

ANALYSIS & DESIGN OF ALGORITHM

PRACTICAL-3

190031968

Kinthali praneeth

IN-LAB:

1)

```
def merge(l,r):
    result = []
    leftIndex,rightIndex = 0,0
    i = 0
    while (i<len(l)+len(r)):
        i += 1
        if (rightIndex>=len(r) or (leftIndex<len(l) and
l[leftIndex]<=r[rightIndex])):
            result.append(l[leftIndex])
            leftIndex += 1
        else:
            result.append(r[rightIndex])
            rightIndex += 1
    return result

def merge_sort(arr):
    n = len(arr)
    if n<=1:
        return arr
    l = merge_sort(arr[:n//2])
    r = merge_sort(arr[n//2:])

    return merge(l,r)
n = int(input())
y = [int(i) for i in input().split()]
x = merge_sort(y)
print(x)
```

PYTH 3.6 (Python 3.6)

```
1 def merge(l,r):
2     result = []
3     leftIndex,rightIndex = 0,0
4     i = 0
5     while (i<len(l)+len(r)):
6         i += 1
7         if (rightIndex>=len(r) or (leftIndex<len(l) and l[leftIndex]<=r[rightIndex]]):
8             result.append(l[leftIndex])
9             leftIndex += 1
10        else:
11            result.append(r[rightIndex])
12            rightIndex += 1
13    return result
14
15 def merge_sort(arr):
16     n = len(arr)
17     if n<=1:
18         return arr
19     l = merge_sort(arr[:n//2])
20     r = merge_sort(arr[n//2:])
21
22     return merge(l,r)
23
24 n = int(input())
25 y = [int(i) for i in input().split()]
26 x = merge_sort(y)
27 print(x)
28
```

Custom Input

```
4
1 9 0 3 6 8
```

Status Successfully executed **Date** 2021-08-04 16:26:22 **Time** 0.02 sec **Mem** 17.968 kB

Input

```
4
1 9 0 3 6 8
```

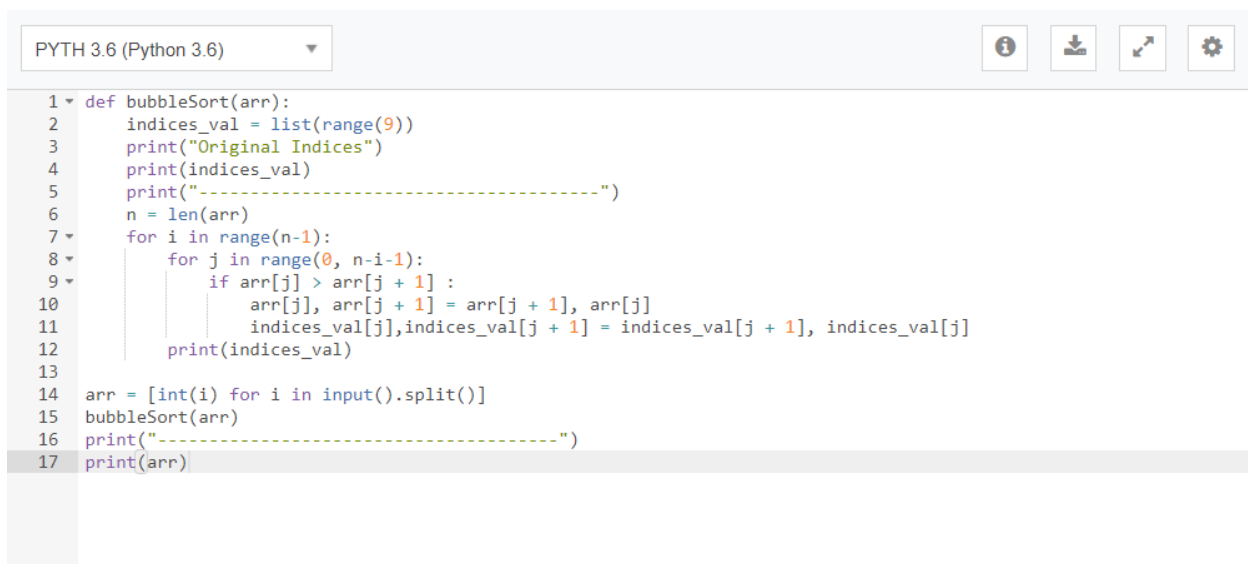
Output

```
[0, 1, 3, 6, 8, 9]
```

2)

```
def bubbleSort(arr):
    indices_val = list(range(9))
    print("Original Indices")
    print(indices_val)
    print("-----")
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j + 1] :
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                indices_val[j], indices_val[j + 1] = indices_val[j + 1], indices_val[j]
        print(indices_val)

arr = [int(i) for i in input().split()]
bubbleSort(arr)
print("-----")
print(arr)
```

A screenshot of a Python 3.6 IDE window. The title bar shows 'PYTH 3.6 (Python 3.6)' and several icons (info, save, share, settings). The code editor contains the same Python code as the previous block, with line numbers 1 through 17 on the left. The code defines a bubbleSort function that sorts an array and tracks indices, then reads input, sorts it, and prints the result. The code is syntax-highlighted with colors like blue for keywords, green for strings, and purple for comments. The last line of the code, 'print(arr)', is highlighted with a light blue background.

```
1 def bubbleSort(arr):
2     indices_val = list(range(9))
3     print("Original Indices")
4     print(indices_val)
5     print("-----")
6     n = len(arr)
7     for i in range(n-1):
8         for j in range(0, n-i-1):
9             if arr[j] > arr[j + 1] :
10                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
11                 indices_val[j], indices_val[j + 1] = indices_val[j + 1], indices_val[j]
12             print(indices_val)
13
14 arr = [int(i) for i in input().split()]
15 bubbleSort(arr)
16 print("-----")
17 print(arr)
```

Status Successfully executed **Date** 2021-08-04 16:21:10 **Time** 0.03 sec **Mem** 17.968 kB



Input

9 5 7 4 3 2 1

Output

Original Indices

[0, 1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 3, 4, 5, 6, 0, 7, 8]

[1, 3, 4, 5, 6, 2, 0, 7, 8]

[3, 4, 5, 6, 1, 2, 0, 7, 8]

[4, 5, 6, 3, 1, 2, 0, 7, 8]

Status Successfully executed **Date** 2021-08-04 16:21:10 **Time** 0.03 sec **Mem** 17.968 kB



Input

9 5 7 4 3 2 1

Output

[3, 4, 5, 6, 1, 2, 0, 7, 8]

[4, 5, 6, 3, 1, 2, 0, 7, 8]

[5, 6, 4, 3, 1, 2, 0, 7, 8]

[6, 5, 4, 3, 1, 2, 0, 7, 8]

[1, 2, 3, 4, 5, 7, 9]