

Lab Sheet 3

Name : Anuvind MP

Roll no : AM.EN.U4AIE22010

1. Print the sum of the first N natural numbers.

```
def sum_(n):  
    if n == 1:  
        return 1  
    else:  
        return n + sum_(n-1)  
  
sum_(5)
```

15

Page ( )

i) def sum\_(n):  $\rightarrow T(n)$   
    if  $n == 1$ :  $\rightarrow 1$   
        return 1  $\rightarrow 1$   
    else:  
        return  $n + \text{sum\_}(n-1) \rightarrow T(n-1)$

•  $T(n) = \begin{cases} T(n-1) + 1, & n > 1 \\ 1, & n = 1 \end{cases}$

~~case 2~~ case 2:  $\Rightarrow O(n * f(n))$   
 $\Rightarrow O(n * 1) = \underline{\underline{O(n)}}$

## 2. Print the product of the first N natural numbers.

```
def prdct(n):  
    if n == 1:  
        return 1  
    else :  
        return n*prdct(n-1)  
  
prdct(5)
```

120

2) def prdct(n):  $\rightarrow T(n)$

if n==1:  $\rightarrow 1$

return 1

else :

return n\*prdct(n-1)  $\rightarrow T(n-1)$

$$T(n) = \begin{cases} T(n-1) + 1, & n > 1 \\ 1, & n = 1 \end{cases}$$

case 2  $\Rightarrow$

$$TC = \underline{\underline{O(n)}}$$



$$T(n) = T(n-2k) + 2^k - 1$$

$$= T(n-2k) = T(2) = 1$$

$$n-2k = 2$$

$$n = 2k + 2 \quad ; \quad k = \frac{n-2}{2}$$

$$T(n) = 1 + 2^{\frac{n-2}{2}-1} = 2^{\frac{n}{2}-1} = 2^{\frac{n}{2}} / 2 = 2^{\frac{n}{2}}$$

$$\text{i.e., } T(n) = 2^{\frac{n}{2}} ; \quad O(2^{\frac{n}{2}})$$

## 4. Calculate $x^y$ .

```
def power(x,y):
    if y == 0:
        return 1
    elif y == 1:
        return x
    else:
        return x*pow(x,y-1)
```

```
power(2,10)
```

4) def power(x,y):  $\rightarrow T(n)$

```

    if y==0:
        return 1
    elif y==1:
        return x
    else:
        return x * power(x,y-1)  $\rightarrow T(n-1)$ 

```

$T(n) = \begin{cases} 1, & n=0 \\ T(n-1)+1, & n>1 \end{cases}$

case 2:  
 $TC = O(n)$

## 5. Print the first N natural numbers.

```

def natural(n):
    if n == 0:
        return False
    else:
        natural(n-1)
        print(n, end = ", ")

```

natural(4)

1, 2, 3, 4,



5) `def natural(n):`  $\rightarrow T(n)$   
     `if n==0:`  
         `return False` }  $\rightarrow 1$   
     `else:`  
         `natural(n-1)`  $\rightarrow T(n-1)$   
         `print(n, end = ", ")`  $\rightarrow 1$

$$T(n) = \begin{cases} 1, & n=0 \\ T(n-1), & n>0 \end{cases}$$

case 2:  $O(n * f(n))$   
 $T = O(n)$

6. Print the first N natural numbers in reverse order.

```
def naturalrev(n):
    if n == 0:
        return False
    else:
        print(n, end = ",")
        naturalrev(n-1)

naturalrev(4)
```

4,3,2,1,

6)  $T(n) = \begin{cases} 1, & n=0 \\ T(n-1)+1, & n>0 \end{cases}$   
     case 2 :  $O(n * f(n))$   
 $T = O(n)$

## 7. Find the GCD(HCF) of two numbers.

```
def GCD(a,b):  
    if b == 0:  
        return a  
    else:  
        return GCD(b, a%b)
```

GCD(10,20)

10

7) def GCD(a,b):  $\rightarrow T(n)$

if  $b == 0$ :  
    return a } — 1

else :

    return GCD(b, a%b)  $\rightarrow T(n/2)$

$$T(n) = \begin{cases} 1, & b=0 \\ T(n/2) + 1, & b>0 \end{cases}$$

~~$T(n)$~~  —  $T(n) = T(n/2) + 1$

$$a=1 ; b=2 ; k=0 ; p=0$$

$$\log_b a = \log_2 1 = 0$$
$$k=0$$

ie, case 2 ;  $\log_b a = k$

(i)  $p > -1$

$$O(n^k \log^{p+1} n)$$

$$TC \Rightarrow O(\log n)$$

## 8. Print the elements of an array.

```
def printArray(arr):  
    if arr == []:  
        return  
    else:  
        print(arr[0], end=' ')  
        printArray(arr[1:])  
  
arr = [1,2,3]  
printArray(arr)
```

1 2 3

8) def printArray(arr):  $\rightarrow T(n)$   
    if arr == [] : {  $\rightarrow 1$   
        return  
    else :  
        print(arr[0], end=' ')  $\rightarrow 1$   
        printArray(arr[1:])  $\rightarrow T(n-1)$   
  
     $T(n) \rightarrow T(n-1) + 1$   
     $\therefore$  case 2:  $O(n * f(n))$   
    TC  $\Rightarrow \underline{O(n)}$



## 9. Print the elements of an array in reverse order.

```
def revPprintArray(arr):  
    if arr == []:  
        return  
    else:  
        revPprintArray(arr[1:])  
        print(arr[0], end = " ")  
  
arr = [1,2,3]  
revPprintArray(arr)
```

3 2 1

q)  $T(n) = T(n-1) + 1$   
∴ case 2:  
 $O(n * f(n))$   
∴  $T.C = \underline{\underline{O(n)}}$

## 10. Reverse a given number.

```
def reverse(n, rev=0):  
    if n == 0:  
        return rev  
    x = n % 10  
    rev = rev * 10 + x  
    return reverse(n // 10, rev)  
reverse(1234)
```

✓ 0.0s

4321

10) def reverse(n, rev=0):  $\rightarrow T(n)$

if n==0: }  
return rev

x = n/10 — 1

rev = rev \* 10 + x — 1

return reverse(n/10, rev)  $\rightarrow T(n/10)$

•  $T(n) = T(n/10) + 1$

a = 1 ; b = 10

$\log_b a = \log_{10} 1 = 0 + 0$

case: 2 ; (i)  $p > -1$

~~$O(n^{\log_{10} 10})$~~   ~~$T(n) = O(n)$~~

$\Rightarrow n^{\log_{10} 10} = \log n$

$k = \log_{10} n + 1$  where k is no. of digits

$\therefore T(n) = O(k) = O(\log_{10} n + 1) = O(\log_{10} n)$

## 11. Check if an array is sorted or not.

```
def check(arr):
    if len(arr) < 2:
        return True
    elif arr[0] > arr[1]:
        return False
    else:
        return check(arr[1:])
```

```
arr = [1,2,4,3]
arr2 = [1,2,3]
print(check(arr))
print(check(arr2))
```

✓ 0.0s

False

True

ii) def check(arr):  $\rightarrow T(n)$

if len(arr) < 2: } — 1  
return True

elif arr[0] > arr[1] } — 1  
return False

else:

return check(arr[1:])  $\rightarrow T(n-1)$

$$T(n) = \begin{cases} T(n-1) + 1, & n > 1 \\ 1, & n = 1 \end{cases}$$

case 2:  $O(n \cdot f(n))$

TC =  $O(n)$