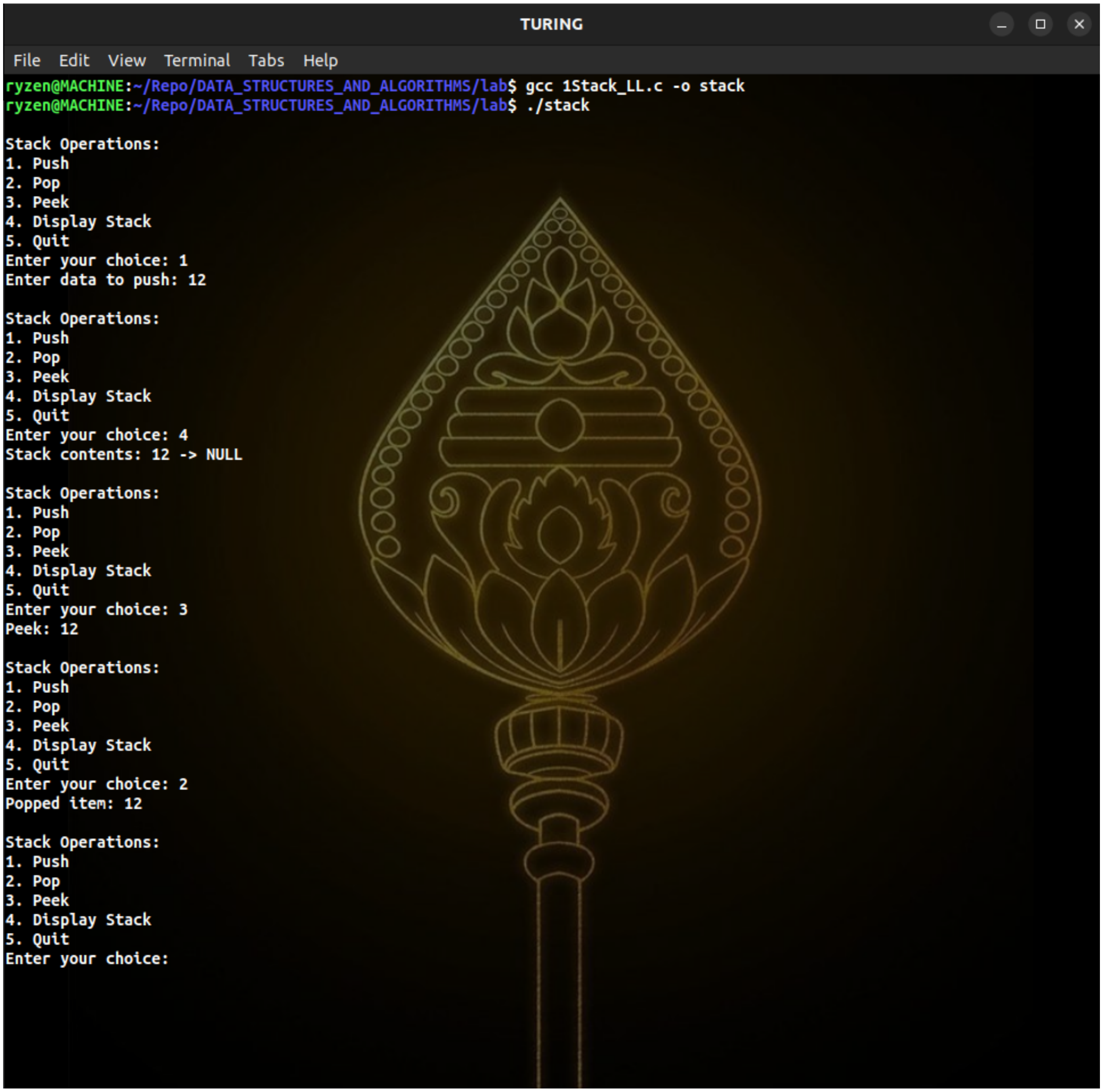
Data Structures and Algorithms

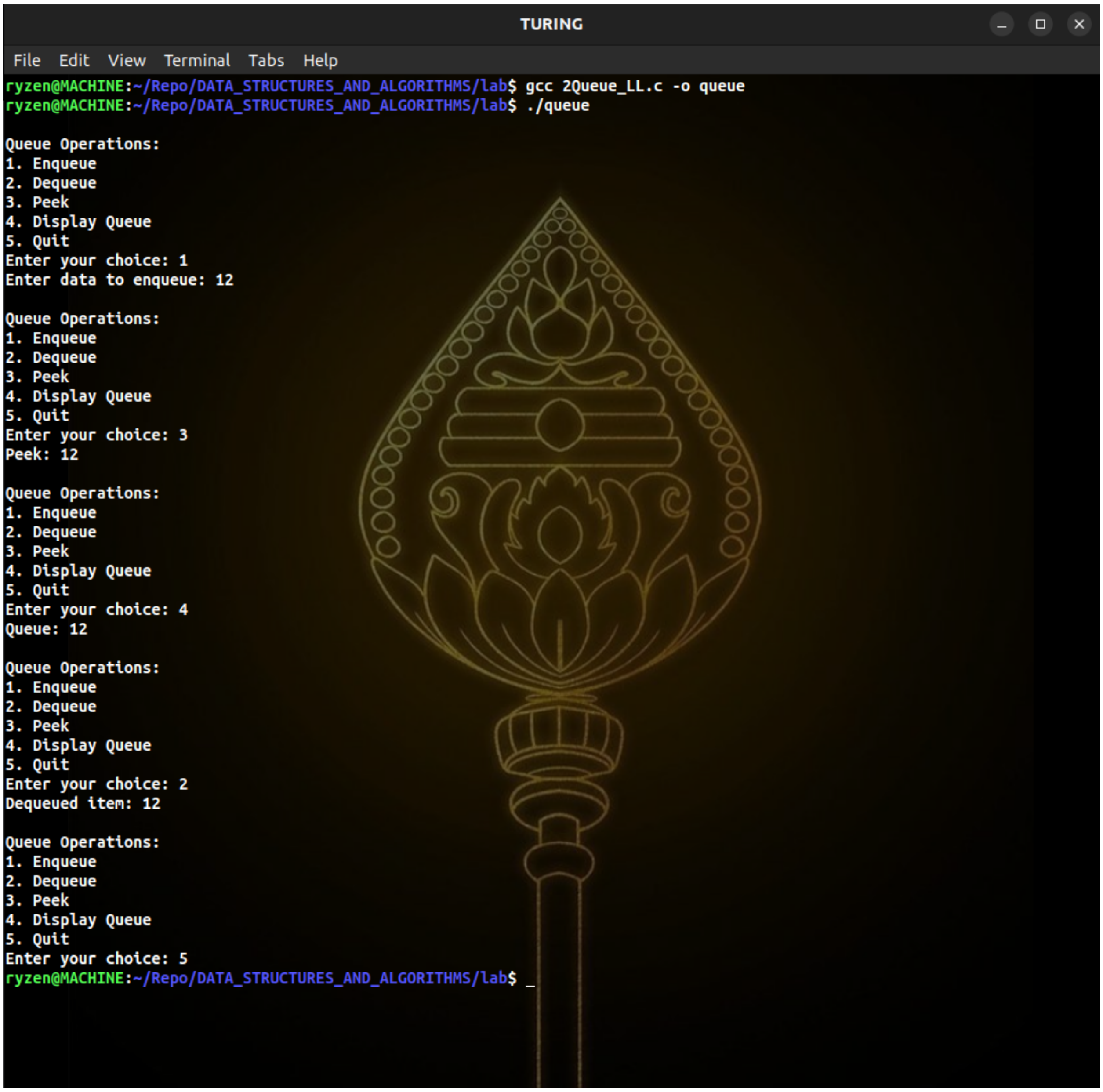
Assignment - II

1. Implement Stack ADT Using Linked List



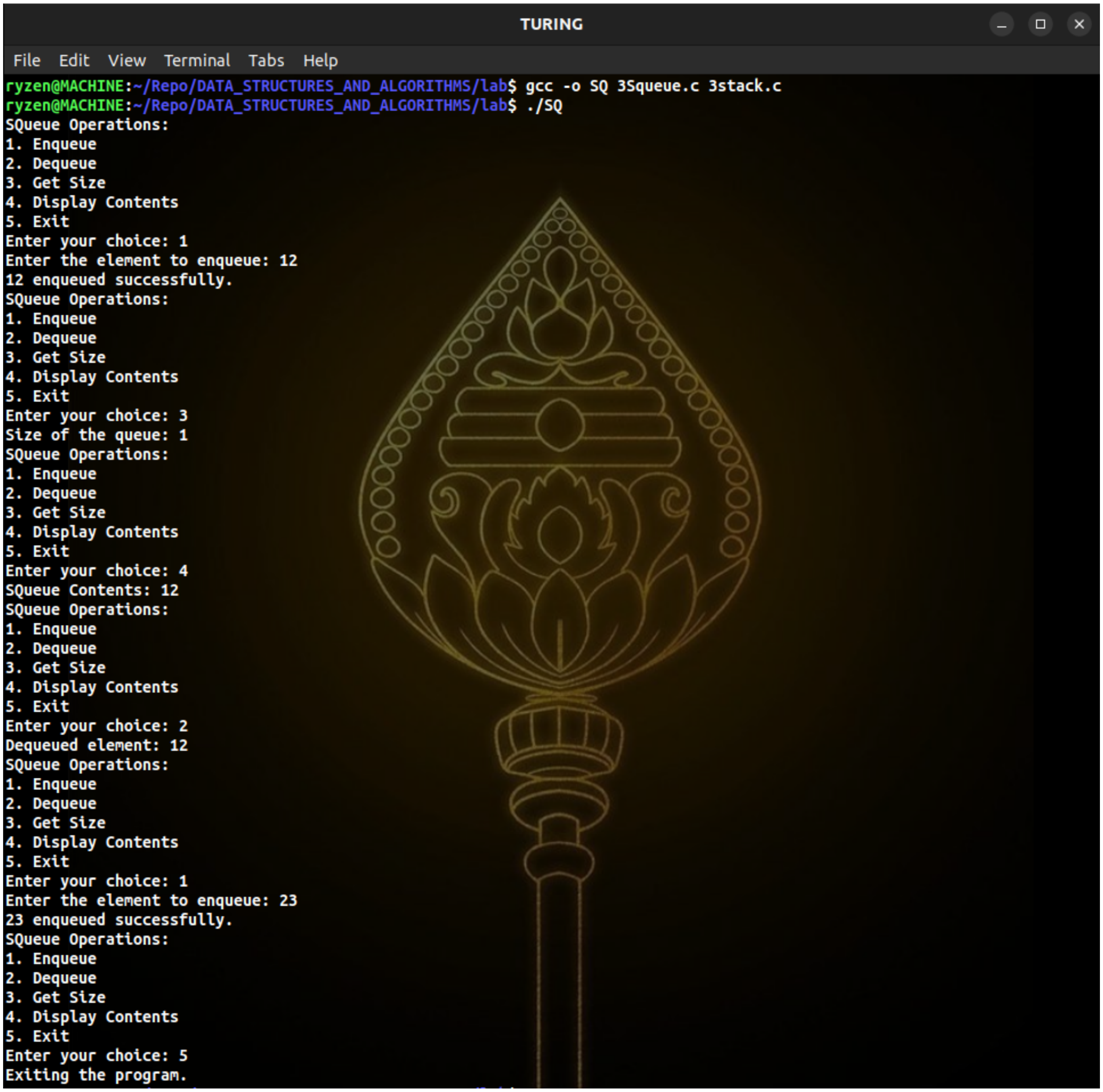
```
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ gcc 1Stack_LL.c -o stack
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ ./stack

Stack Operations:
1. Push
2. Pop
3. Peek
4. Display Stack
5. Quit
Enter your choice: 1
Enter data to push: 12

Stack Operations:
1. Push
2. Pop
3. Peek
4. Display Stack
5. Quit
Enter your choice: 4
Stack contents: 12 -> NULL

Stack Operations:
1. Push
2. Pop
3. Peek
4. Display Stack
5. Quit
Enter your choice: 3
Peek: 12

Stack Operations:
1. Push
2. Pop
3. Peek
4. Display Stack
5. Quit
Enter your choice: 2
Popped item: 12

Stack Operations:
1. Push
2. Pop
3. Peek
4. Display Stack
5. Quit
Enter your choice:
```

2. Implement Queue ADT Using Linked List



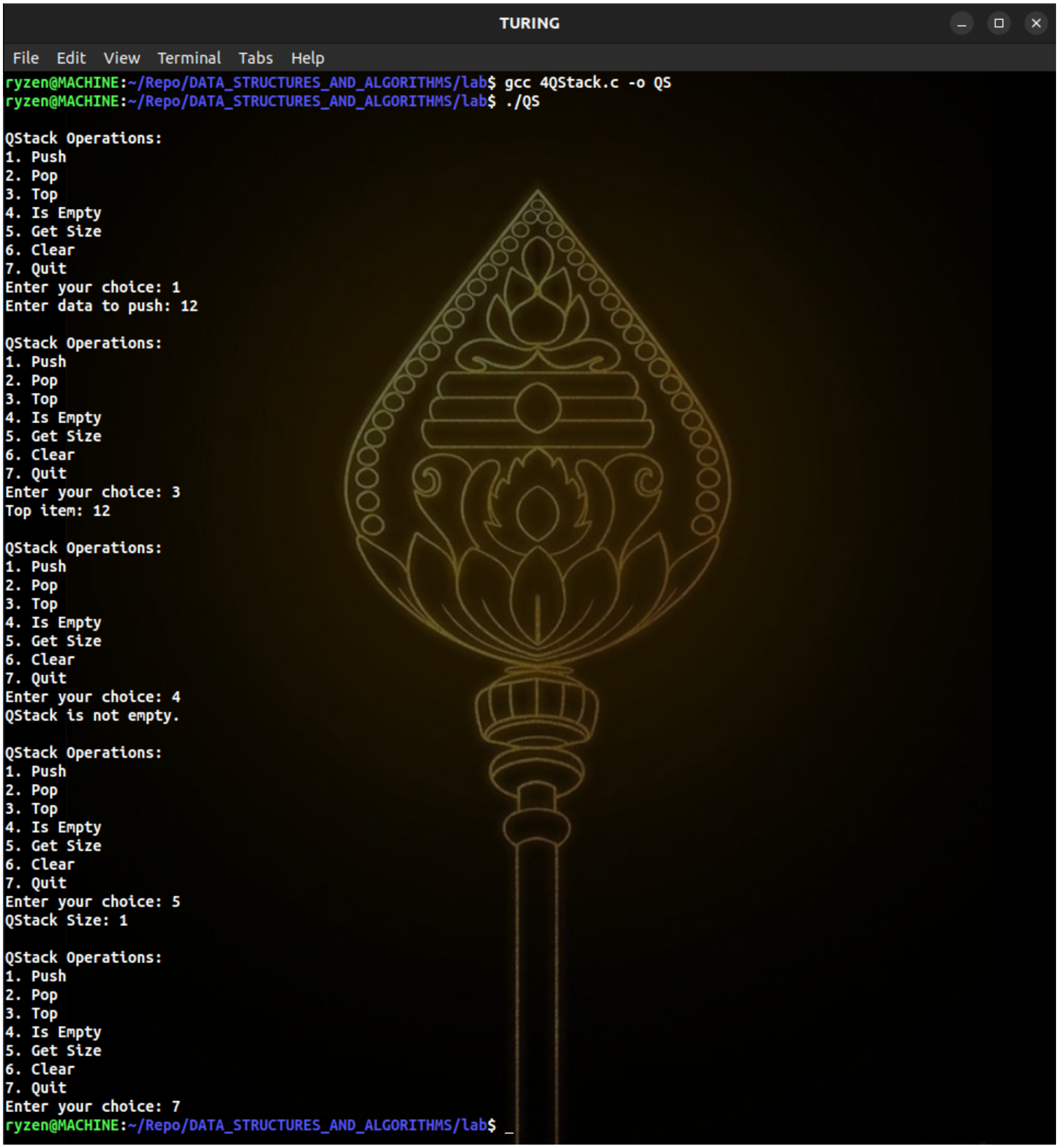
```
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ gcc 2Queue_LL.c -o queue
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ ./queue

Queue Operations:
1. Enqueue
2. Dequeue
3. Peek
4. Display Queue
5. Quit
Enter your choice: 1
Enter data to enqueue: 12

Queue Operations:
1. Enqueue
2. Dequeue
3. Peek
4. Display Queue
5. Quit
Enter your choice: 3
Peek: 12

Queue Operations:
1. Enqueue
2. Dequeue
3. Peek
4. Display Queue
5. Quit
Enter your choice: 4
Queue: 12

Queue Operations:
1. Enqueue
2. Dequeue
3. Peek
4. Display Queue
5. Quit
Enter your choice: 2
Dequeued item: 12

Queue Operations:
1. Enqueue
2. Dequeue
3. Peek
4. Display Queue
5. Quit
Enter your choice: 5
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ _
```

3. Implement Queue Data Structure Called SQueue Using Stack



```
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ gcc -o SQ 3Squeue.c 3stack.c
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ ./SQ
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 1
Enter the element to enqueue: 12
12 enqueued successfully.
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 3
Size of the queue: 1
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 4
SQueue Contents: 12
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 2
Dequeued element: 12
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 1
Enter the element to enqueue: 23
23 enqueued successfully.
SQueue Operations:
1. Enqueue
2. Dequeue
3. Get Size
4. Display Contents
5. Exit
Enter your choice: 5
Exiting the program.
```

4. Implement Stack Data Structure Called QStack Using Queue



```
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ gcc 4QStack.c -o QS
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ ./QS

QStack Operations:
1. Push
2. Pop
3. Top
4. Is Empty
5. Get Size
6. Clear
7. Quit
Enter your choice: 1
Enter data to push: 12

QStack Operations:
1. Push
2. Pop
3. Top
4. Is Empty
5. Get Size
6. Clear
7. Quit
Enter your choice: 3
Top item: 12

QStack Operations:
1. Push
2. Pop
3. Top
4. Is Empty
5. Get Size
6. Clear
7. Quit
Enter your choice: 4
QStack is not empty.

QStack Operations:
1. Push
2. Pop
3. Top
4. Is Empty
5. Get Size
6. Clear
7. Quit
Enter your choice: 5
QStack Size: 1

QStack Operations:
1. Push
2. Pop
3. Top
4. Is Empty
5. Get Size
6. Clear
7. Quit
Enter your choice: 7
ryzen@MACHINE:~/Repo/DATA_STRUCTURES_AND_ALGORITHMS/lab$ _
```