Recursion 1. Print Decreasing

Ques → n = 5 (given)

o/p → 5, 4, 3, 2, 1

Steps to only recursion.

Faith :-

Expectation :-

Hint: only

for (n){ if (n == 0) return → base case

print(n)

print (n-1).

n=0

```
public static void printDecreasing(int n){
    if(n_==_0){
        return;
    }
    System.out.println(n);        //work
    printDecreasing(n-1);
}
```

dry run

Important:
- Dry Run
- Memory Stack
- Output
- every line

h/w

Recursion

PM I

called

faith

take Recursion for

Expectation

n+1

P(5)

P(4) } faith

Expectation?

Print Increasing

Recursion

Call

func( )

func( )

Hint

Recursion

Call

Call

f/E

Stack

Code

Dry Run

Stack line

return

that yeh.

Print Increasing.

PI(n)

f. if( )

}

PI(n-1)

f/E

i.e

Code

Dry

Stack Diagram

n=5

n=5

n=3

n=2

n=1

1

2

3

4

5

n=3

1

2

3

fn(n)

1. if (n==0) return

2. fn(n-1)

3. fn(n-1)

n=2

n=1

1

2

3

Print Increasing Decreasing

Sample Input

1

Sample Output

3
2
1
1
2
3

8

2

4

1

7

3

3

call ( ) → n-1

3

F/E

SD

CODE

DRY

```
public static void pdi(int n){
    if(n == 0){
        return;
    }
    System.out.println(n);
    pdi(n-1);
    System.out.println(n);
}
```

n=3

3
2
1
1
2
3

Factorial

4! — 720

3! — 120

2! → 6

4! = 6×5×4×3×2×1

n! = (n×(n-1)×(n-2)×(n-3)...1

6! = 6 × 5! = 6 × 120 = 720

5! = 5 × 4! = 5 × 24 = 120

4! = 4 × 3!

F/E

for n! = 6 × 5! faith = 720

(n-1)! faith = n!

exp.

return 1;

p is always 1

2! = 2

3! = 3×2! = 6

```
public static int factorial(int n){
    if(n == 0){
        return 1;
    }
    int recans = factorial(n-1);
    int myAns = n * recans;
    return myAns;
}
```

myAns = factorial(n)

out = 120

print(out);

n=5

n=4

n=3

n=2

n=1

1! = 1

n = 0

1

1

2

6

24

120

n=1

1! = 1

n=0

1

1

n=1

1! = 1

factorial(n-1)

n=0

1

5!

n=5

```
public static int factorial(int n){
    return (n == 0) ? 1 : n*factorial(n-1);
}
```

if (n==0) return 1;

n! = n × (n-1) × (n-2) ... 1

condition ? T : F ;

for (int i = 1; i<=n; i++)

{ fact = fact * i }

return fact

120

5 × 4!

4 × 3!

3 × 2!

2 × 1!

1