* Problems To Solve Using Recursion

1. Print all numbers from 1 to n..

def print\_number(number):  
 if number==0:  
 return  
 print\_number(number-1)  
 print(number)  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 n=int(input())  
 print\_number(n)

1. Print all numbers from n to 1..

def print\_number\_re(number):  
 if number==0:  
 return  
 print(number)  
 print\_number\_re(number-1)  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 n=int(input())  
 print\_number\_re(n)

1. Sum of all numbers..
2. def sum\_number(number):  
    if len(number)==1:  
    return number[0]  
    p=number[0]+sum\_number(number[1:])  
    return p  
     
     
     
     
   if \_\_name\_\_=="\_\_main\_\_":  
    n=[1,2,3,4,5]  
    v=sum\_number(n)  
    print(v)

4.Find max number from a list…

def max\_number(number):  
 if len(number)==1:  
 return number[0]  
 first\_num=number[0]  
 max\_value=max\_number(number[1:])  
 if first\_num>max\_value:  
 return first\_num  
 return max\_value  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 n=[1111,2222,10883,234,245]  
 v=max\_number(n)  
 print(v)

5.Find min number from a list…..

def min\_number(number):  
 if len(number)==1:  
 return number[0]  
 first\_num=number[0]  
 min\_value=min\_number(number[1:])  
 if first\_num<min\_value:  
 return first\_num  
 return min\_value  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 n=[12,141,2,3,24,45]  
 v=min\_number(n)  
 print(v)

6. Write a function for mutliply(a, b), where a and b are both positive integers, but you can only use the + or − operators.

def multiply\_num(a,b):  
 if b==0:  
 return 0  
 if b==1:  
 return a  
  
 return a + multiply\_num(a,b-1)  
  
  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 a=4  
 b=3  
 v=multiply\_num(a,b)  
 print(v)

7. Reverse a list without using a loop..

def reverse\_list(list1):  
  
 if len(list1)==0:  
 return []  
 return [list1[-1]]+ reverse\_list(list1[:-1])

8. Find factorial of a number..

def factorial(fact):  
   
 if fact==0:  
 return 1  
  
 return fact \* factorial(fact-1)  
  
  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 a=5  
  
 v=factorial(a)  
 print(v)

9. Check if a string is palindrome or not…

def palindrome(text):  
  
 if len(text)==0 or len(text)==1:  
 print("Palindrome")  
 else:  
 if text[0]==text[-1] :  
 palindrome(text[1:-1])  
 else:  
 print("Not Palindreom")

10. Power of 2 - [https://leetcode.com/problems/power-of-two/](https://l.facebook.com/l.php?u=https%3A%2F%2Fleetcode.com%2Fproblems%2Fpower-of-two%2F%3Ffbclid%3DIwAR3w-5qHRKyfKxHXa3odrNDUbg3rShlT1DZ42raVFhAXW0Db5r6Iq8HNcNc&h=AT2oXEqh2nA9Weo-0jkNpxjM7KUALx96btfdBb5iFSTsuOHeqT2OdlLAW0vE4XuTFAHvG5EtEEXyFz_wgyQbYHPUm1elF9CRzl7XPWKc6cS0lmBqFOZWoqWSj_d6c-NNvSJW&__tn__=-UK-R&c%5b0%5d=AT0x5DkorqS7soS0JwSBG5QwW5fYWBAQdkuD3xef0bnT4QB-FXOw5ciukCOjtO7MMvjv4UNJ7ReWVNM17i_a59qUmSEM8qrVdvl8Kj9useF0L_SNSjuZpEfPBoYnXT5v3njH3TTMG7UqUbsm2nlv0Raprh-XmFV8Nz5Kpsd6zcd714Yb5UW4YJLd3oR9GZ8rOmXWZVP_rOR05iLR9OJEOOE)

def power\_num(number,power):  
 if power==0:  
 return 0  
 if power==1:  
 return number  
  
 return number \* power\_num(number,power-1)  
  
  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 number=int(input())  
 power=int(input())  
 v=power\_num(number,power)  
 print(v)

11. [https://leetcode.com/problems/power-of-three/](https://leetcode.com/problems/power-of-three/?fbclid=IwAR2-6dJPWZFYAOl3yCgWa3KoJSll_B3JbNmL8sBFW_3ObHdCsB4LQVRm1QA)

def power\_num(number,power):  
 if power==0:  
 return 0  
 if power==1:  
 return number  
  
 return number \* power\_num(number,power-1)  
  
  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 number=int(input())  
 power=int(input())  
 v=power\_num(number,power)  
 print(v)

12. [https://leetcode.com/problems/power-of-four/](https://leetcode.com/problems/power-of-four/?fbclid=IwAR0J_OxljYXA8F2m4Aqb8aMjn1y7WfoHaLkeFl5bAxt2T3Pz98AidTZdtDs)

def power\_num(number,power):  
 if power==0:  
 return 0  
 if power==1:  
 return number  
  
 return number \* power\_num(number,power-1)  
  
  
  
  
  
if \_\_name\_\_=="\_\_main\_\_":  
 number=int(input())  
 power=int(input())  
 v=power\_num(number,power)  
 print(v)

13. [https://leetcode.com/problems/fibonacci-number/](https://leetcode.com/problems/fibonacci-number/?fbclid=IwAR2-6dJPWZFYAOl3yCgWa3KoJSll_B3JbNmL8sBFW_3ObHdCsB4LQVRm1QA)

*def feb(n):  
 if n==0:  
 return 0  
 if n==1:  
 return 1  
 return feb(n-1)+feb(n-2)*