

## *Project Report*

CSC227| 3rd Trimester 2022/2023

Name	ID	Section
<b><i>Dona Aloraini (leader)</i></b>	<b><i>442201363</i></b>	<b><i>75860</i></b>
<b><i>Deema Albarrak</i></b>	<b><i>442201855</i></b>	
<b><i>Jude Alsubaie</i></b>	<b><i>441201049</i></b>	
<b><i>Sarah Alajlan</i></b>	<b><i>442202314</i></b>	
<b><i>Norah Alguraishi</i></b>	<b><i>442201375</i></b>	

## Table of contents

Table of contents	2
Task distribution	2
Implementation	3
1. First-fit	3
2. Best-fit	4
3. Worst-fit	7
4. Insufficient Allocation	9
5. De-allocation	10
6. Reflection	11
7. Peer evaluation	12

## Task distribution

Name	Task
<i>Deema Albarrak</i>	initializeMemory(), displayMenu(), Insufficient allocation summary.
<i>Dona Aloraini</i>	allocateMemory()(Best Fit Code and Summary), Partition Class.
<i>Jude Alsubaie</i>	WriteOutputToFile(), Deallocation summary, Reflection.
<i>Sarah Alajlan</i>	allocateMemory()(Worst Fit Code and Summary), deallocateMemory().
<i>Norah Alguraishi</i>	allocateMemory()(First Fit Code and Summary), displayMemoryStatus().

# Implementation

## 1. First-fit

```
Enter the number of partitions: 6
Enter the size of each partition in KB:
Partition 1: 300
Partition 2: 600
Partition 3: 350
Partition 4: 200
Partition 5: 750
Partition 6: 125
Enter the allocation strategy (F for First-fit, B for Best-fit, W for Worst-fit): F
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: P1
Enter process size (KB): 115
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: P2
Enter process size (KB): 755
Insufficient memory to allocate the process.
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: P3
Enter process size (KB): 358
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: P4
Enter process size (KB): 200
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: P5
Enter process size (KB): 275
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3
```

Memory Partition Information:

Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address
300	allocated	P1	185	0	299
600	allocated	P3	242	300	899
350	allocated	P4	150	900	1249
200	free	Null	-1	1250	1449
750	allocated	P5	475	1450	2199
125	free	Null	-1	2200	2324

Memory State:  
[P1 | P3 | P4 | H | P5 | H]

## report.txt :

Memory Partition Information:						
Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address	
300	allocated	p1	185	0	299	
600	allocated	p3	242	300	899	
350	allocated	p4	150	900	1249	
200	free	Null	-1	1250	1449	
750	allocated	p5	475	1450	2199	
125	free	Null	-1	2200	2324	
Memory State: [p1   p3   p4   H   p5   H]						

## Summary:

When allocationStrategy is set to "F" or "f" (meaning First Fit), the code finds the first available partition that is large enough to support the process, marks it as "allocated," and assigns the process to it. Internal fragmentation is calculated and recorded in the Partition object as well.

## 2. Best-fit

```
Enter the number of partitions: 6
Enter the size of each partition in KB:
Partition 1: 300
Partition 2: 600
Partition 3: 350
Partition 4: 200
Partition 5: 750
Partition 6: 125
Enter the allocation strategy (F for First-fit, B for Best-fit, W for Worst-fit): b

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p1
Enter process size (KB): 115
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p2
Enter process size (KB): 755
Insufficient memory to allocate the process.
```

```

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p3
Enter process size (KB): 358
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p4
Enter process size (KB): 200
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p5
Enter process size (KB): 275
Memory allocated successfully!

```

```

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3

Memory Partition Information:
+-----+-----+-----+-----+-----+-----+
| Partition Size | Partition Status | Process Number | Fragmentation | Starting | Ending |
|               |                 |                | Size          | Address  | Address |
+-----+-----+-----+-----+-----+-----+
| 300           | allocated       | p5             | 25            | 0        | 299     |
+-----+-----+-----+-----+-----+-----+
| 600           | allocated       | p3             | 242           | 300      | 899     |
+-----+-----+-----+-----+-----+-----+
| 350           | free           | Null           | -1            | 900      | 1249    |
+-----+-----+-----+-----+-----+-----+
| 200           | allocated       | p4             | 0             | 1250     | 1449    |
+-----+-----+-----+-----+-----+-----+
| 750           | free           | Null           | -1            | 1450     | 2199    |
+-----+-----+-----+-----+-----+-----+
| 125           | allocated       | p1             | 10            | 2200     | 2324    |
+-----+-----+-----+-----+-----+-----+

Memory State:
[p5 | p3 | H | p4 | H | p1]

```

## report.txt :

```

Memory Partition Information:
+-----+-----+-----+-----+-----+-----+
| Partition Size | Partition Status | Process Number | Fragmentation | Starting | Ending |
|               |                 |                | Size          | Address  | Address |
+-----+-----+-----+-----+-----+-----+
| 300           | allocated       | p5             | 25            | 0        | 299     |
+-----+-----+-----+-----+-----+-----+
| 600           | allocated       | p3             | 242           | 300      | 899     |
+-----+-----+-----+-----+-----+-----+
| 350           | free           | Null           | -1            | 900      | 1249    |
+-----+-----+-----+-----+-----+-----+
| 200           | allocated       | p4             | 0             | 1250     | 1449    |
+-----+-----+-----+-----+-----+-----+
| 750           | free           | Null           | -1            | 1450     | 2199    |
+-----+-----+-----+-----+-----+-----+
| 125           | allocated       | p1             | 10            | 2200     | 2324    |
+-----+-----+-----+-----+-----+-----+

Memory State: [p5 | p3 | H | p4 | H | p1]

```

## Summary:

When the allocation strategy is set to "Best-fit," the program iterates through the list of memory partitions to find the one that is best suited to the process being allocated. It looks for a free partition that is larger than or equal to the process's size. The "Best-fit" strategy seeks to reduce internal fragmentation by selecting the partition with the least amount of unused space (fragmentation) that can support the process.

### 3. Worst-fit

```
Enter the number of partitions: 6
Enter the size of each partition in KB:
Partition 1: 300
Partition 2: 600
Partition 3: 350
Partition 4: 200
Partition 5: 750
Partition 6: 125
Enter the allocation strategy (F for First-fit, B for Best-fit, W for Worst-fit): w

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p1
Enter process size (KB): 115
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p2
Enter process size (KB): 755
Insufficient memory to allocate the process.
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p3
Enter process size (KB): 358
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p4
Enter process size (KB): 200
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p5
Enter process size (KB): 275
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3
```

Memory Partition Information:

Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address
300	allocated	p5	25	0	299
600	allocated	p3	242	300	899
350	allocated	p4	150	900	1249
200	free	Null	-1	1250	1449
750	allocated	p1	635	1450	2199
125	free	Null	-1	2200	2324

Memory State:

[p5 | p3 | p4 | H | p1 | H]

## report.txt :

Memory Partition Information:						
Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address	
300	allocated	p5	25	0	299	
600	allocated	p3	242	300	899	
350	allocated	p4	150	900	1249	
200	free	Null	-1	1250	1449	
750	allocated	p1	635	1450	2199	
125	free	Null	-1	2200	2324	
Memory State: [p5   p3   p4   H   p1   H]						

## Summary:

The Worst Fit allocation strategy iterates through the list of memory partitions, looking for the largest free partition that can accommodate the process being allocated. It looks for a free partition that is larger than or equal to the process's size. The Worst Fit strategy seeks to maximize external fragmentation by choosing the largest available partition, which may result in small gaps of unused memory between allocated partitions.

## 4. Insufficient Allocation

```
Enter the number of partitions: 2
Enter the size of each partition in KB:
Partition 1: 300
Partition 2: 600
Enter the allocation strategy (F for First-fit, B for Best-fit, W for Worst-fit): f

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p1
Enter process size (KB): 115
Memory allocated successfully!

Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p2
Enter process size (KB): 755
Insufficient memory to allocate the process.
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3

Memory Partition Information:
+-----+-----+-----+-----+-----+-----+
| Partition Size | Partition Status | Process Number | Fragmentation | Starting | Ending |
|               |                 |               | Size          | Address  | Address |
+-----+-----+-----+-----+-----+-----+
| 300           | allocated       | p1            | 185           | 0        | 299     |
+-----+-----+-----+-----+-----+-----+
| 600           | free            | Null          | -1            | 300      | 899     |
+-----+-----+-----+-----+-----+-----+

Memory State:
[p1 | H]
```

report.txt :

```
Memory Partition Information:
+-----+-----+-----+-----+-----+-----+
| Partition Size | Partition Status | Process Number | Fragmentation | Starting | Ending |
|               |                 |               | Size          | Address  | Address |
+-----+-----+-----+-----+-----+-----+
| 300           | allocated       | p1            | 185           | 0        | 299     |
+-----+-----+-----+-----+-----+-----+
| 600           | free            | Null          | -1            | 300      | 899     |
+-----+-----+-----+-----+-----+-----+

Memory State: [p1 | H]
```

### Summary:

This starts with a loop for the partition sizes, for every partition, it checks whether it can accommodate the requested process size by an if statement that only allocates the process if it's not bigger than the partition size. Lastly it prints a message letting the user know that there's not enough space for this process.



## 5. De-allocation

```
Enter the number of partitions: 2
Enter the size of each partition in KB:
Partition 1: 300
Partition 2: 600
Enter the allocation strategy (F for First-fit, B for Best-fit, W for Worst-fit): F
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p1
Enter process size (KB): 115
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 1
Enter process ID: p2
Enter process size (KB): 200
Memory allocated successfully!
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3
```

Memory Partition Information:

Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address
300	allocated	p1	185	0	299
600	allocated	p2	400	300	899

Memory State:

[p1 | p2]

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 2
Enter the process' ID
p1
```

```
Menu:
1. Allocate a block of memory
2. De-allocate a block of memory
3. Report detailed information about memory partitions
4. Exit
Enter your choice: 3
```

Memory Partition Information:

Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address
300	free	Null	-1	0	299
600	allocated	p2	400	300	899

Memory State:

[H | p2]

### report.txt :

Memory Partition Information:

Partition Size	Partition Status	Process Number	Fragmentation Size	Starting Address	Ending Address
300	free	Null	-1	0	299
600	allocated	p2	400	300	899

Memory State: [H | p2]

### Summary:

In the `deallocateMemory()` method, the code iterates through the memory partitions and searches for a partition associated with the specified process name and updates its status to “free”, process ID to null, and internal fragmentation to -1. If the process is not found, it prints the process is not found.

## 6. Reflection

We thoroughly enjoyed the project as we investigated various memory allocation techniques. First-fit was a strategy that stood out for being fast. When compared to the Best-fit and Worst-fit techniques, this method quickly allocates the first available partition that can handle a process. First-fit may, however, increase internal fragmentation since smaller processes may take up more partition space. The Best-fit technique, on the other hand, chose the partition that is the minimum size greater than the process in order to reduce internal fragmentation. Memory usage increases as the amount of unused space is decreased. This method can take a while because it involves looking through all of the possible partitions to find the one that fits the situation the best. The Worst-fit technique, in contrast, places more emphasis on choosing the largest available partition for a process. However, it often results in inefficient memory utilization, also it's time-consuming as it requires searching for the worst-fit partition among all available partitions.

## 7. Peer evaluation

Criteria	Dona Aloraini	Deema Albarrak	Jude Alsubaie	Sarah Alajlan	Norah Alguraishi
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
Time 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