King Saud University College of Computer Information Sciences Computer Science Department	
CSC227 **Operating System**	**3rd semester** **2022-2023**
Instructor	**Afnan Algobail**
Section	**69222**

CSC227 Project
Memory Management

Section: 69222 | Group #6

Student Name	Student ID
Aljohara Aljubair	442200138
Waad Alahmed	442200402
Deema Alosaimi	442200426
Lama Alyahya	442200470
Rawan Alotaibi	442201374

Contents

1.Task distribution:

	Name	Task
1	**Aljohara Aljubair**	Coding: Main class and Partition Organize report and README file Share screen during Zoom meetings
2	**Waad Alahmed**	Coding: Main class and Partition Organize report and README file Actively participate in Zoom meetings
3	**Deema Alosaimi**	Coding: Main class and Partition Organize report and README file Actively participate in Zoom meetings
4	**Lama Alyahya**	Coding: Main class and Partition Organize report and README file Actively participate in Zoom meetings
5	**Rawan Alotaibi**	Coding: Main class and Partition Organize report and README file Actively participate in Zoom meetings

2.Partition class:

We define a Partition class that represent a memory partition. Each partition has properties such as status (free or allocated), size, starting and ending addresses, process ID, and internal fragmentation.

3. Main class:

The main method is the entry point of the program. It prompts the user to enter the number of partitions (M), creates an array of Partition objects, and initializes each partition with the provided size.

The user is asked to choose an allocation strategy (First-fit, Best-fit, or Worst-fit) to allocate memory blocks.

The program enters a menu loop where the user can choose different options.

3.1 Option 1(Allocation):

Allocate a block of memory. The user is prompted to enter a process ID and size. Depending on the chosen allocation strategy:

First-Fit :

The First-Fit memory allocation approach is a simple and widely used algorithm. It searches for the first available memory block that is large enough to accommodate the incoming process or data. It starts from the beginning of the memory space and scans through the memory blocks sequentially until it finds a suitable block.

Best-Fit:

The Best-Fit memory allocation approach aims to minimize internal fragmentation. It searches for the smallest available memory block that is still large enough to accommodate the incoming process or data. The algorithm compares the sizes of all available memory blocks and selects the one that best matches the required size.

Worst-Fit:

The Worst-Fit memory allocation approach focuses on maximizing the remaining free space in memory. It searches for the largest available memory block to accommodate the incoming process or data.

3.2 Option 2(Deallocate):

Deallocate a block of memory. The user enters a process ID, and the program searches for the corresponding partition and marks it as free.

3.2 Option 3(Report detailed information)

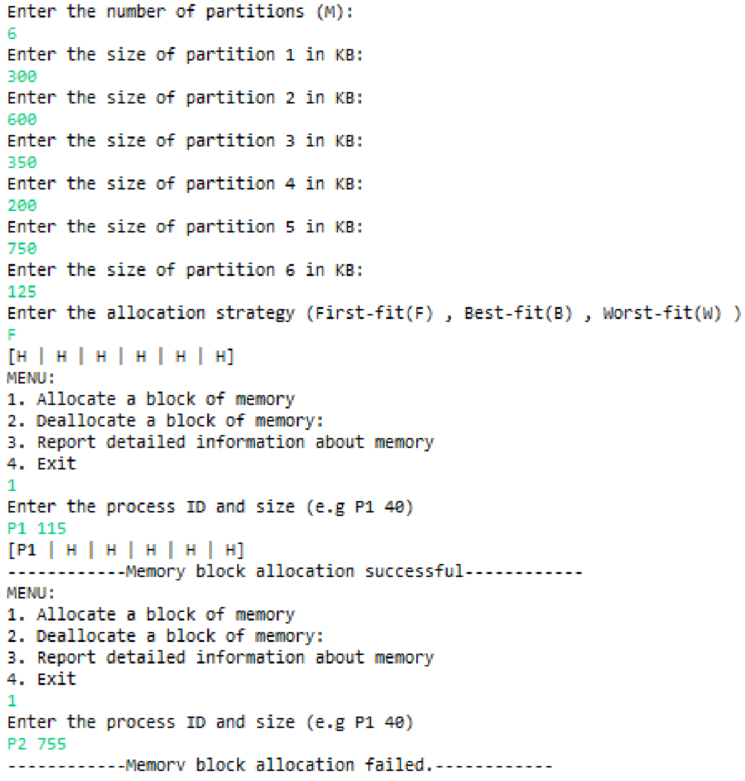
The program will displays the status, size, addresses, process ID, and internal fragmentation of each partition and write it in “Report.txt”.

4. Screenshot of the output:

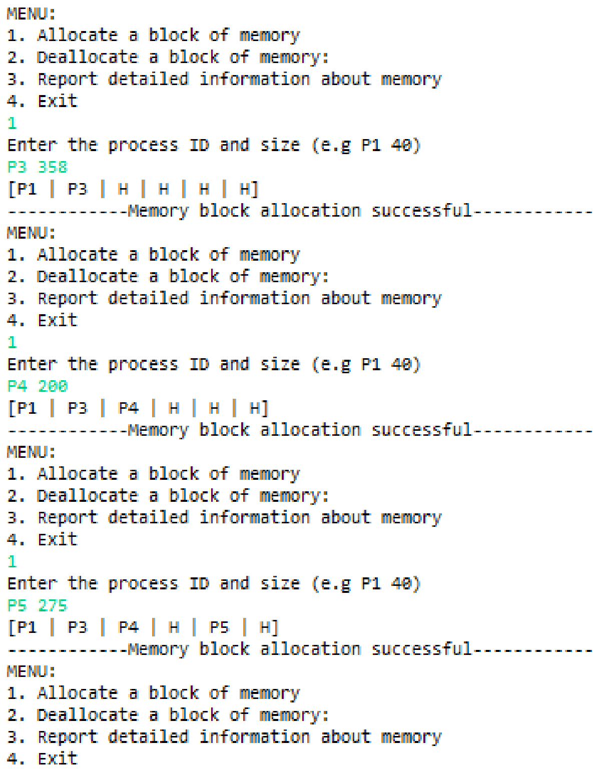
Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order). And five processes of size PI(115 KB), P2(755 KB), P3(358 KB), P4(200 KB), and P5(275 KB) (in order) .

Note: Assume the search always start at the beginning of the memory list.

4.1 First-Fit:

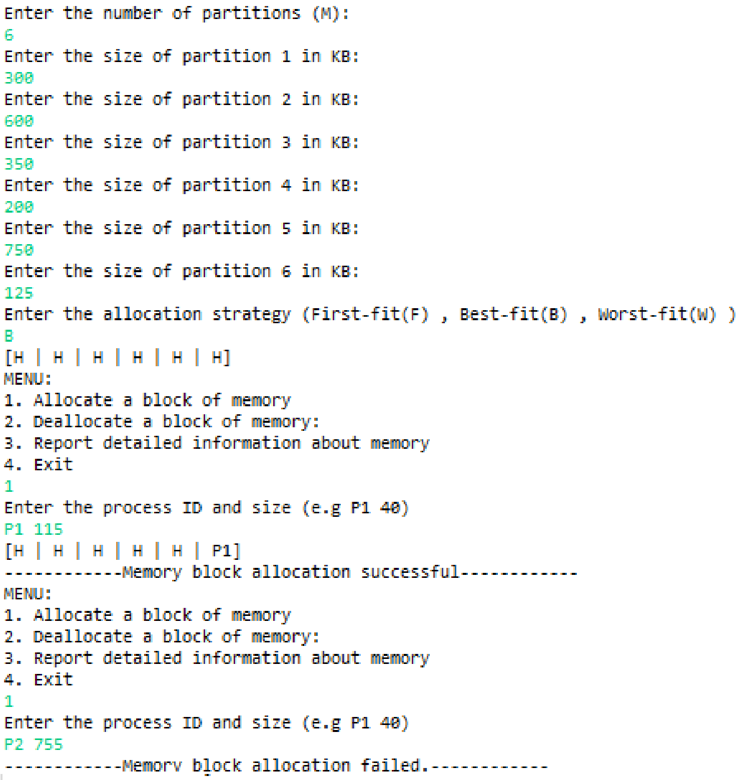


```
Enter the number of partitions (M):
6
Enter the size of partition 1 in KB:
300
Enter the size of partition 2 in KB:
600
Enter the size of partition 3 in KB:
350
Enter the size of partition 4 in KB:
200
Enter the size of partition 5 in KB:
750
Enter the size of partition 6 in KB:
125
Enter the allocation strategy (First-fit(F) , Best-fit(B) , Worst-fit(W) )
F
[H | H | H | H | H | H]
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P1 115
[P1 | H | H | H | H | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P2 755
------------Memory block allocation failed.------------
```

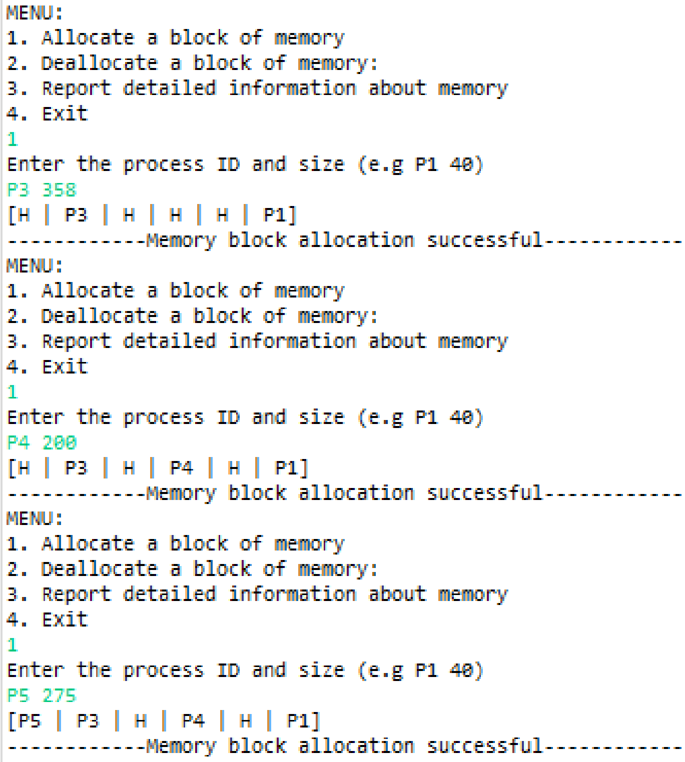


```
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P3 358
[P1 | P3 | H | H | H | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P4 200
[P1 | P3 | P4 | H | H | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P5 275
[P1 | P3 | P4 | H | P5 | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
```

4.2 Best-Fit:

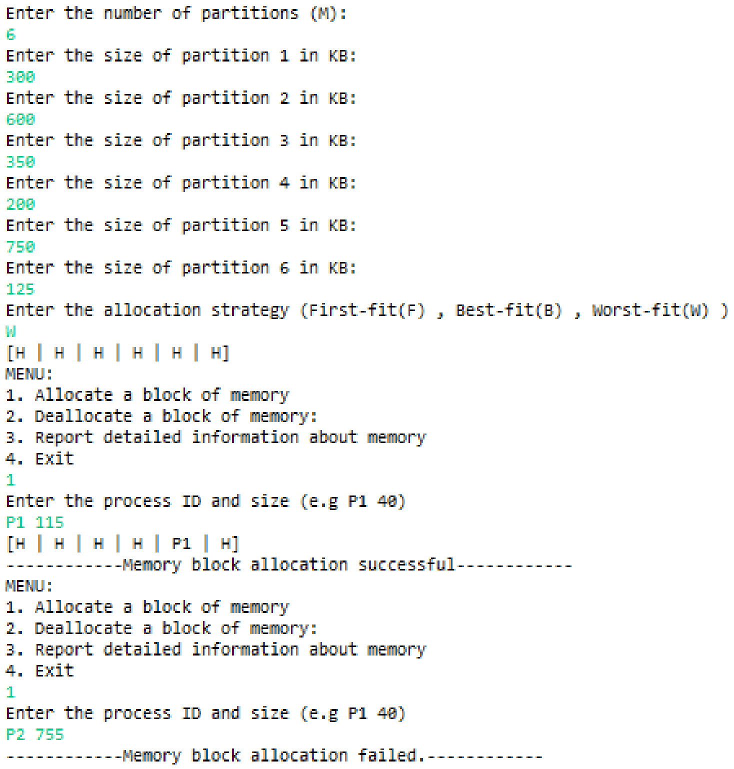


```
Enter the number of partitions (M):
6
Enter the size of partition 1 in KB:
300
Enter the size of partition 2 in KB:
600
Enter the size of partition 3 in KB:
350
Enter the size of partition 4 in KB:
200
Enter the size of partition 5 in KB:
750
Enter the size of partition 6 in KB:
125
Enter the allocation strategy (First-fit(F) , Best-fit(B) , Worst-fit(W) )
B
[H | H | H | H | H | H]
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P1 115
[H | H | H | H | H | P1]
-----------Memory block allocation successful-----------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P2 755
-----------Memory block allocation failed.------------
```

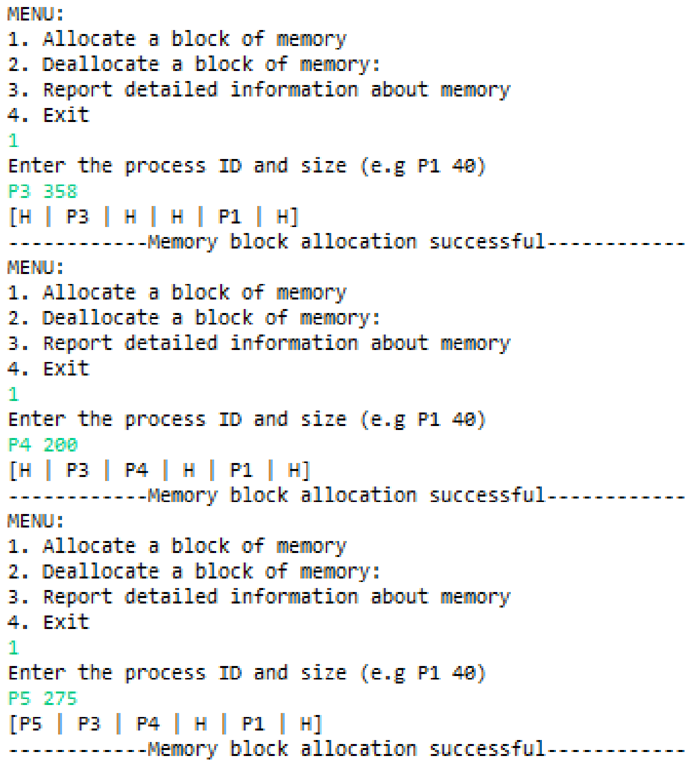


```
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P3 358
[H | P3 | H | H | H | P1]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P4 200
[H | P3 | H | P4 | H | P1]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P5 275
[P5 | P3 | H | P4 | H | P1]
------------Memory block allocation successful------------
```

4.3 Worst-Fit:

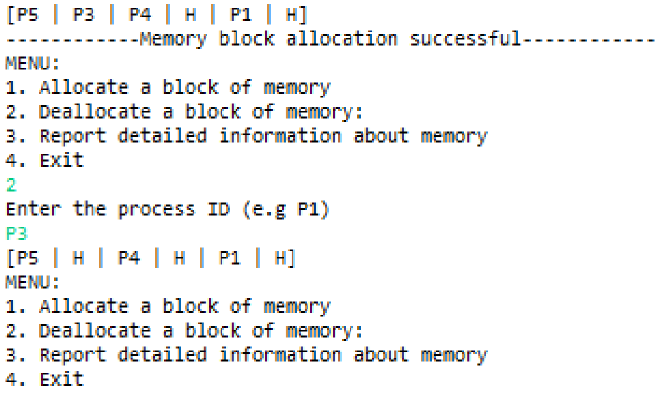


```
Enter the number of partitions (M):
6
Enter the size of partition 1 in KB:
300
Enter the size of partition 2 in KB:
600
Enter the size of partition 3 in KB:
350
Enter the size of partition 4 in KB:
200
Enter the size of partition 5 in KB:
750
Enter the size of partition 6 in KB:
125
Enter the allocation strategy (First-fit(F) , Best-fit(B) , Worst-fit(W) )
W
[H | H | H | H | H | H]
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P1 115
[H | H | H | H | P1 | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P2 755
------------Memory block allocation failed.------------
```



```
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P3 358
[H | P3 | H | H | P1 | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P4 200
[H | P3 | P4 | H | P1 | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
1
Enter the process ID and size (e.g P1 40)
P5 275
[P5 | P3 | P4 | H | P1 | H]
------------Memory block allocation successful------------
```

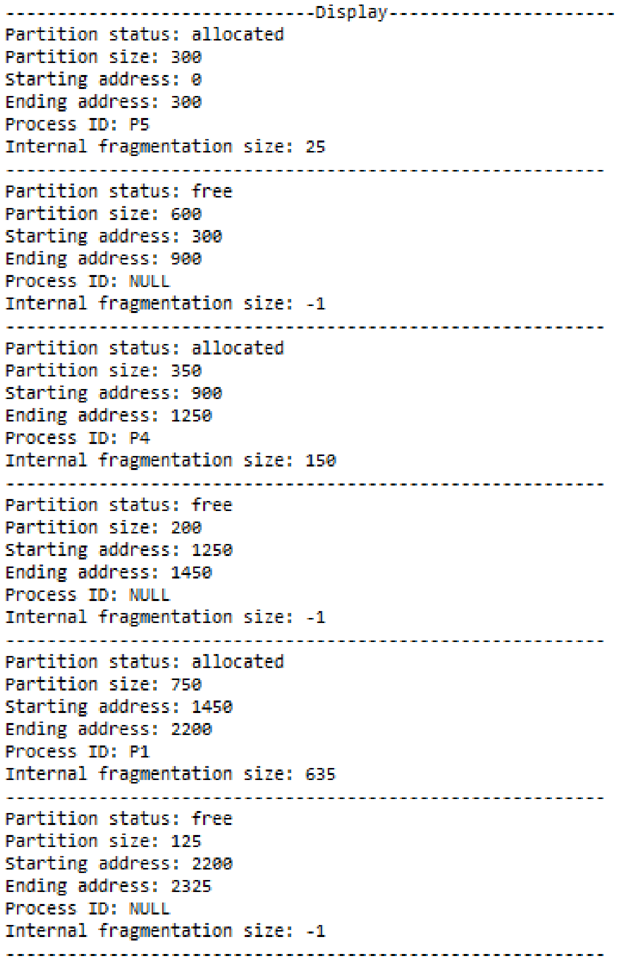
4.4 Deallocation:



```
[P5 | P3 | P4 | H | P1 | H]
------------Memory block allocation successful------------
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
2
Enter the process ID (e.g P1)
P3
[P5 | H | P4 | H | P1 | H]
MENU:
1. Allocate a block of memory
2. Deallocate a block of memory:
3. Report detailed information about memory
4. Exit
```

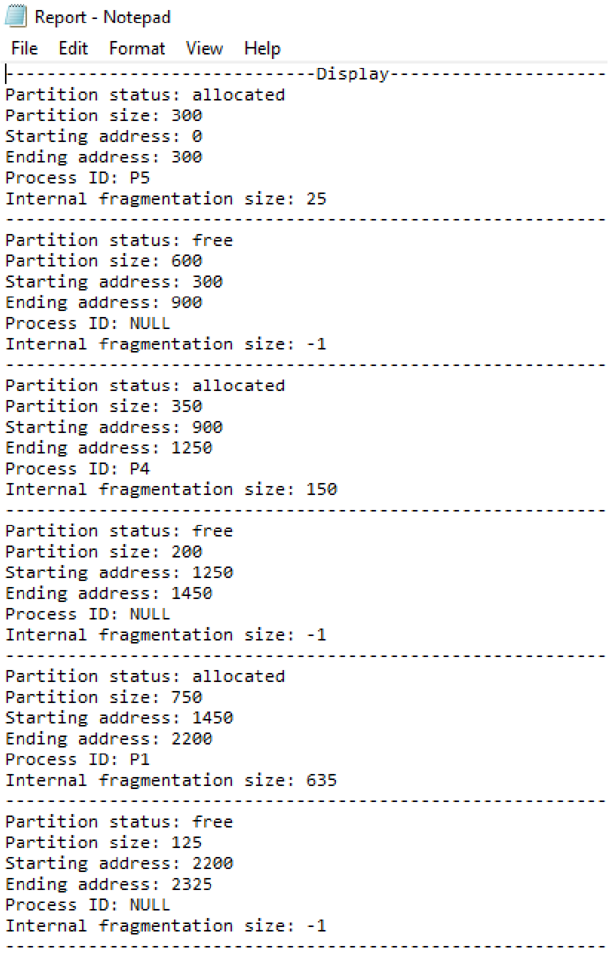
4.5 Report detailed information:

- **On the console:**



```
-----------------------------Display---------------------
Partition status: allocated
Partition size: 300
Starting address: 0
Ending address: 300
Process ID: P5
Internal fragmentation size: 25
----------------------------------------------------------
Partition status: free
Partition size: 600
Starting address: 300
Ending address: 900
Process ID: NULL
Internal fragmentation size: -1
----------------------------------------------------------
Partition status: allocated
Partition size: 350
Starting address: 900
Ending address: 1250
Process ID: P4
Internal fragmentation size: 150
----------------------------------------------------------
Partition status: free
Partition size: 200
Starting address: 1250
Ending address: 1450
Process ID: NULL
Internal fragmentation size: -1
----------------------------------------------------------
Partition status: allocated
Partition size: 750
Starting address: 1450
Ending address: 2200
Process ID: P1
Internal fragmentation size: 635
----------------------------------------------------------
Partition status: free
Partition size: 125
Starting address: 2200
Ending address: 2325
Process ID: NULL
Internal fragmentation size: -1
----------------------------------------------------------
```

- **On the file”Report.txt”:**



```
-----------------------------Display--------------------
Partition status: allocated
Partition size: 300
Starting address: 0
Ending address: 300
Process ID: P5
Internal fragmentation size: 25
---------------------------------------------------------
Partition status: free
Partition size: 600
Starting address: 300
Ending address: 900
Process ID: NULL
Internal fragmentation size: -1
---------------------------------------------------------
Partition status: allocated
Partition size: 350
Starting address: 900
Ending address: 1250
Process ID: P4
Internal fragmentation size: 150
---------------------------------------------------------
Partition status: free
Partition size: 200
Starting address: 1250
Ending address: 1450
Process ID: NULL
Internal fragmentation size: -1
---------------------------------------------------------
Partition status: allocated
Partition size: 750
Starting address: 1450
Ending address: 2200
Process ID: P1
Internal fragmentation size: 635
---------------------------------------------------------
Partition status: free
Partition size: 125
Starting address: 2200
Ending address: 2325
Process ID: NULL
Internal fragmentation size: -1
---------------------------------------------------------
```

5.Evaluating the performance:

the choice of memory allocation strategy in a contiguous memory management system plays a significant role in determining the overall efficiency and utilization of memory. Each strategy, such as First-Fit, Best-Fit, and Worst-Fit, has its own advantages and trade-offs.

First-Fit, being simple and efficient in terms of time complexity, may suffer from internal fragmentation. This occurs when allocated blocks are larger than necessary, resulting in unused space within the blocks. Best-Fit, on the other hand, aims to reduce wasted space and improve memory utilization by finding the most suitable block. However, this strategy requires additional search time to locate the optimal block, which can impact performance.
Worst-Fit, suitable for scenarios where larger partitions are preferred for future allocations, maximizes the remaining free space. However, it may result in increased fragmentation and slightly lower efficiency compared to other strategies.

When selecting an allocation strategy, it is essential to consider the specific requirements and constraints of the system. Factors such as the expected size of allocations, frequency of allocation and deallocation operations, and the importance of minimizing wasted space and fragmentation should be taken into account. By carefully evaluating these factors, the most appropriate strategy can be chosen to achieve optimal memory management and utilization in the system.

6. Team work Evaluation:

Criteria	Student1	Student2	Student3	Student4	Student5
Work division: Contributed equally to the work	1	1	1	1	1
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA	1	1	1	1	1
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions	1	1	1	1	1
T ime management: Attending on time, being ready to start the demo, good time management in discussion and demo	1	1	1	1	1
Total/4	**4**	**4**	**4**	**4**	**4**