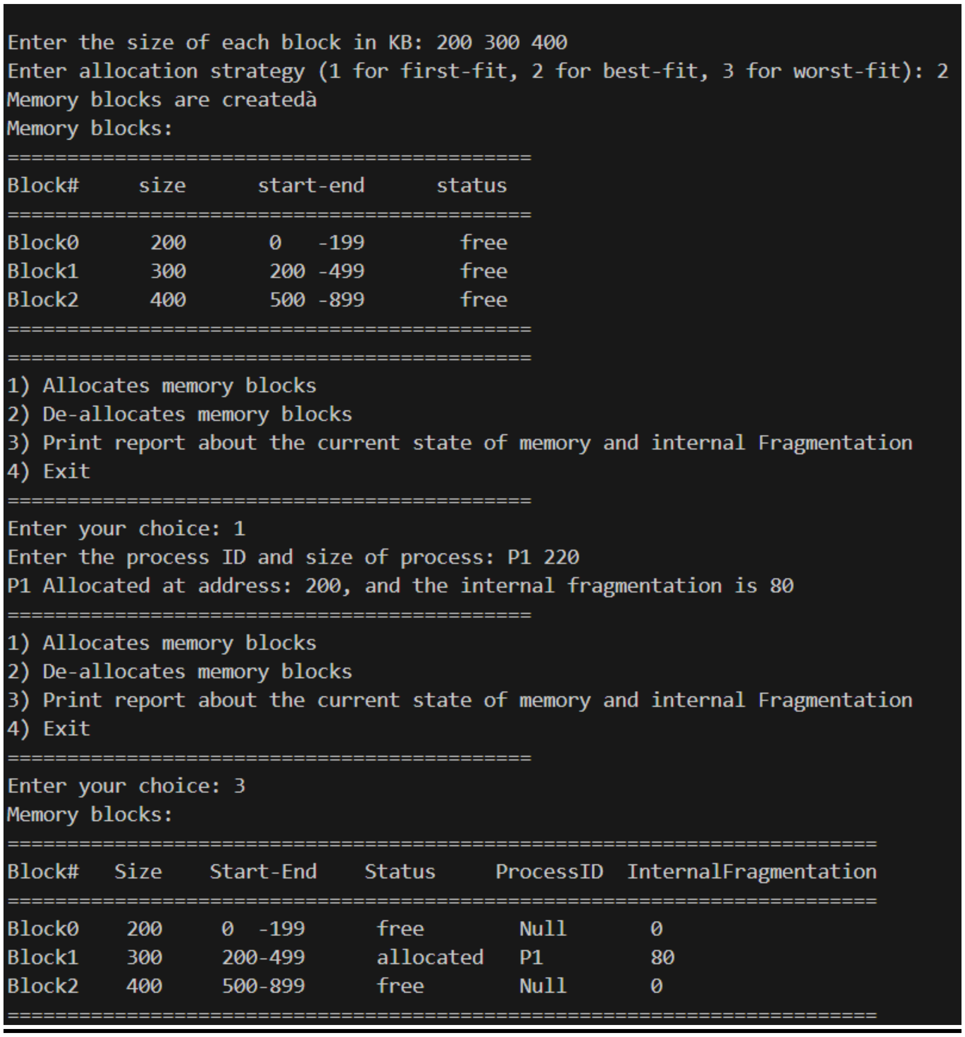
King Saud University College of Computer and Information Sciences Computer Science Department	
CSC 227 *Operating Systems*	*Second semester* *2025-1446*

CSC 227
Programming Assignment 2
Memory Fragmentation Simulation

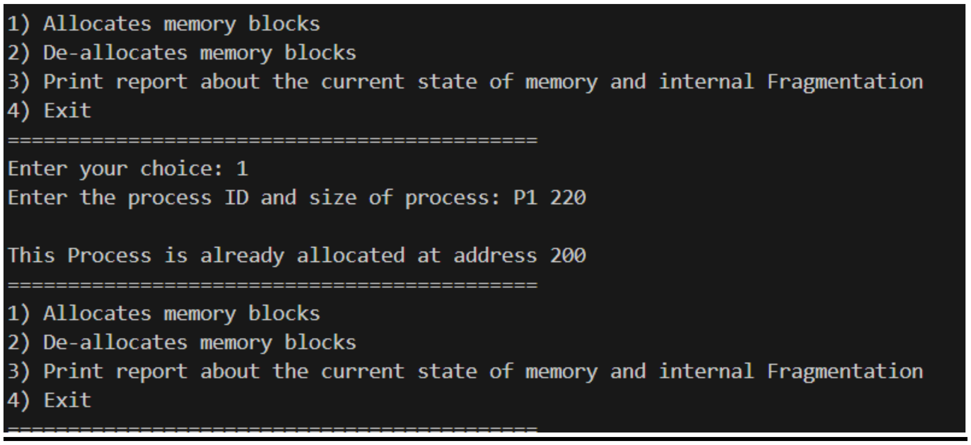
Group #6		Phase #2	
Students			
Name	ID	Section	Tasks
Eman Ameen (Group Leader)	444200073	56304	Implementation of Best-Fit algorithm, method: isAllocated(), isFull()
Lama Abusaada	444200090		Implementation of First-Fit algorithm
Laura Almasoud	444200982		Set up the main, handle user input, class Block, method printInitialMemory()
Tasneem Almusalma	444200111		Implementation of method printReport(), deallocate()
Ghaliyah Alkhaledy	444200534		Implementation of Worst-Fit algorithm

<u>sample inputs/outputs:</u>



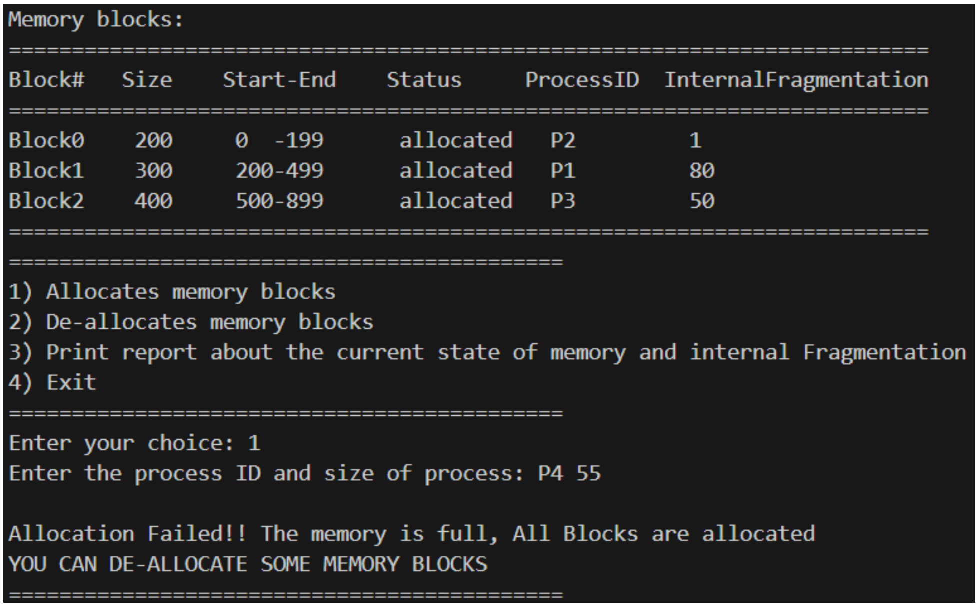
```
Enter the size of each block in KB: 200 300 400
Enter allocation strategy (1 for first-fit, 2 for best-fit, 3 for worst-fit): 2
Memory blocks are createdà
Memory blocks:
============================================
Block#     size      start-end      status
============================================
Block0      200       0   -199        free
Block1      300       200 -499        free
Block2      400       500 -899        free
============================================
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P1 220
P1 Allocated at address: 200, and the internal fragmentation is 80
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 3
Memory blocks:
==========================================================================
Block#   Size    Start-End    Status     ProcessID  InternalFragmentation
==========================================================================
Block0    200     0  -199      free        Null       0
Block1    300     200-499      allocated   P1         80
Block2    400     500-899      free        Null       0
==========================================================================
```

Allocate P1 with size 220 (Best-Fit) and print the report of the memory state after the allocation



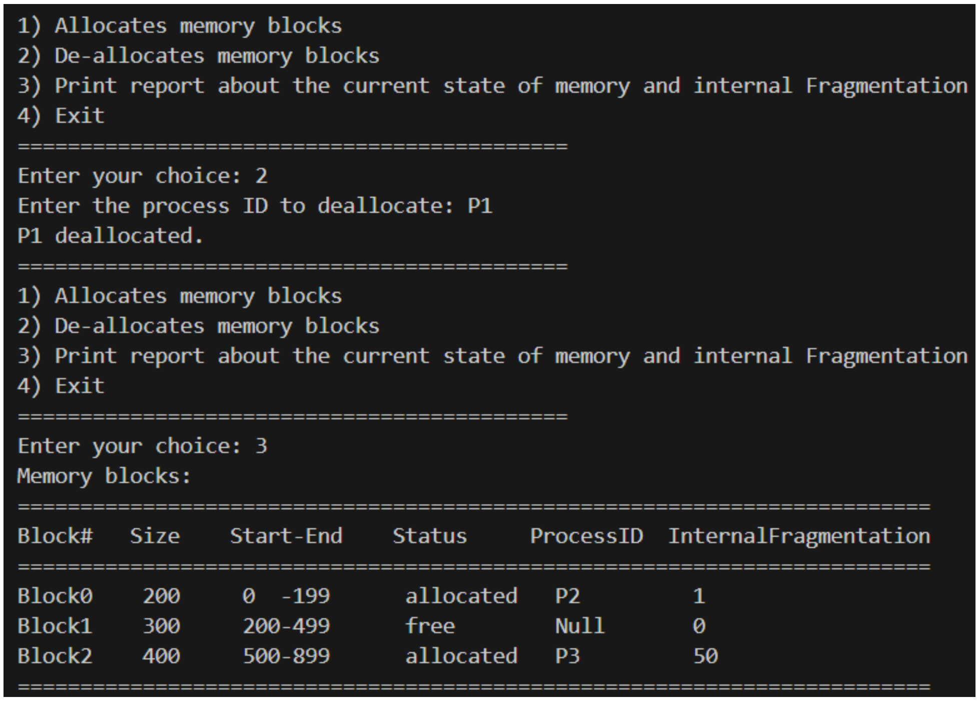
```
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P1 220

This Process is already allocated at address 200
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
```

Allocate P1, which is already allocated in the previous picture



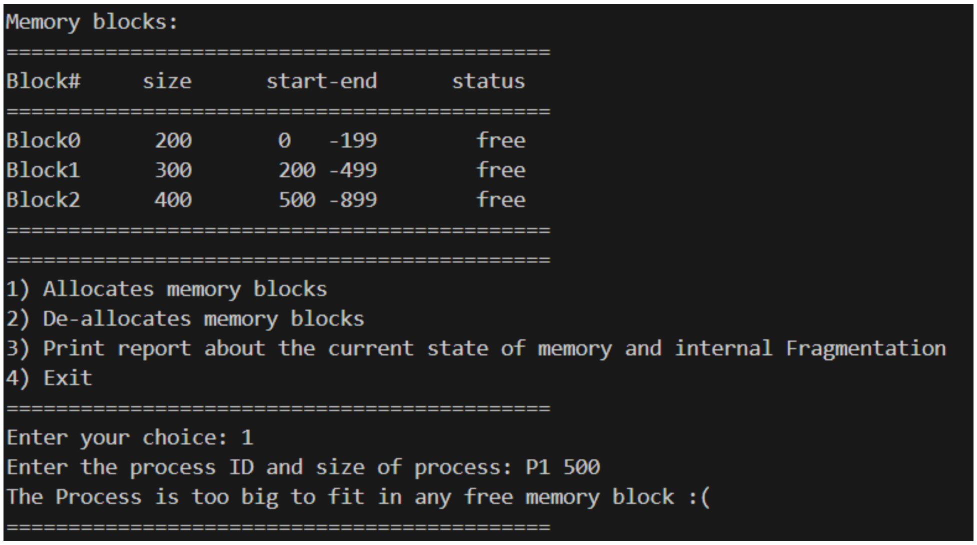
```
Memory blocks:
=========================================================================
Block#   Size    Start-End    Status     ProcessID  InternalFragmentation
=========================================================================
Block0    200     0  -199      allocated    P2         1
Block1    300     200-499      allocated    P1         80
Block2    400     500-899      allocated    P3         50
=========================================================================
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P4 55

Allocation Failed!! The memory is full, All Blocks are allocated
YOU CAN DE-ALLOCATE SOME MEMORY BLOCKS
============================================
```

print the report of the memory state(to show the memory is full), then try to allocate a new process P4



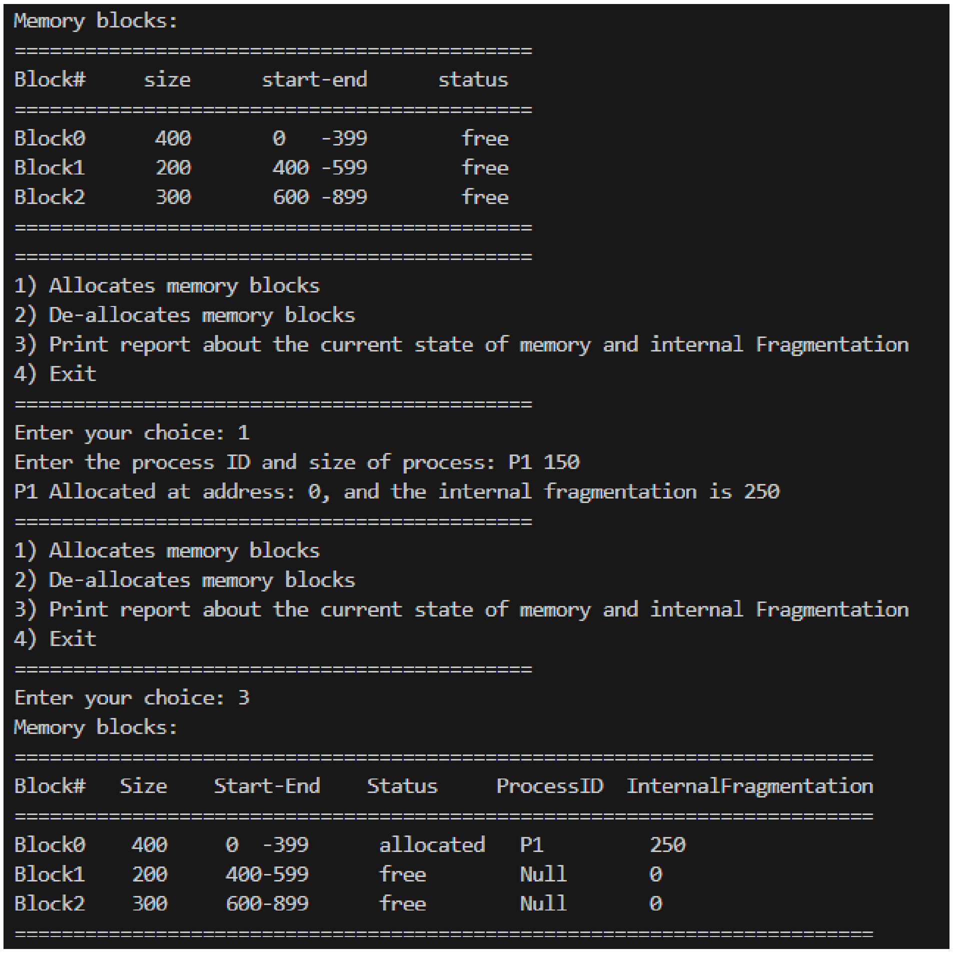
```
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 2
Enter the process ID to deallocate: P1
P1 deallocated.
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 3
Memory blocks:
=========================================================================
Block#   Size    Start-End    Status     ProcessID  InternalFragmentation
=========================================================================
Block0    200     0  -199      allocated    P2         1
Block1    300     200-499      free         Null       0
Block2    400     500-899      allocated    P3         50
=========================================================================
```

Deallocate P1, then print the report of the memory state after Deallocation



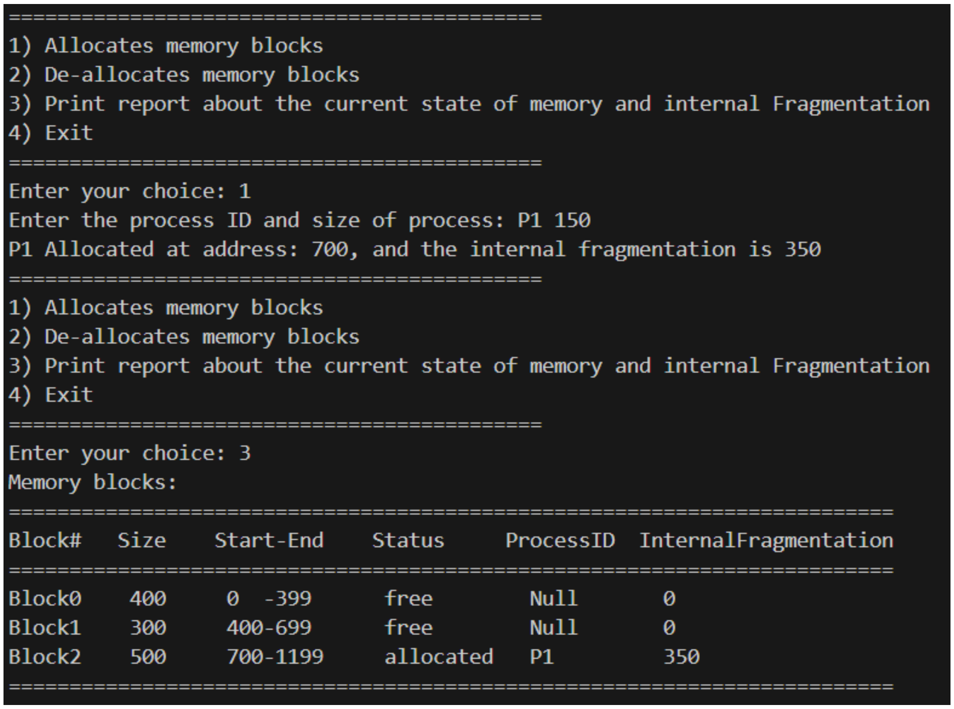
```
Memory blocks:
============================================
Block#     size      start-end      status
============================================
Block0      200       0    -199        free
Block1      300       200 -499        free
Block2      400       500 -899        free
============================================
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P1 500
The Process is too big to fit in any free memory block :(
============================================
```

Trying to allocate a Process with size bigger than all memory blocks size



```
Memory blocks:
============================================
Block#     size      start-end      status
============================================
Block0      400       0    -399        free
Block1      200       400 -599        free
Block2      300       600 -899        free
============================================
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P1 150
P1 Allocated at address: 0, and the internal fragmentation is 250
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 3
Memory blocks:
=========================================================================
Block#   Size    Start-End    Status     ProcessID  InternalFragmentation
=========================================================================
Block0    400     0  -399      allocated   P1         250
Block1    200     400-599      free        Null       0
Block2    300     600-899      free        Null       0
=========================================================================
```

Allocate P1 with size 150 in memory blocks: 400,200,300 (First-Fit), then print memory state report



```
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 1
Enter the process ID and size of process: P1 150
P1 Allocated at address: 700, and the internal fragmentation is 350
============================================
1) Allocates memory blocks
2) De-allocates memory blocks
3) Print report about the current state of memory and internal Fragmentation
4) Exit
============================================
Enter your choice: 3
Memory blocks:
=========================================================================
Block#   Size    Start-End    Status     ProcessID  InternalFragmentation
=========================================================================
Block0    400     0  -399      free       Null       0
Block1    300     400-699      free       Null       0
Block2    500     700-1199     allocated  P1         350
=========================================================================
```

Allocate P1 with size 150 in memory blocks: 400, 300, 500 (Worst-Fit) then print memory state report