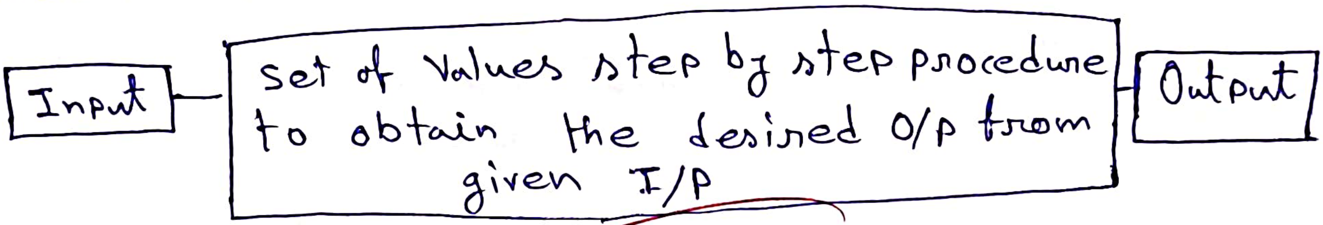
Assignment- 8A D:02.04.25

~~impost~~ ~~aro~~

Sorting in Python:

It's a fundamental operation in comuter science and programming. It involves arraenging data in a perticular order, Typically ascending or decending. Sorting maker searching, processing and analyzing data with more efficiency. In Python programming language it provides various sorting technieques including builtin sorting methods and manual implementation of well known algorythm.

[Note: Algorythm: An algorythm is a finite sequence of well defined instruction on a set of rules designed to accomplish a specific task or solve a problems. An Algorythm is used to solve problems perform calculation or automate task by following step by step procedures that leads to a desired outcome.]


Input
set of Values step by step procedure to obtain the desired O/p from given I/P
Output

Types of Sorting in Python:

1. Builtin In sorting method -

```
Input: numbers  = [1,7,9, 3, 5,6]
       new_num  = sorted (numbers)
       rev_num  = sorted (numbers, reverse = True)
       print (new_num)
       print (rev_num)

Output: [1,3,4, 5,6, 7,8 9]
        [9, 7, 6, 5,4,3,1]
```

Input:

```
numbers = [1,7,9,3,6,4]
numbers.sort()
print(numbers)
numbers.sort(reverse=True)
print(numbers)
```

Output:

```
[1, 3, 4, 6, 7,9]
[9, 7, 6, 4, 3, 1]
```

[Note: 1st method doesn't change or modifies the list and in the 2nd method modifies the original list]

2. Sorting using custom Key:

Input:

```
words = ["Kiwi", "Oak", "Apple", "Banana", "Orange"]
Sorted_words = Sorted(words, Key = len)
print(Sorted_words)
```

Output: ['Oak', 'Kiwi', 'Apple', 'Banana', 'Orange']

3. Manual sorting algorythms:

Python allows implementing sorting algorythms manually

i. Bubblesort Algorythm:

It's a simple comparison based algorythm where adjusent elements are repeatedly swapped if they are in wrong order, This process continuous until the entire list is sorted. Bubblesort algorythm is simple but also inefficient.

a. Time complexity -

- Base Case: O(n)

- **Worst Case:** $O(n^2)$

Input:

```
def bubblesort(arr):
    n = len(arr)
    for i in range(n):
        swapped = False
        for j in range(0,n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True
            if not swapped:
                break
num = [15, 11, 7, 10]
bubblesort(num)
print(num)
```

02.04.25