Military Institute of Science & Technology (MIST)
Department of Computer Science and Engineering
Evaluation - 2, Course: CSE 204: Computer Data Structure and Algorithms - I Sessional (Sec-A)

Time: 40 minutes **Full marks: 20**

Name:	**Roll:**

Question

Have you seen the series *Game of Thrones*? It is widely regarded as one of the best television series ever made. In *Game of Thrones*, the Seven Kingdoms of Westeros are ruled from the Iron Throne in King's Landing. Among all the noble houses, the Lannisters have one of the most skilled and disciplined armies.

The Lannisters always organize their army in ascending order of height. They employ a more efficient approach to avoid the complexity of sorting the whole lineup whenever a new soldier joins the line. Upon a new soldier's arrival, they insert them directly into their appropriate position in the lineup. For instance, if the current lineup is [1, 2, 2, 3, 4] and a soldier of height 3 joins, they insert them at the third index after the last 2. The updated lineup becomes [1, 2, 2, **3**, 3, 4]. (Implemented in the template)

Discipline is strictly enforced within the Lannister ranks. If a soldier disobeys the Lord of Casterly Rock, the Lord will identify the soldier by their index (**the Lord prefers zero indexing**). That soldier will be sentenced to death and removed from the lineup. For example, if the lineup is [1, 2, 2, 3, 3, 4] and the soldier at index 3 disobeys the Lord's command, the Lord will point out that soldier. Consequently, the disobedient soldier will be executed, resulting in a new lineup [1, 2, 2, 3, 4].

Imagine a scenario where the Lannisters need to fight in two battles simultaneously. In such cases, they divide their army into two groups: army1 and army2. Army1 consists of soldiers at even indices (0, 2, 4, ...), while army2 comprises odd indices (1, 3, 5, ...). For instance, if the original army lineup was [1, 1, 2, 2, 3, 4, 4], after splitting, army1 would be [1, 2, 3, 4] and army2 would be [1, 2, 4].

A strange and brutal event occurs whenever Cersei Lannister visits Casterly Rock. Upon her arrival, she meticulously inspects the lineup. If she notices any duplicate heights, she orders the execution of all soldiers with those duplicate heights, removing them from the lineup. For instance, if the lineup before Cersei's arrival is [1, 2, 2, 3, 4, 4], the new lineup becomes [1, 2, 3, 4] after her ruthless purge.

The Lord of Casterly Rock has discovered that you are a skilled programmer. In Westeros, a new data structure, the **Linked List**, is becoming increasingly popular. The Lord of Casterly Rock has approached you to create an efficient system to manage their armies using this new technology. Your task is to develop three functions:

`ExecuteTheTraitor()`, `SplitArmy()`, and `PurgeDuplicates()`.

At first, the Lord of Casterly Rock approached a programmer to develop this system. However, he was executed when it was discovered that he was incompetent and failed to properly implement any of the three functions. The Lord has now turned to you, **offering some reward for each completed function**. He has provided you with the unfinished code from the previous programmer and tasked you with properly developing the system within 40 minutes to avoid execution and earn your rewards.

Military Institute of Science & Technology (MIST)
Department of Computer Science and Engineering
Evaluation - 2, Course: CSE 204: Computer Data Structure and Algorithms - I Sessional (Sec-A)

Sample Input

```
Greetings!!!

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 1

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 1

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 2

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 3

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
```

Sample Input (cont.)

```
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 3

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 1
Enter soldier height: 4

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 5
Lineup: [1, 1, 2, 3, 3, 4, ]

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 2
Enter soldier index: 2

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 5
Lineup: [1, 1, 3, 3, 4, ]
```

Sample Input (cont.)

```
Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 3
Even army: Lineup: [1, 3, 4, ]
Odd army: Lineup: [1, 3, ]

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 5
Lineup: [1, 1, 3, 3, 4, ]

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 4

Options:
1. Insert a Solder
2. Execute the Traitor
3. Split the Force
4. Purge Duplicates
5. ShowLineups
6. Exit
Select your option: 5
Lineup: [1, 3, 4, ]
```

Marking Criteria	Marks
ExecuteTheTraitor() implementation	06
SplitArmy() implementation	07
PurgeDuplicates() implementation	07