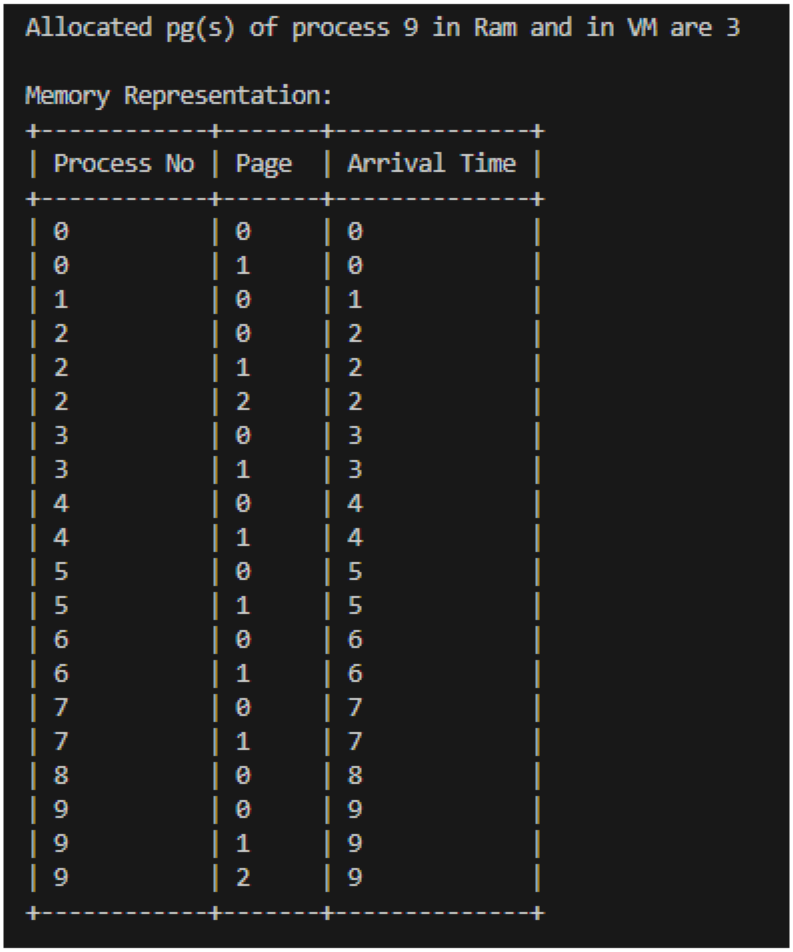
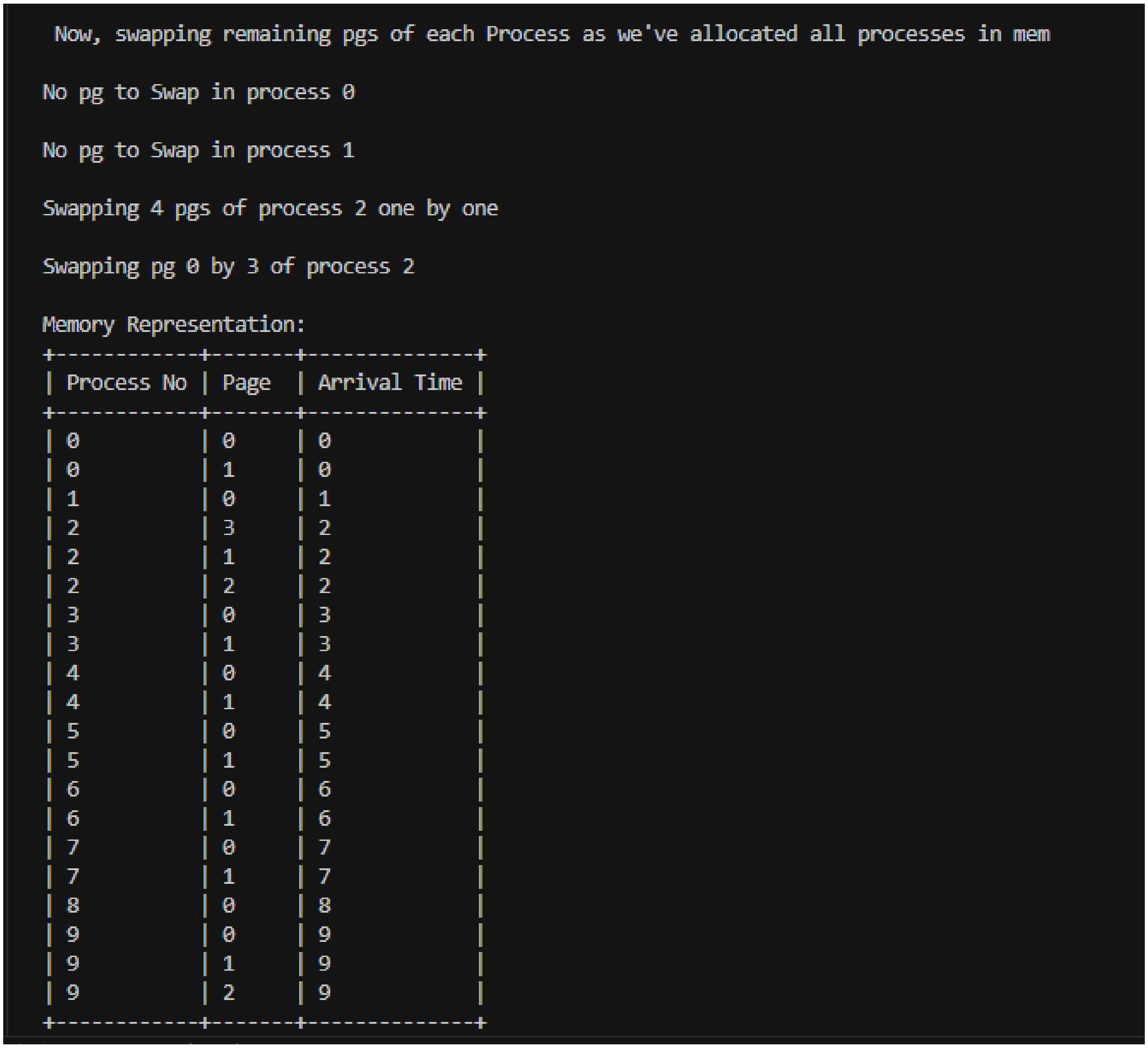


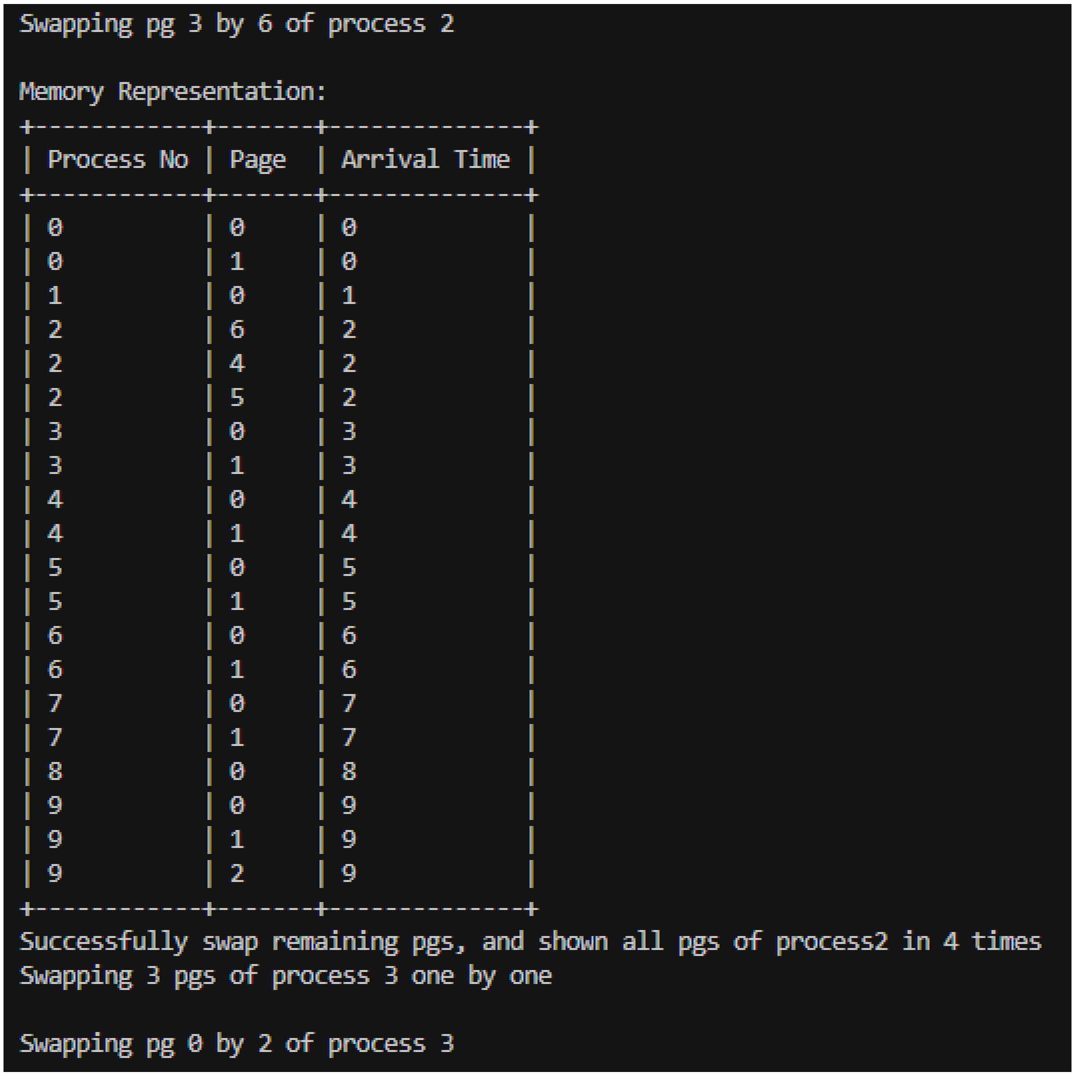
```
Allocated pg(s) of process 9 in Ram and in VM are 3

Memory Representation:
+------------+-------+--------------+
| Process No | Page  | Arrival Time |
+------------+-------+--------------+
| 0          | 0     | 0            |
| 0          | 1     | 0            |
| 1          | 0     | 1            |
| 2          | 0     | 2            |
| 2          | 1     | 2            |
| 2          | 2     | 2            |
| 3          | 0     | 3            |
| 3          | 1     | 3            |
| 4          | 0     | 4            |
| 4          | 1     | 4            |
| 5          | 0     | 5            |
| 5          | 1     | 5            |
| 6          | 0     | 6            |
| 6          | 1     | 6            |
| 7          | 0     | 7            |
| 7          | 1     | 7            |
| 8          | 0     | 8            |
| 9          | 0     | 9            |
| 9          | 1     | 9            |
| 9          | 2     | 9            |
+------------+-------+--------------+
```

Here frames /per process assign is 2 pgs as 20 frames in mem and 10 processes, but note that as process 1 have only 1 pg, and 1pg were unallocated ,so next process which is 2 starts just next to allocated frame, and as process 2 fills unallocated frame for process1, so process 3 gets only 2 pg, as process3 contains total 5 pgs, so 2 are allocated and 3 are remaining. Similarly in case of process 8 and 9



```
 Now, swapping remaining pgs of each Process as we've allocated all processes in mem

No pg to Swap in process 0

No pg to Swap in process 1

Swapping 4 pgs of process 2 one by one

Swapping pg 0 by 3 of process 2

Memory Representation:
+------------+-------+--------------+
| Process No | Page  | Arrival Time |
+------------+-------+--------------+
| 0          | 0     | 0            |
| 0          | 1     | 0            |
| 1          | 0     | 1            |
| 2          | 3     | 2            |
| 2          | 1     | 2            |
| 2          | 2     | 2            |
| 3          | 0     | 3            |
| 3          | 1     | 3            |
| 4          | 0     | 4            |
| 4          | 1     | 4            |
| 5          | 0     | 5            |
| 5          | 1     | 5            |
| 6          | 0     | 6            |
| 6          | 1     | 6            |
| 7          | 0     | 7            |
| 7          | 1     | 7            |
| 8          | 0     | 8            |
| 9          | 0     | 9            |
| 9          | 1     | 9            |
| 9          | 2     | 9            |
+------------+-------+--------------+
```

As all processes pgs are initially allocated in mem, so now we’re swapping remaining pgs, so we go through each process and then swap each pg at a time, until all pgs of each processes’s are shown in memory at least once.



```
Swapping pg 3 by 6 of process 2

Memory Representation:
+------------+-------+--------------+
| Process No | Page  | Arrival Time |
+------------+-------+--------------+
| 0          | 0     | 0            |
| 0          | 1     | 0            |
| 1          | 0     | 1            |
| 2          | 6     | 2            |
| 2          | 4     | 2            |
| 2          | 5     | 2            |
| 3          | 0     | 3            |
| 3          | 1     | 3            |
| 4          | 0     | 4            |
| 4          | 1     | 4            |
| 5          | 0     | 5            |
| 5          | 1     | 5            |
| 6          | 0     | 6            |
| 6          | 1     | 6            |
| 7          | 0     | 7            |
| 7          | 1     | 7            |
| 8          | 0     | 8            |
| 9          | 0     | 9            |
| 9          | 1     | 9            |
| 9          | 2     | 9            |
+------------+-------+--------------+
Successfully swap remaining pgs, and shown all pgs of process2 in 4 times
Swapping 3 pgs of process 3 one by one

Swapping pg 0 by 2 of process 3
```

Here as process2 have 7 pgs, so initially 3 pgs were allocated then in each iteration 1 pg is swap at a time, so 4 iterations are required