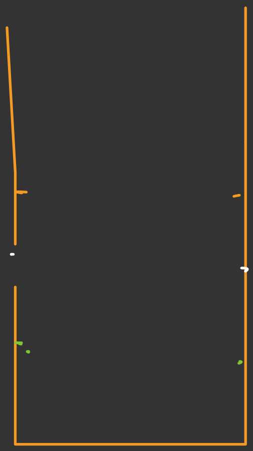
Recursion

- → Funtions
- → Memory management (functions)

```
p   s   void   Print (int   n) {
    syso ( n);
    Print2() ;
 }


p  s  void  Print2() {
     syso (Hi);
 }  ←


main()  {
     int  a = 5;
     syso (a);
    Print (4),

         argument

     Print 2 ();
    Syso ("End Main")
}  ←
```

Console

```
5
4
Hi
Hi
End Main
```



Stack

```
p s   void   print1(int n) {
        syso(n);
        print2(2)
}

p s void   print2(int n)
        syso(n)
        print3(3);
}
p s void print3(int n)
      syso(n)
      print4(4)
 }

p s void print4(int n)
      syso(n)
      print5(5);
}

p s void print5(int n)
      syso(n)
 }

  main() {
   syso(Main Starts)
   Print1(1);
}  syso(Main Ends)
```

Console

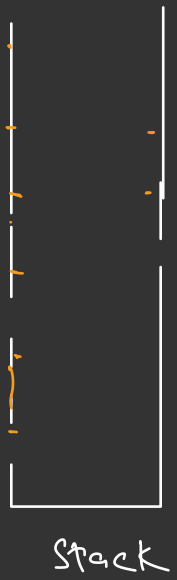
```
Main Starts
1
2
3
4
5
Main Ends
```

stack

```
p     s     void   print (int n) {
    if(n == 5) return;

    syso (n);
    print (n+1);
}
main() {
   Syso ("Main starts");
    print (1); <—

}
```

Console

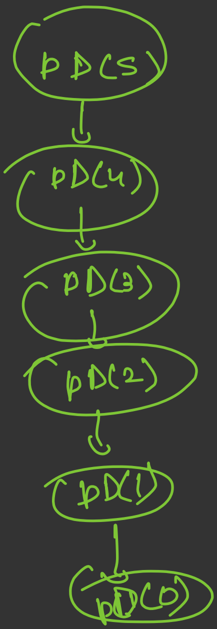
```
Main Starts
1
2
3
4
```



}

```
P s    void    print Decreasing (int n){
  if(n = =0) {

      return;
  }

   Syso (n); ⇽
   print Decreasing (n-1);
}
```

Recursion tree



printDecreasing(5)

5

4

3

2

1

P 4

P 3

P 2

P 1

P 0

Ques Print Decreasing

n = 5

Expectations	Faith	Combine
5	4	syso(5); print Decreasing(4)
4	3	
3	2	
2	1	
1		

```
syso(n)
print Decreasing(n-1);
```

Ques PrintIncreasing

1 2 3 . . . n

n = 5

Expectation	Faith	Combine

1	1	PI(4)
		syso(5);

2	2	
3	3	
4	4	
5		

```
PrintIncreasing(n-1);
syso(n);
```

```
P1 (n) {
    if(n==1) {
        syso(1)
        return;
    }

    P1 (n-1);

    syso(n)

}
```

```
if(n==0)
    return;
```

P1	1
P1	2
P1	3
P1	4
P1	5

Console

Ques Print DecreasingIncreasing

Expectation

n=5

```
5
4
3
2
1
1
2
3
4
5
```