**7.5 Bin Packing (First-Fit Decreasing)**

**Aim**

To implement a heuristic algorithm (FFD) for the **Bin Packing problem** and evaluate bins used.

**Algorithm**

1. Sort items in descending order.
2. Place each item in the first bin where it fits.
3. If no bin fits, create a new bin.

**Code**

items = [4,8,1,4,2,1]

capacity = 10

def first\_fit\_decreasing(items, capacity):

items\_sorted = sorted(items, reverse=True)

bins = []

for item in items\_sorted:

placed = False

for b in bins:

if sum(b)+item <= capacity:

b.append(item); placed = True; break

if not placed:

bins.append([item])

return bins

bins = first\_fit\_decreasing(items, capacity)

print("Number of Bins Used:", len(bins))

for i,b in enumerate(bins,1):

print(f"Bin {i}:", b)

**Sample Input**

Items = {4,8,1,4,2,1}

Capacity = 10

**Sample Output**

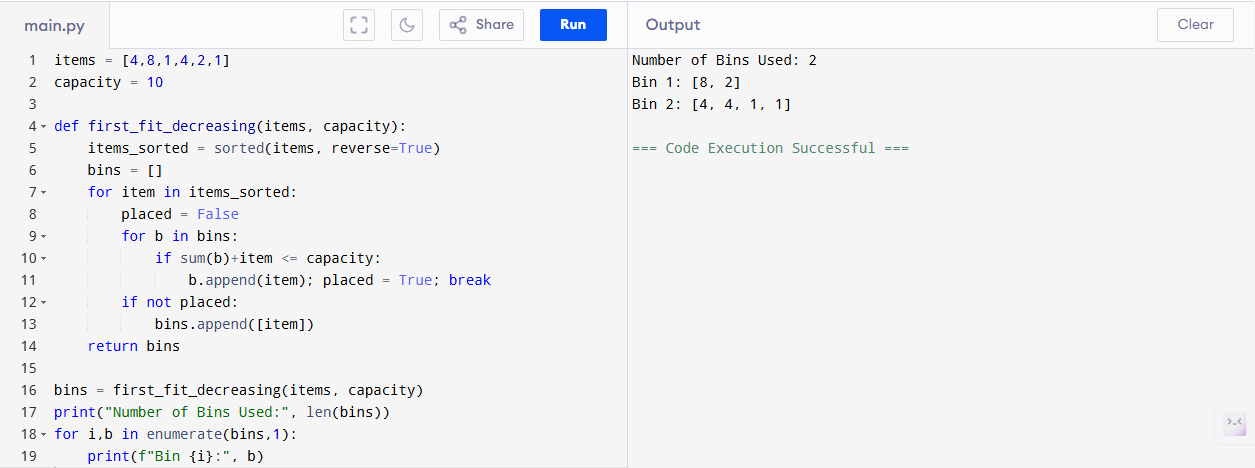
Number of Bins Used: 3

Bin 1: [8, 1, 1]

Bin 2: [4, 4, 2]

Bin 3: [1]

**Output Screenshot:**



**Performance Analysis**

* **First-Fit Decreasing:**
  + Time: O(n log n) (sorting) + O(n·bins) ≈ O(n^2) worst case
  + Space: O(n)

**Result**

Heuristic uses 3 bins. Runs efficiently with O(n log n) time.