Q.2.

(a) Greedy Approach:

(1) Sort the weight array in descending order.

(2) For each item, we start with the first bin and try to put the next weight in the first weight. If not possible then we go to the second bin and if not possible in the second third bin and so on... In each step as we add the item we decrease the capacity of the bin.

(b)

↳ Algorithm — Next page.

We have n weights:

Step — ① : Sort the weight array in decreasing order.

Step — ② : Put the n bins and e capacity Bin[n] and initialize require_bin = 0

Step — ③ : for each item i, do step 4 and 5

Step — ④ : if (weight[i] <= Bin[j]) do

Bin[j] - weight[i]

Step — ⑤. if j == require_bin then that means no bin can hold the current item so we increase require_Bin.

require_Bin++

Bin[require_bin] = Bin[require_bin] - weight[i]

Step — ⑥ require_bin will be the required number of bins.

(c)

$w = \{2, 5, 4, 7, 1, 3, 8\}$

capacity $c = 10$.

Sort items:

$\{8, 7, 5, 4, 3, 2, 1\}$.

Current Bin status = $\{10, 10, 10, 10, 10, 10, 10\}$.

for item at index = (0)

required_bin = 1

current Bin status = $\{2, 10, 10, 10, 10, 10, 10\}$

↑

(10 - 8)

for item at index = (1)

required_bin = 2

current BIN status = $\{2, 3, 10, 10, 10, 10, 10\}$

↑

(10 - 3)

for item at index (2)

required bin = 3

current BIN status = $\{2, 3, 5, 10, 10, 10, 10\}$

↑

(10 - 5) = 5

For item at index = (8)

required_bin = 3

current bin status : { 2, 3, 1, 10, 10, 10, 10 }
↑
(5-4)

For item at index = (4)

required_bin = 3

current bin status : { 2, 0, 1, 10, 10, 10, 10 }
↑
(3-3)

For item at index = (5)

required bin = 3

current bin status : { 0, 0, 1, 10, 10, 10, 10 }
↑
(2-2)

For item at index = (6)

required bin = 3

current bin status : { 0, 0, 0, 10, 10, 10, 10 }
↑
(1-1)

∴ Total Number of bins required for the items are **3**