Answer to #16 in Chapter 2, 8th edition

This question can be answered using the following pseudocode and example which show that all free sections of memory can be merged into one contiguous block without using secondary storage as a temporary holding area. It could generate a discussion on additional hardware components needed to implement this method of compaction.

Example:

**Main Memory**

0 K

|  |
| --- |
| Operating System |
| Job 1 (5 K) |
| Empty (10 K) |
| Job 2 (15 K) |
| Empty (5 K) |
| Job 3 (8 K) |
| Empty (2 K) |

60 K

Job 4 requires 15K and a block that large is not yet available. Therefore, compaction is needed.

**Pseudocode:**

compaction-case = 0

k = 1

DO-WHILE there are free‑partitions

j = 1

DO-WHILE there are active jobs

IF free-partition-size(k) = active-job-size(j)

THEN compaction-case = 1

relocate active job(j) into free-partition(k)

ELSE

IF free-partition-size(k) > active job-size(j)

THEN compaction-case = 2

relocate active-job(j) into free-partition(k)

free-partition-size (k) = free-partition-size (k) - active-job-size(j)

ELSE compaction-case = 3

difference = active job‑size(j) - free-partition‑size(k)

beginning = 0

DO-WHILE beginning not = active job-size(j)

ending = beginning + free-partition-size(k)

relocate active job(j) from beginning to ending

release active job(j) from beginning to ending

beginning = ending

difference = difference - free-partition-size(k)

IF difference < 0

THEN difference = difference - free-partition-size(k)

free-partition-size(k) = difference

END-IF

END-DO-WHILE

END-IF

END-IF

j = j + 1

END-DO-WHILE

k = k + l

END-DO-WHILE

IF compaction‑case = 0

THEN "Compaction is not possible"

ELSE

perform update of free and busy lists

END-IF