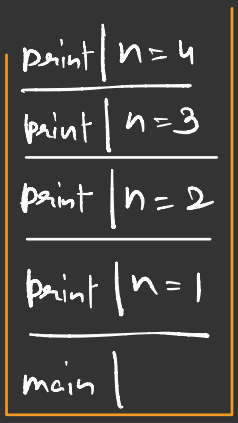
Console

1 2 3 4 4

Function call stack

```
p s void print(int n) {
    sysO(n);
    print(n+1);
}

main() {
  print(1);
}
```




Console

1 2 3 4

# Qwes Print from 1 to n Print Increasing

## Expectation

print(5)

1
2
3
4
5

## Faith

print(4)

1
2
3
4

## Combine

```
b s v print(5) {
    print(4)
    sysо(5)
}
```

3

Generalise

```
b s v print(n) {
    if (n < 1) return;
    print(n-1)
    sysо(n)
}
```

Base case

3

1 2 3 4 5

Ques Print Decreasing ⇒ 5 4 3 2 1

5

Expectation

PD(5)

5
4
3
2
1

Faith

PD(4)

4
3
2
1

Combine

syso(5);
PD(4)

```
psv PD(int n) {
    if(n==0) return;
    syso(n);
    PD(n-1);
}
```

5 4 3 2 1

| PD 0 |
|---|
| PD 1 |
| PD 2 |
| PD 3 |
| PD 4 |
| PD 5 |

# Ques Print Decreasing Increasing

## Expectation

PD1(4)

n=4

4
3
2
1
1
2
3
4

## Faith

PD1(3)

3
2
1
1
2
3

## Combine

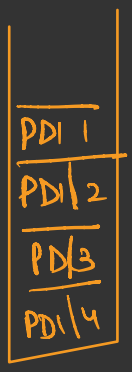
syso(4)
PD1(3)
syso(4)

Generalise

```
p s v PD1(int n){
    if(n==0) return;

    syso(n)
    PD1(n-1)
    syso(n)
}
```

3




4 3 2 1

Query Factorial

n ⇒ n!

4! ⇒ 4 × 3 × 2 × 1 ⇒ 24

Expectation

fact(5)

5 × 4 × 3 × 2 × 1

⇒ 120

Faith

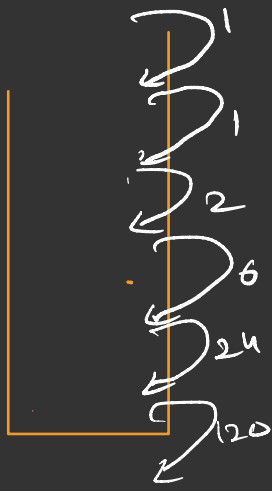
fact(4)

4 × 3 × 2 × 1

⇒ 24

Combine

return 5 * fact(4)




Geradise

```
p s int fact(int n){
    if(n==0) return 1;
return  n * fact(n-1);
}
```

Ques Fibnacci nth

| 0th | 1st | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 ... |

Expectation

fib(6)
→ 3 + 5
⇒ 8

Faith

fib(5) ⇒ 5
fib(4) ⇒ 3

Combine