CS 378h, Spring 2026, Laboratory 3
Transaction Count

Assigned: Tuesday, March 31, 2026
Due: Thursday, April 9, 2026, by 9:30 am

1 Introduction

In this lab, you will learn about the memory-performance effects of high transaction-count procedures. Given a set of transaction database entries and a set of transaction queries, this laboratory asks you to write C-language code to compute the number transaction queries that appear in a transaction database. Thus, this laboratory investigates the number of transaction queries that appear in a *database* of transactions.

The challenge to this laboratory is to manage the large number of transaction entries (˜1 billion) and the also large number (˜10 to 100 million) of transaction queries. Given a *database* of transaction entries, your programs needs to return the total number of transaction queries that are present in the database.

The challenge in this laboratory is to arrange the look-ups in a manner that reduces the time required process the entire set of transaction queries.

2 Logistics

You are expected to work on this lab alone. However, you may communicate with others concerning your understanding of C code and the various tools; e.g., you may discuss the compiler, assembler, linker, loader, and other systems issues. The results that you submit in response to laboratory must be created and provided by you alone.

Any clarifications and revisions to the laboratory will be posted on the top-level course webpage. By Monday, April 6, 2026, we will post several test file pairs (a database file and a queries file) to serve as a test corpus. Any clarifications and revisions to the laboratory will be posted on the top-level course webpage.

3 Transaction Counting Program

Your transaction-count program should accept two file arguments, such as with a program named **trans-cnt**:

```
trans-cnt <database> <queries>
```

You program should return a natural number with the aggregate total number of times the queries values appear in the database. Consider the database file:

```
3
9
7
3
5
```

and the query file:

```
3
7
4
7
```

For this database and these queries, the overall count (answer) should be: 4.

Your task in this laboratory is to write a program that can accept up to one billion (2^{30}) database items (natural numbers) where each database item (number) and each query (number) will be between 0 and 2^{50}. And, when given up to one hundred million queries (natural numbers), your program should return the aggregate total number of times each query appears in its corresponding database.

4 Evaluation

You are expected to write a report that explains timing results of running your programs with varying-sized arguments. In addition, for each database-queries pair that are provided, you need to report your best times. We should be able to replicate your results by compiling your programs and running your program on the database and query pairs we provide.

Your entire write-up should be quite succinct, shorter than 100 lines of C-language comment. In addition, you must submit your code. The class TA will post how to do this. Your write-up should be included as a C-language-style comment at the front of your solution – and it must in ASCII only be 100 lines or less in length! Longer write-ups will have points deducted!

By Monday (April 6th), the **CS378h** TA will post several pairs (database and queries) of test input files.

The maximum score for this laboratory is 100 points. The value of the individual components is as follows.

- The numeric result (a single natural number) for each dataset-queries pair provided (there will be two or three such pairs).
- An explanation of your timing results data (10 points).
- An explanation of the code that you wrote that documents your approach and rationale for copying solutions (30 points). This explanation should include a discussion of how one can assure the correctness of your implementation especially when the source and destination addresses overlap.

5 Hints

Work incrementally. Don't try to do everything at once. Get your code to function correctly, and then improve its performance.

6 Hand In Instructions

Please follow the instructions below for turning in your work.

- Make sure you have included your identifying information in your submission. In this case, you will update the file you submit with your name, UTID. Submit only a single C-language file that contains both your report (in ASCII) and your code.
- The solution file you submit must be named:

  `<YourUTID>-transactions.c.`

The **CS378h** TA will post instructions as to how to submit your laboratory report by using **CANVAS**.

Remember: you code must compile on the machines we are using for this class! Be sure to confirm that the single C-language file you submit for this Laboratory will compile and that it can accept two file names (a database file and queries file pair).