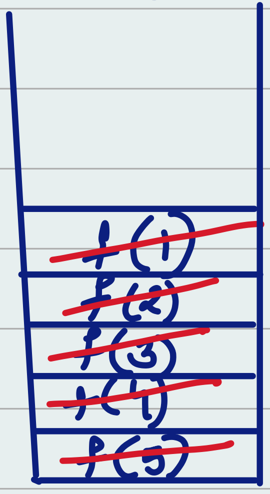
Q) N = 5, Print all the numbers from N till 1 : 5, 4, 3, 2, 1

$\underset{5}{f(5)} \to \underset{4}{f(4)} \to \underset{3}{f(3)} \to \underset{2}{f(2)} \to \underset{1}{f(1)}$



```
fun (n)
    prints (n)
    fun (n-1)
```

base condition if n == 0

```
        return;
```

```
{ psvm (   ) {
    fun (5);
}
    static void fun (int n) {
        if (n == 0) {
            return;
        }
        System out. println (n);
        fun (n-1);
    }
}
```

So for printing 1 to 5 using recursion calls fun (n·6)

& if $(n==6)$
return;

also fun' would be fun$(n+1)$

Q) factorial of a number:

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

$\text{Fact}(5) = 5 * \text{Fact}(4)$
$= 5 * 4 * \text{fact}(3)$
↳ $3 * \text{fact}(2)$
↳ $2 * \text{fact}(1)$

$F(N) = N * F(N-1)$

Tree:

$F(5) = 5 * F(5-1) = 120$
$F(5) = 5 * F(4)$ 24
$F(4) = 4 * F(4-1)$ 6
$F(3) = 3 * F(3-1)$ 2
$F(2) = 2 * F(1)$ 2
$F(1) = 1$
Return;

Base Case $F(1) = 1$

Code:

```
  psvm (    ){
     int ans = 5,
     system out-println (factorial (ans));
  }
  static int factorial (int n){
        if (n <= 1){
              return 1,
        }
          return n * factor(n-1);
  }
```

Q) Sum of digits

N = 1342 Ans = 1+3+4+2 = 10

remainder = N%10

N = N/10

1+ sumof digitsFun((342)

3 + F (42)

$\frac{134\boxed{2}}{10} = 2 \rightarrow \frac{13\boxed{4}}{10} = 4 \rightarrow \frac{13}{10} \Rightarrow 3 \rightarrow \frac{1}{10} \rightarrow 1$

$F(N) = F(N/10) + (N\%10)$

$F(N) = F(N/10) +$ remainder

$F(1342)$ Complexity : $O\log(N)$

return $\boxed{2 + F(134)}$

$\xrightarrow{}$ return $(4 + F(13)))$

Code in mac

$n--$ Vs $--n$ // $n = n-1$

$n--$) first pass for $n = 5$

↓ subtract first than pass for $n = 5$

Q:] Reverse a number using recursion.

$N = 1$ to 5 return 5 to 1.

or $N = \underbrace{1825}_{f}$ to 5281

$1825 \longrightarrow 5 + f(182)$

↓

$2 + f(18)$

↓

$8 + f(1) \rightarrow$ return base condition

take the last element put it in the first.

$\boxed{F(N) = (N \% 10) \quad F(N/10)}$

1)

```
fun (n){ void
    if(n ==0){
        return;
    }
    remainder=n%10;
    sum=sum*10+rem;
    fun(n/10);
}
```

1825

$5+f(182)$

$(5*10+2)+f(18)$

$52+f(18)$

$521 \ +f(8)$

2) $N = 12345$

$5*10000 + 1234$

$5*10^4+f(1234)$

$4\times10^3+f(123)$

$3\times100+f(12)$

$2\times10+f(1)$ → return

$F(N, \text{arguments}) =$

$\underbrace{(N\%10)}_{\text{rem}}10^{\text{arguments}-1} +$

$F(N/10, \text{arguments}-1)$

Here helper function is created.

In code line

```
return remainder *(long)(Math.pow(10,digits-1))
    + helper(n/10, digits-1);
```

Q) Palindrome or not.

$N = 12321$

~~start~~ ~~s~~ se ~~e~~ ~~end~~

1 2 3 2 1

How can we solve this through recursion

Code in the mac through recursion.

Q) Count number of Zeroes in a number.

$N = 30204$ $\boxed{Ans = 2}$

① If we take count inside argument

$F(N, 0) \rightarrow$ If digit = 0

Imp to count

$F(N/10, C+1)$

Ex: $\frac{100}{10} = \overset{2}{10}/10 = 1$

$0+1 = 1$

↳ $1+1 = 2$

else $F(N/10, C)$

$\frac{30204}{10} = \frac{3020}{10}$

$= C+1 = 0+1$

$= 1$

$\frac{302}{10} = 30$

$\frac{30}{10} =$

↳ $1+1 = 2$

$F(30204, 0)$ ↓ 2

$F(3020, 0)$ 2

$F(30, 1)$ 2

F (30 , 1) 2

F (3 , 2) 2

F (0, 2) return ans

```
public class Count_0 {
    psvm(){
      Sout (count (30210));
  }
   static int count (int n) {
        return helper( n, 0);
   }
// special patterns, how to pass a value
to above calls.

 static int helper( int n, int count){
       if (n==0){
          return count;
       }
        int remainder = n%10;
           if (remainder ==0){
             return helper( n/10, count+1)
    return helper(n/10, count);
}
```

★ Special example to return same value to above function calls

---

Q) Given an integer num, return the number of steps to reduce it to zero

In one step, the current number is even, you have to divide it by 2, otherwise subtract one from it

Input num = 14
output = 6

1) 14 is even, divide by 2 obtain 7.
2) 7 - 1 = 6 → second step.
3) 6 / 2 = 3 → 3rd step
4) 3 - 1 = 2 → 4th step
5) 2/2 = 1 → 5th step
6) 1 - 1 = 0 → 6th step
↳ obtain

stop //

helper function(We need to pass these values in recursion calls, that is why we need to put it in the argument (helper function)

code:

```
class Solution {
        public in numberOfSteps(int num){
            return helper(num, 0);

    static int helper (int num, int steps){

        if (num = = 0){
            return steps;
        }
        if (num % 2 == 0){
            return helper(num/2,
                          count+1);
        }
        return helper(num-1,steps+1);
        }
}
```