Competitive Coding

Level 0 output:

```
1. Insert at head
2. Insert at tail
3. Traverse backward
4. Traverse backward
5. Exit
Enter your choice: 1
Enter data to insert at head:
9. Insert at tail
1. Insert at head
1. Traverse forward
4. Traverse backward
5. Exit
Enter your choice: 1
Enter data to insert at head:
9. Insert at head
1. Insert at head
1. Insert at head
1. Insert at head
2. Insert at tail
3. Traverse backward
5. Exit
Enter your choice: 1
Enter your choice: 1
Enter data to insert at head:
6. Exit
Enter your choice: 1
Enter data to insert at head:
7. Exit
Enter your choice: 1
Enter data to insert at tail
1. Traverse backward
5. Exit
Enter your choice: 2
Enter your choice: 2
Enter data to insert at tail: 1
1. Insert at head
1. Insert at head
1. Insert at head
2. Insert at tail
3. Traverse backward
5. Exit
Enter your choice: 2
Enter data to insert at tail: 1
1. Insert at head
2. Insert at tail
3. Traverse forward
4. Traverse backward
5. Exit
Enter your choice: 2
Enter data to insert at tail: 1
1. Insert at head
2. Insert at tail
3. Traverse hackward
5. Exit
Enter your choice: 3
Enter your choice: 4
Enter your choice: 5
Enter your choice: 5
Enter your choice: 6
Enter your choice: 7
Enter your choice: 8
Enter your choice: 9
Enter your choice: 9
Enter your choice: 1
Enter your choice: 9
Enter your choice: 9
Enter your choice: 1
Enter your choice: 9
Enter your choice: 9
Enter your choice: 1
Enter your choice: 9
Enter your choice:
```

```
Enter your choice: 2
Enter data to insert at tail: 1

2. Insert at beal

2. Insert at tail

3. Insert at tail

3. Insert are forward

4. Insert at head

2. Insert at tail: 5

1. Insert at head

2. Insert at tail

3. Insert at tail

3. Insert at tail

3. Insert as forward

4. Insert at tail: 8

1. Insert at tail: 8

2. Insert at
```

Level 1:

Output:

```
1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 1
Enter value to push: 3
  1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 1
Enter value to push: 4
 1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 2
Popped value: 4
  1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 3
Top value: 3
 1. Push
2. Pop
3. Top
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 2
Popped value: 4
    1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 3
Top value: 3
    1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 4
Minimum value: 3
    1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
6. Exit
Enter your choice: 5
Maximum value: 3
    1. Push
2. Pop
3. Top
4. Get Minimum
5. Get Maximum
```

Level 2:

Output:

```
1. Add interval
2. Get intervals
3. Exit
Enter your choice: 1
Enter start of interval: 1 5
Enter end of interval:
1. Add interval
2. Get intervals
3. Exit
Enter your choice: 1
Enter start of interval: 6 8
Enter end of interval:
1. Add interval
2. Get intervals
3. Exit
Enter your choice: 1
Enter start of interval: 4 7
Enter end of interval:
1. Add interval
2. Get intervals
3. Exit
Enter your choice: 2
Current intervals: [[1, 8]]
1. Add interval
2. Get intervals
3. Exit
Enter your choice:
```

Level 3:

Output:

```
    Set key-value pair

2. Get value by key
3. Print cache entries
4. Exit
Enter your choice: 1
Enter key: my key
Enter value: Hello, Avishek!
Enter expiry time in seconds: 60
1. Set key-value pair
2. Get value by key
3. Print cache entries
4. Exit
Enter your choice: 2
Enter key: my_key
Value for key 'my_key': Hello,Avishek!
1. Set key-value pair
2. Get value by key
3. Print cache entries
4. Exit
Enter your choice: 3
Index 0:
Index 1:
Index 2:
Index 3:
Index 4:
Key: my_key, Value: Hello, Avishek!, Expiry Time: 1741368909
Index 5:
Index 6:
Index 7:
Index 8:
Index 9:
1. Set key-value pair
2. Get value by key
3. Print cache entries
4. Exit
Enter your choice:
```