Data : Assignment 3

part1

```
 import javax.swing.JOptionPane;


public class QueueTimer {
	public static void main(String[] args)
	{

	String message = JOptionPane.showInputDialog(null, "Enter size of queue");
	int size;
	long LLen;//enqueue
	long LLde;
	long Aen;//dequeue
	long Ade;



	size = Integer.parseInt(message);
	LLQueue list = new LLQueue();
	CircleQueue Q1 = new CircleQueue(size);

	long Timer = System.nanoTime();

	for(int i =0; i<size; i++) {

		list.enqueue(i);

	}
```

```
Long endTimer = System.nanoTime();
LLen = endTimer - Timer;

Timer = System.nanoTime();

for(int i =0; i<size; i++) {

	list.dequeue();

}

endTimer = System.nanoTime();
LLde = endTimer - Timer;

Timer = System.nanoTime();

for(int i =0; i<size; i++) {

	Q1.enqueue(i);

}

endTimer = System.nanoTime();
Aen = endTimer - Timer;

Timer = System.nanoTime();
```

```
        for(int i =0; i<size; i++) {

                Q1.dequeue();

        }

        endTimer = System.nanoTime();
        Ade = endTimer - Timer;

        System.out.println("LLenqueue:"+LLen);
        System.out.println("LLdequeue:"+LLde);
        System.out.println("Aenqueue:"+Aen);
        System.out.println("Adequeue:"+Ade);

}
}
```

part2

Theoretical complexity of Array and linked list queue implementations

Array enqueue:

to put something into an array we just insert it at the rear position so it is

O(1)

Array dequeue:

to take something out of the array we just remove the object at the read

O(1)

LL enqueue:

to insert an element into the LL we just Insert it after the tail

O(1)

LL dequeue:

to remove an element from the LL we move to the element behind the tail and delete next

O(1)

part 3

strings:	100	1000	10000	100000
Aenqueue time:	15481	228059	842383	3014607
Adequeue time:	9440	87221	420626	1536378
LLenqueue time:	1118017	6648447	139132198	15247080732
LLdequeue time:	289604	5516082	144494604	17186705145

my results dont seem to be quite right, i think that my implementation of my LLqueue is quite bad