# **Test Document — Zip Code Group Project 3.0**

**Course:** CSCI 331-54  
**Instructor:** Andrew Anda  
**Group Members:** Kiran Kadariya, Kevin G, Sagar N, Abdulaziz A  
**Submission Date:** 04/09/2025

## **Test Cases**

### ✅ Test Case 1: Generate Blocked Sequence Set File

**Objective:**  
Verify that the program reads records from a CSV file and generates a .txt file organized into length-indicated blocks with correct metadata.

**Steps:**

1. Ensure zip\_codes.csv is located in ./resources/.
2. Run the program without any flags:
3. ./zip\_project
4. Open and inspect the output file in ./output/blocked\_sequence\_set.txt.

**Expected Result:**

* The file begins with a packed header.
* Block count is shown.
* Each block includes:
  + Block number
  + Record count
  + Next block pointer
  + Length-indicated record strings

**Actual Result:** ✅ Pass

### ✅ Test Case 2: Dump Blocks in Physical Order

**Objective:**  
Verify that blocks are displayed in the order they are stored on disk.

**Steps:**

1. Run the program with:
2. ./zip\_project --dumpPhysical
3. Review the console output.

**Expected Output:**

* Each block displays:
  + Block Number: N
  + Next Block: N+1 or -1
  + A list of records (index, city, state, county)

**Actual Result:** ✅ Pass

### ✅ Test Case 3: Dump Blocks in Logical Order

**Objective:**  
Ensure the program correctly follows the nextBlock pointer to list blocks logically.

**Steps:**

1. Run:
2. ./zip\_project --dumpLogical
3. Verify that blocks are output by following their nextBlock links (not file order).

**Expected Output:**

* Blocks appear in linked order.
* For an uninterrupted chain, physical and logical dumps should match.

**Actual Result:** ✅ Pass

### ✅ Test Case 4: Buffer and Serialization Accuracy

**Objective:**  
Verify that buffer packing/unpacking maintains data integrity.

**Steps:**

1. Check that each record is stored using a length prefix.
2. Inspect Buffer.cpp's output strings.
3. Confirm that unpacked records match original content.

**Expected Behavior:**

* Record format: "23,1001,Agawam,MA,Hampden"
* After unpacking: "1001,Agawam,MA,Hampden"

**Actual Result:** ✅ Pass

### ✅ Test Case 5: Header Format and Block Metadata

**Objective:**  
Validate that the file header and block headers contain appropriate metadata.

**Steps:**

1. Open blocked\_sequence\_set.txt.
2. Review:
   * File header line (e.g., BSS,1.0,...)
   * Block headers: "blockNumber,recordCount,nextBlock"

**Expected Format:**

"39,BSS,1.0,HEADER\_SIZE,BlockSize,NumBlocks"

"13645" <-- Number of blocks

"90,0,3,1" <-- Block 0: 3 records, points to Block 1

**Actual Result:** ✅ Pass

## **Test Summary**

| **Test Case** | **Status** |
| --- | --- |
| Generate Blocked Sequence Set | ✅ Pass |
| Dump Blocks in Physical Order | ✅ Pass |
| Dump Blocks in Logical Order | ✅ Pass |
| Buffer Pack/Unpack Correctness | ✅ Pass |
| Header and Block Metadata Validation | ✅ Pass |

## **Conclusion**

All tests confirm that **Part 3 of the Zip Code Group Project** performs as specified:

* Records are accurately grouped into blocks.
* Each block includes correct metadata and packed records.
* Dump methods work correctly in both physical and logical order.
* Buffer classes maintain proper string packing and unpacking.