Lru cache implementation using hash tables and queues.

**Name USN**

Dweepa 01FB16ECS138

Lakshana 01FB16ECS175

Ishita 01FB16ECS143

Kusumanjali 01FB16ECS173

Chandana 01FB17ECS706

There are **2 programs attached:** lru.c and lru\_copy.c.

- **lru.c** just gives the broad understanding of the program.

- **lru\_copy.c** gives the detailed approach of the program.

**TO EXECUTE:**

**$gcc lru.c**

**$./a.out**

**WHAT THE PROGRAM DOES:**

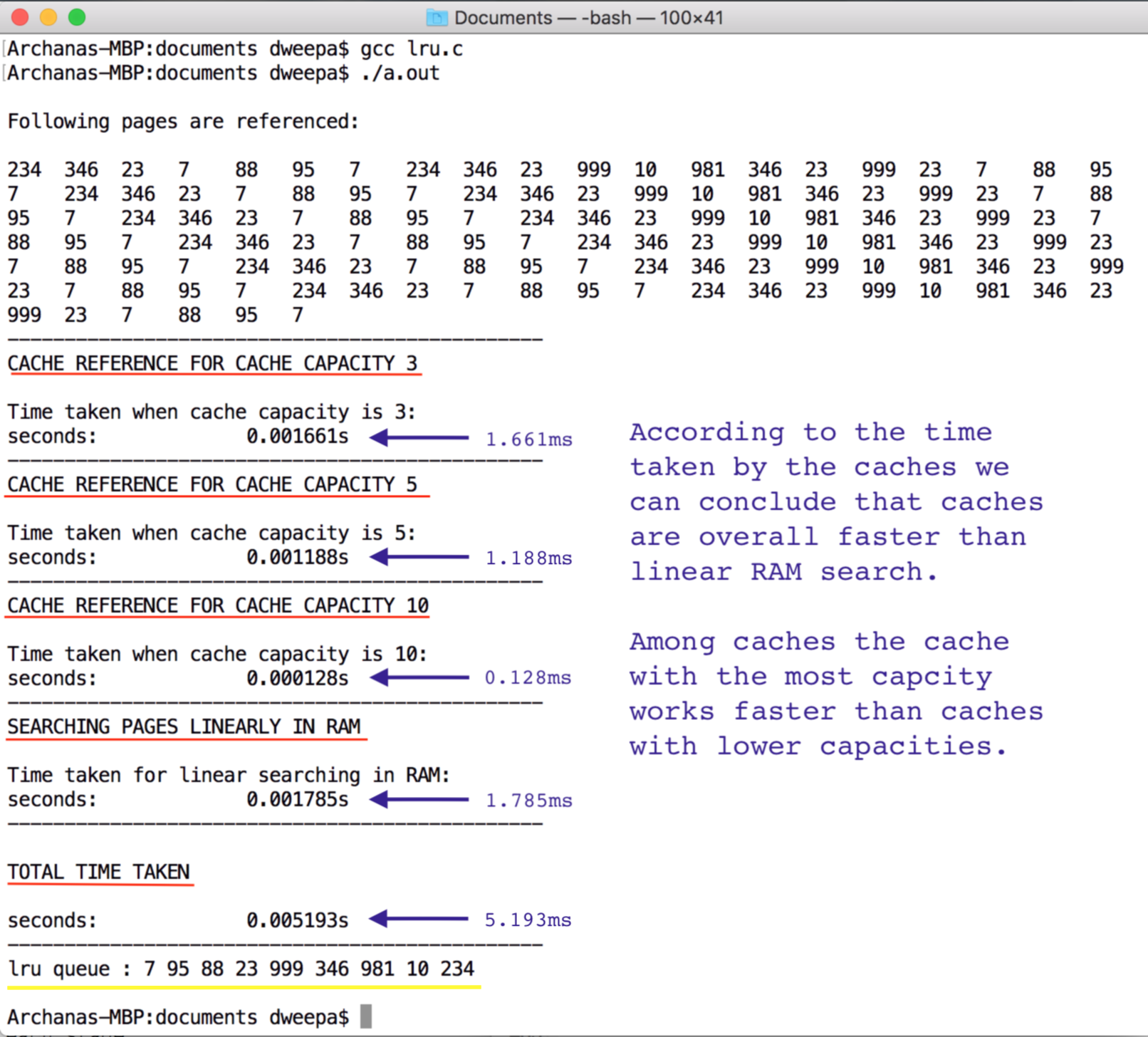
The program mimics a cache using hash tables and a queue. The hash tables provide for easy access of the element and the queue keeps track of the most recently and least recently used elements.

This program times the references from caches with 3 different capacities and direct linear searching through ram to see which is the most efficient.

1. Cache capacity:3 - slowest among all 3 caches.

2. Cache capacity:5 - faster than the previous one but slower than the next one.

3. Cache capacity:10 - fastest among all 3 caches.

4. RAM linear search - slower than all the cache

**TO EXECUTE lru\_copy.c**

**$gcc lru\_copy.c**

**$./a.out**

When prompted to enter the cache capacity please enter the capacities 3, 5 and 10 to see each of their working.

Enter 0 after entering all the three cache capacities to see the integrated result.

Here are some screenshots:

