

fibonacci using recursion

```
In [1]: def recur_fibo(n):  
        if n<=1:  
            return n  
        else:  
            return(recur_fibo(n-1)+ recur_fibo(n-2))  
n=10  
if n<= 0:  
    print(" enter a integer")  
else:  
    print("Fibonacci sequence:")  
    for i in range(n):  
        print(recur_fibo(i))
```

Fibonacci sequence:

0
1
1
2
3
5
8
13
21
34

armstrong number

```
In [ ]: n=int(input("enter a number"))  
num = n  
digit, sum = 0, 0  
length = len(str(num))  
for i in range(length):  
    digit = int(num%10)  
    num = num/10  
    sum += pow(digit,length)  
if sum==n:  
    print("Armstrong")  
else:  
    print("Not Armstrong")
```

gcd using recursion

```
In [ ]: def gcd(a, b):
        if a == b:
            return a
        elif a < b:
            return gcd(b, a)
        else:
            return gcd(b, a - b)
a = 25
b = 45
print(gcd(a, b))
```

asce and desce order in one dimensionl array

```
In [ ]: def printOrder(arr,n) :
        arr.sort()
        i = 0
        while (i< n/ 2 ) :
            print( arr[i])
            i = i + 1
        j = n - 1
        while j >= n / 2 :
            print(arr[j])
            j = j - 1
arr = [5, 4, 6, 2, 1, 3, 8, 9, 7]
n = len(arr)
printOrder(arr, n)
```

largest element in an array

```
In [ ]: a = [10, 89, 9, 56, 4, 80, 8]
max_element = a[0]
for i in range(len(a)):
    if a[i] > max_element:
        max_element = a[i]
print (max_element)
```

factorial using recursion

```
In [ ]: def recur_factorial(n):
        if n == 1:
            return n
        else:
            return n*recur_factorial(n-1)
number = int(input("enter a number "))
print("The factorial of", number, "is", recur_factorial(number))
```

copy string

```
In [ ]: def myCopy(s1,s2):  
        for i in range(len(s1)):   
            s2[i]=s1[i]  
        return "".join(s2)  
s1=list("teju")  
s2=[""]*len(s1)  
print(myCopy(s1,s2))
```

reverse a string

```
In [ ]: def reverse(string):  
        if len(string) == 0:  
            return string  
        else:  
            return reverse(string[1:]) + string[0]  
a = str(input("Enter the string to be reversed: "))  
print(reverse(a))
```

prime number

```
In [ ]: num = 10  
flag = 0  
for i in range(2,num):  
    if num%i==0:  
        flag = 1  
        break  
if flag == 1:  
    print('Not Prime')  
else:  
    print("Prime")
```

prime number using recursion

```
In [ ]: def Prime_Number(n, i=2):  
        if n== i:  
            return True  
        elif n % i== 0:  
            return False  
        return Prime_Number(n, i + 1)  
n = 9  
if Prime_Number(n):  
    print( n, "is Prime")  
else:  
    print(n, "is not a Prime")
```

palindrome or not

```
In [ ]: input_string = 'teju'
rev = input_string[::-1]
if input_string == rev:
    print(" is Palindrome")
else:
    print(" is not Palindrome")
```

time complicity:

```
In [ ]: fibonacci series:  $O(2^n)$ 
armstrong number:  $O(n)$ 
gcd:  $O(\log n)$ 
largest element:  $O(n)$ 
factorial:  $O(n)$ 
copy string:  $O(m)$  because length is used
reverse:  $O(n)$ 
prime number:  $O(\sqrt{n})$ 
palindrome:  $O(n)$ 
```