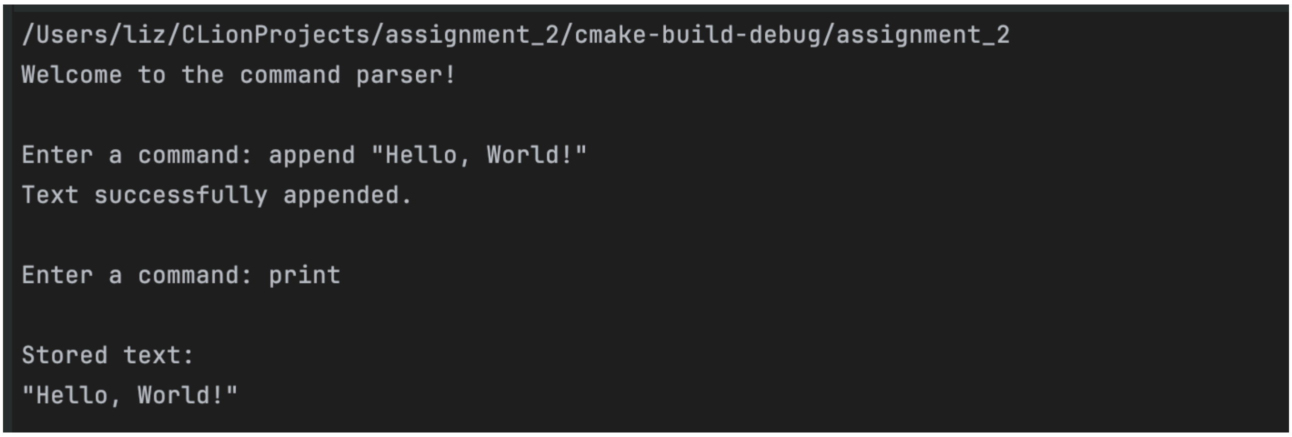
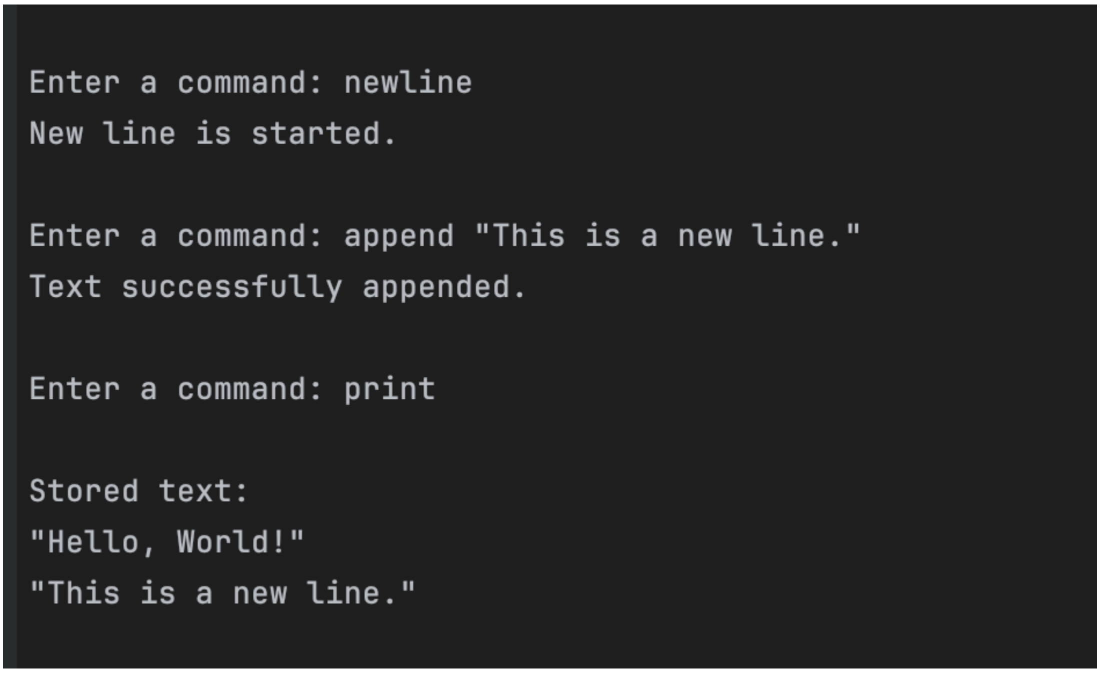
Test Scenarios

Scenario 1: Basic Text Operations
1. Append and Print
   - Command: `append "Hello, World!"`
   - Expected Result: Text contains "Hello, World!"
   - Command: `print`
   - Expected Result: Output shows "Hello, World!»



```
/Users/liz/CLionProjects/assignment_2/cmake-build-debug/assignment_2
Welcome to the command parser!

Enter a command: append "Hello, World!"
Text successfully appended.

Enter a command: print

Stored text:
"Hello, World!"
```

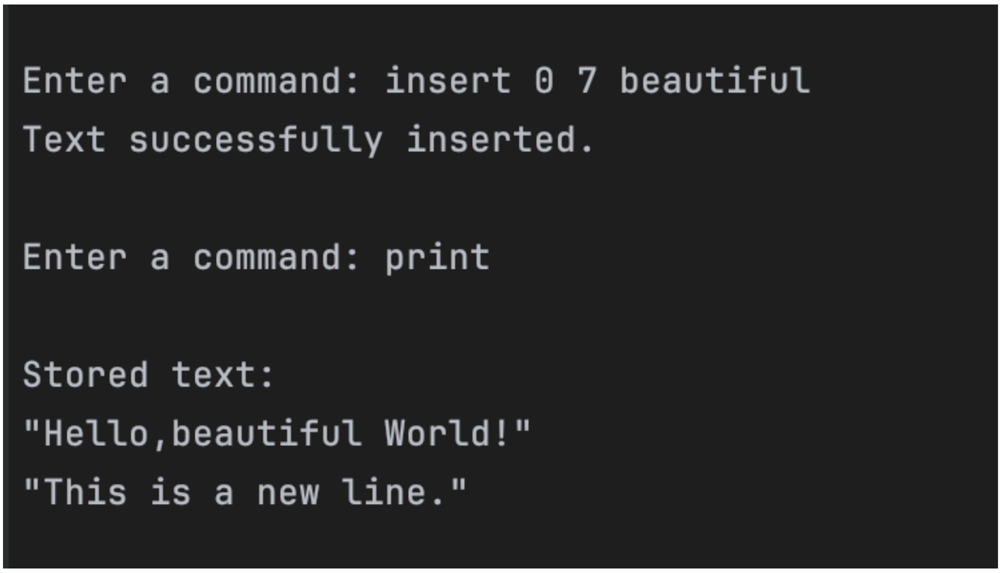
2. Start New Line and Append
   - Command: `newline`
   - Command: `append "This is a new line."`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     Hello, World!
     This is a new line.
     ```



```
Enter a command: newline
New line is started.

Enter a command: append "This is a new line."
Text successfully appended.

Enter a command: print

Stored text:
"Hello, World!"
"This is a new line."
```

3. Insert Text
   - Command: `insert 0 7 "beautiful "`
   - Command: `print`

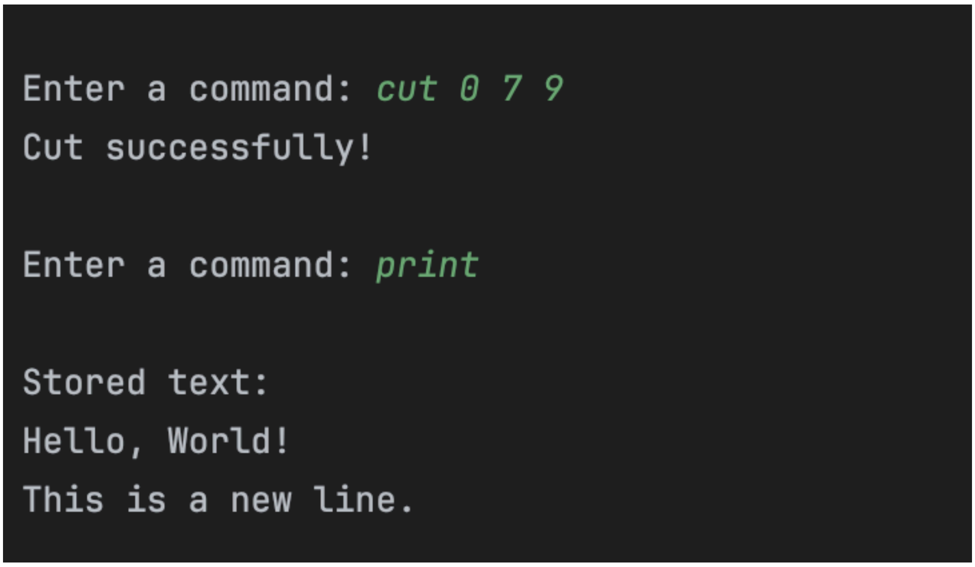
  - Expected Result: Output shows:
    ```
    Hello, beautiful World!
    This is a new line.
    ```



```
Enter a command: insert 0 7 beautiful
Text successfully inserted.

Enter a command: print

Stored text:
"Hello,beautiful World!"
"This is a new line."
```

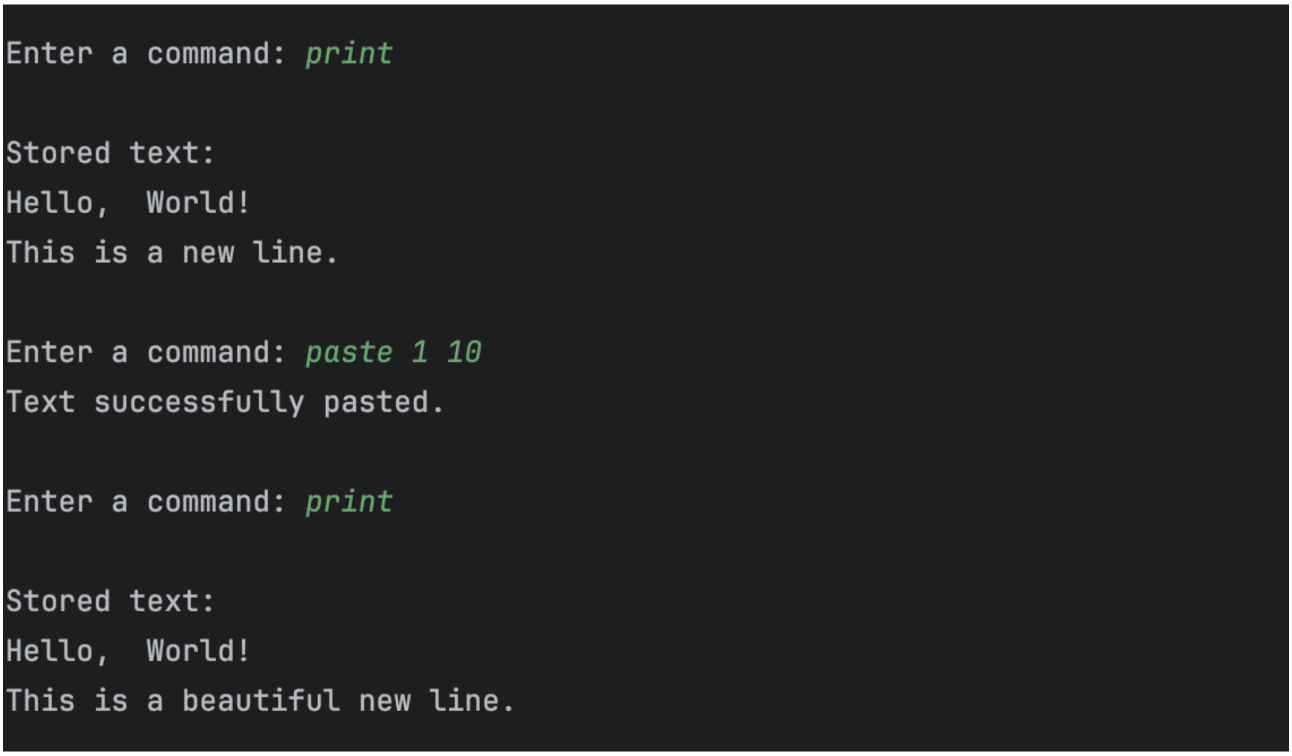
Scenario 2: Cut, Copy, Paste Operations

1. Cut Text
  - Command: `cut 0 7 9`
  - Command: `print`
  - Expected Result: Output shows:
    ```
    Hello, World!
    This is a new line.
    ```



```
Enter a command: cut 0 7 9
Cut successfully!

Enter a command: print

Stored text:
Hello, World!
This is a new line.
```

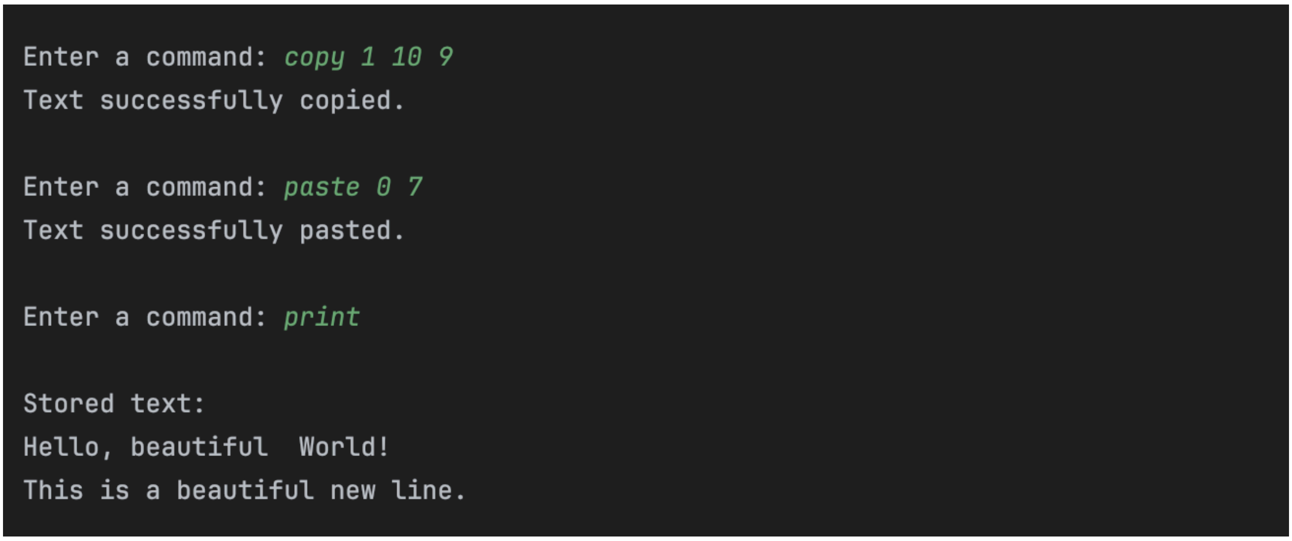
2. Paste Text
  - Command: `paste 1 10`
  - Command: `print`
  - Expected Result: Output shows:
    ```
    ```

Hello, World!
This is a beautiful new line.
```



```
Enter a command: print

Stored text:
Hello,  World!
This is a new line.

Enter a command: paste 1 10
Text successfully pasted.

Enter a command: print

Stored text:
Hello,  World!
This is a beautiful new line.
```

3. Copy Text and Paste
    - Command: `copy 1 10 9`
    - Command: `paste 0 7`
    - Command: `print`
    - Expected Result: Output shows:

```
Hello, beautiful World!
This is a beautiful new line.
```



```
Enter a command: copy 1 10 9
Text successfully copied.

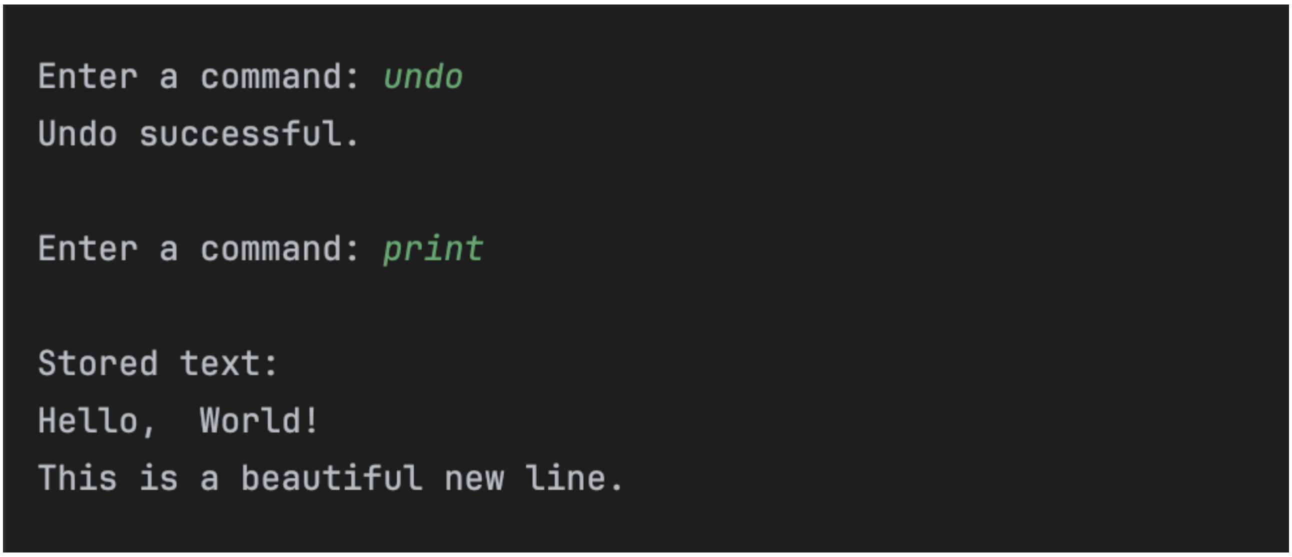
Enter a command: paste 0 7
Text successfully pasted.

Enter a command: print

Stored text:
Hello, beautiful  World!
This is a beautiful new line.
```
```

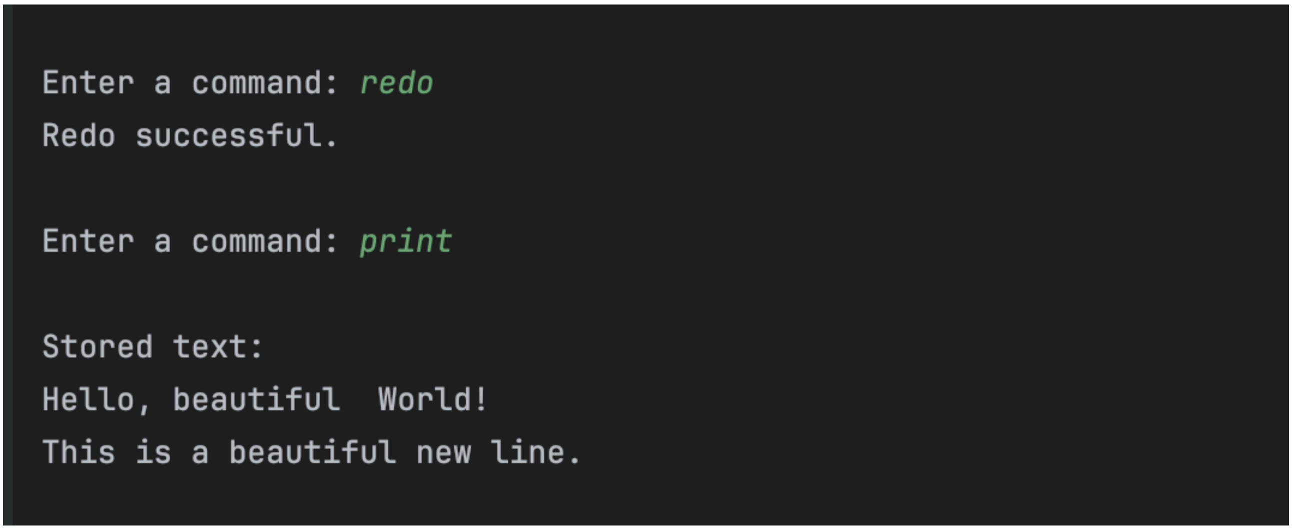
Scenario 3: Undo and Redo Operations

1. Undo Last Paste
    - Command: `undo`
    - Command: `print`
    - Expected Result: Output shows:
      ```
      Hello, World!
      This is a beautiful new line.
      ```



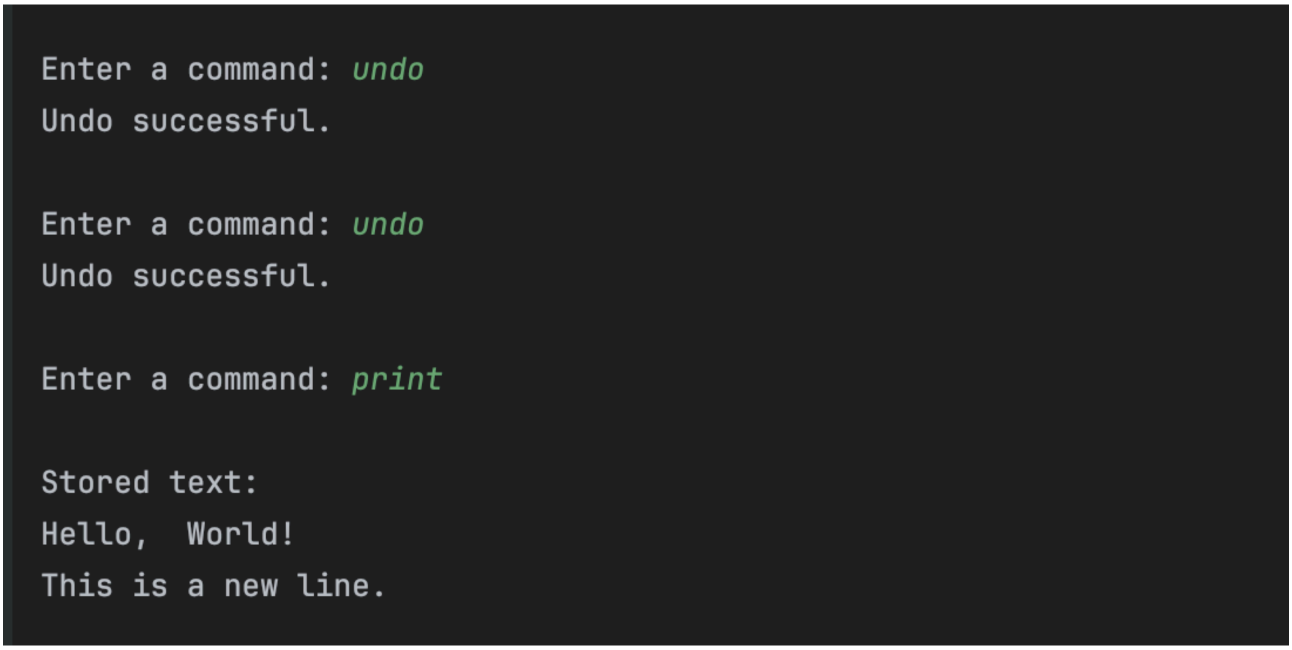
```
Enter a command: undo
Undo successful.

Enter a command: print

Stored text:
Hello,  World!
This is a beautiful new line.
```

2. Redo Last Paste
    - Command: `redo`
    - Command: `print`
    - Expected Result: Output shows:
      ```
      Hello, beautiful World!
      This is a beautiful new line.
      ```



```
Enter a command: redo
Redo successful.

Enter a command: print

Stored text:
Hello, beautiful  World!
This is a beautiful new line.
```

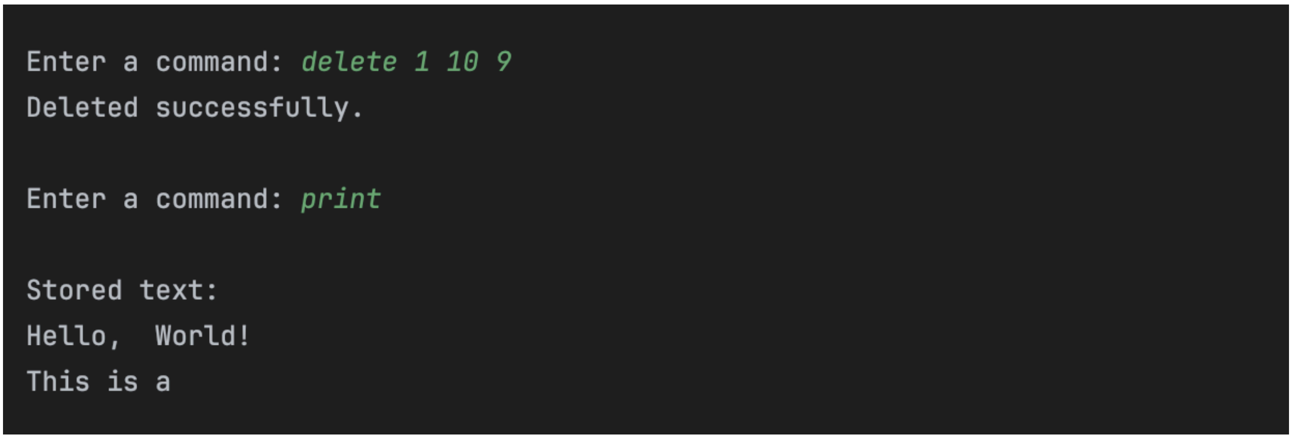
3. Undo Insert and Cut
   - Command: `undo`
   - Command: `undo`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     Hello, World!
     This is a new line.
     ```



```
Enter a command: undo
Undo successful.

Enter a command: undo
Undo successful.

Enter a command: print

Stored text:
Hello,  World!
This is a new line.
```

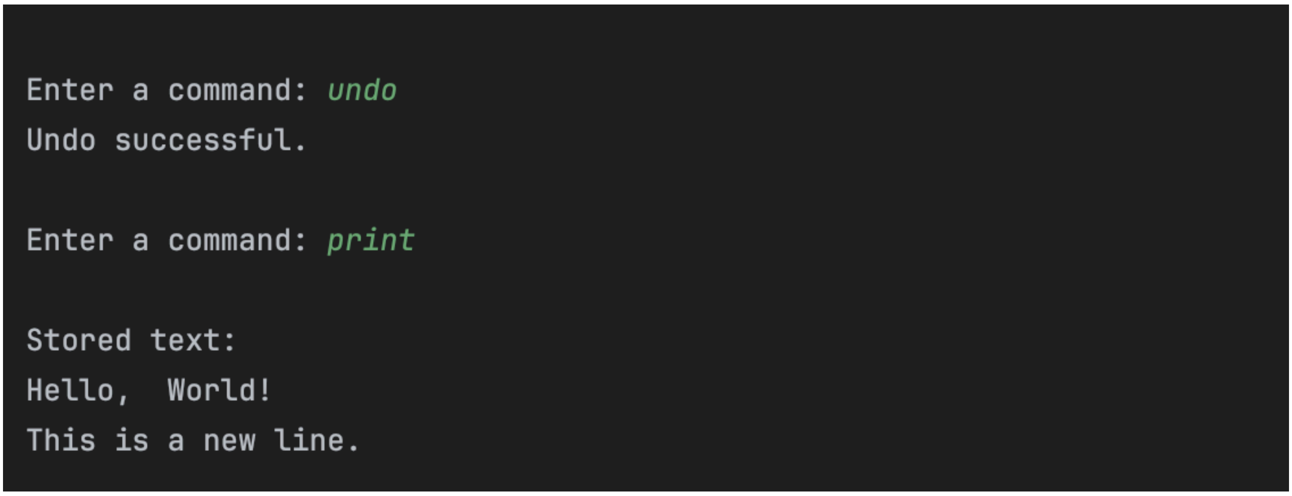
Scenario 4: Delete Text

1. Delete Text
   - Command: `delete 1 10 9`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     Hello, World!
     This is a
     ```



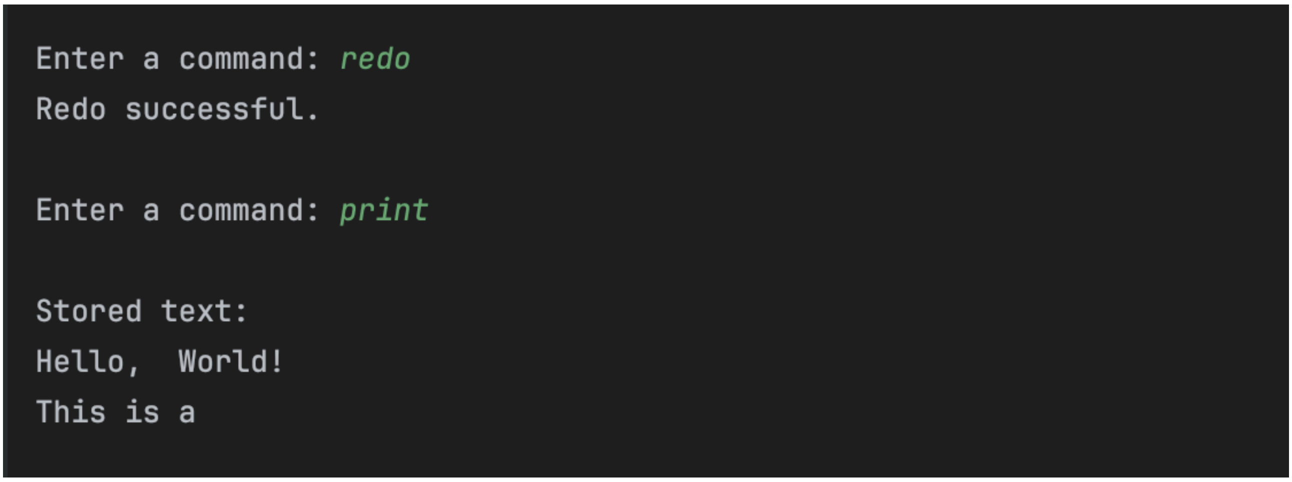
```
Enter a command: delete 1 10 9
Deleted successfully.

Enter a command: print

Stored text:
Hello,  World!
This is a
```

2. **Undo and Redo Delete**
  - Command: `undo`
  - Command: `print`
  - Expected Result: Output shows:
    ```
    Hello, World!
    This is a new line.
    ```



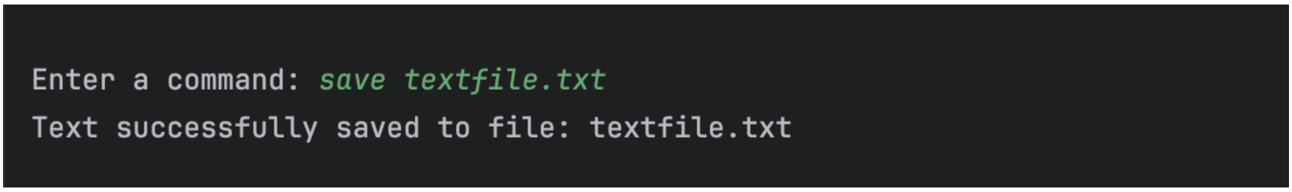
```
Enter a command: undo
Undo successful.

Enter a command: print

Stored text:
Hello,  World!
This is a new line.
```

  - Command: `redo`
  - Command: `print`
  - Expected Result: Output shows:
    ```
    Hello, World!
    This is a
    ```



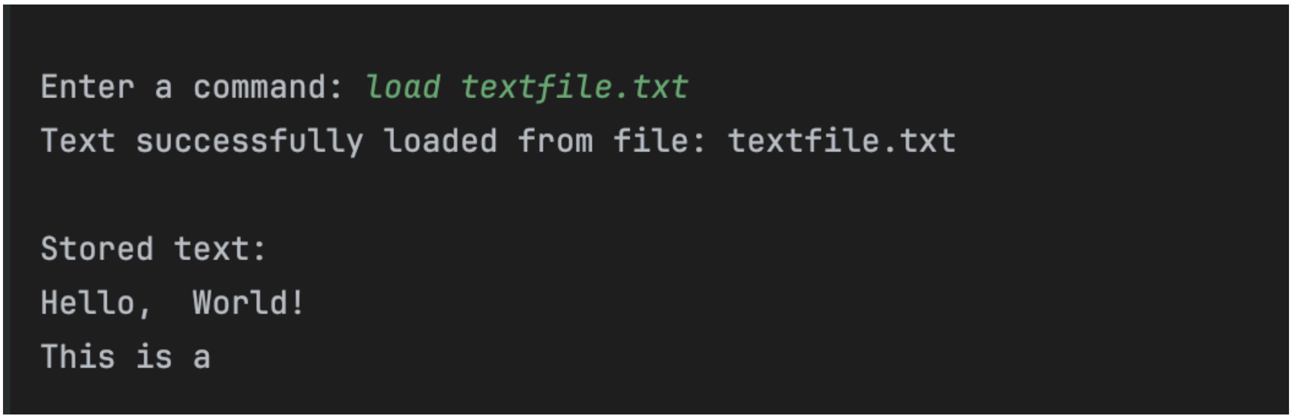
```
Enter a command: redo
Redo successful.

Enter a command: print

Stored text:
Hello,  World!
This is a
```

Scenario 5: Save and Load File
1. Save to File
  - Command: `save testfile.txt`
  - Expected Result: File "testfile.txt" is created with the current text content.



```
Enter a command: save textfile.txt
Text successfully saved to file: textfile.txt
```

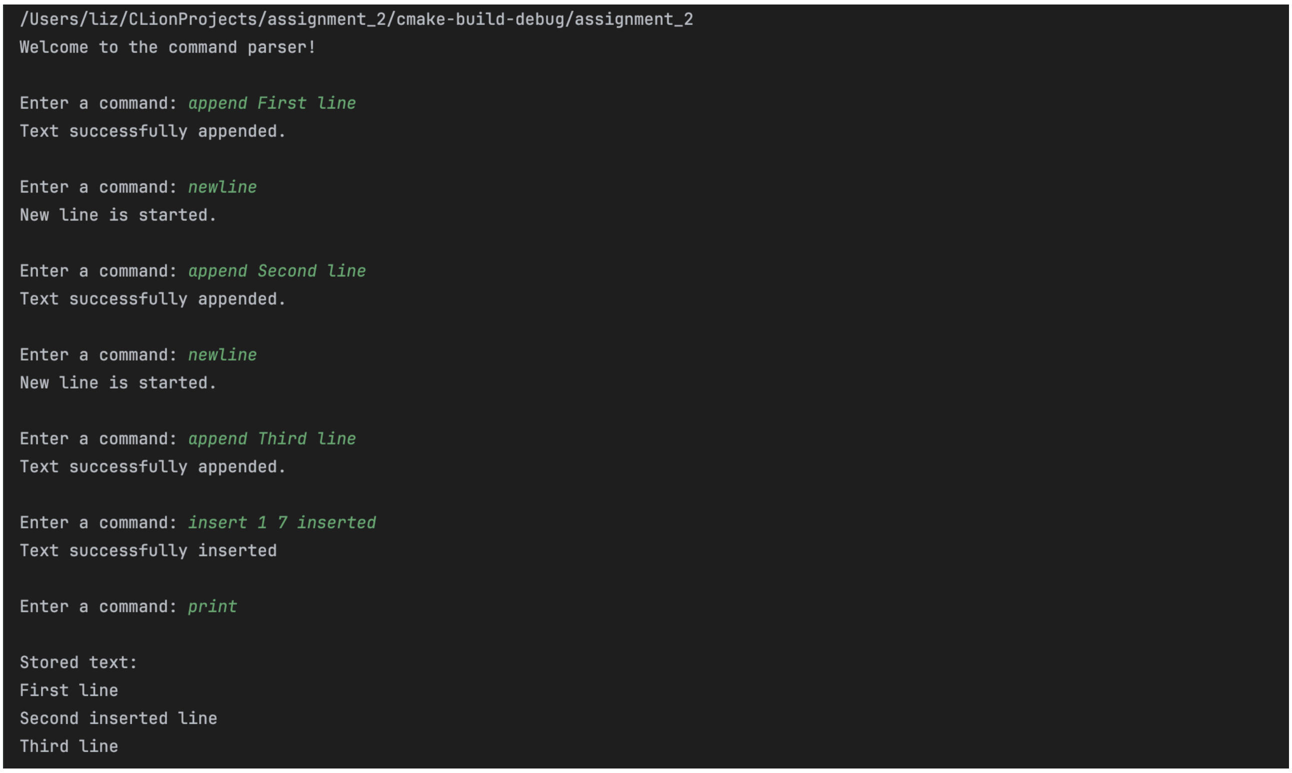
2. Load from File
   - Command: `load testfile.txt`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     Hello, World!
     This is a
     ```



```
Enter a command: load textfile.txt
Text successfully loaded from file: textfile.txt

Stored text:
Hello,  World!
This is a
```

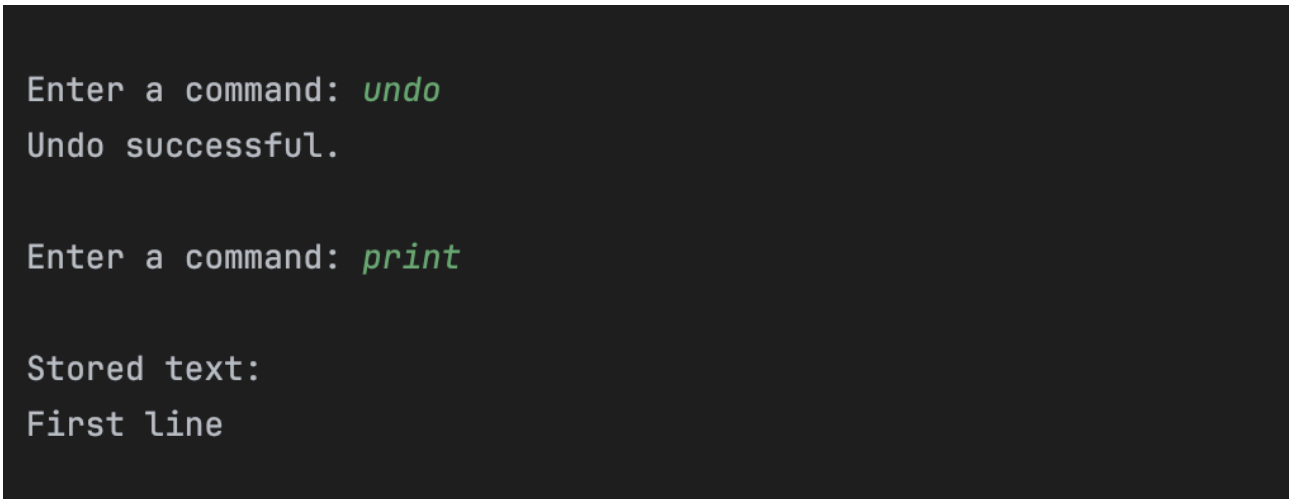
Scenario 6: Multiple Edits and Undo/Redo

1. Append Multiple Lines and Insert Text
   - Command: `append "First line."`
   - Command: `newline`
   - Command: `append "Second line."`
   - Command: `newline`
   - Command: `append "Third line."`
   - Command: `insert 1 7 " inserted"`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     First line.
     Second inserted line.
     Third line.
     ```



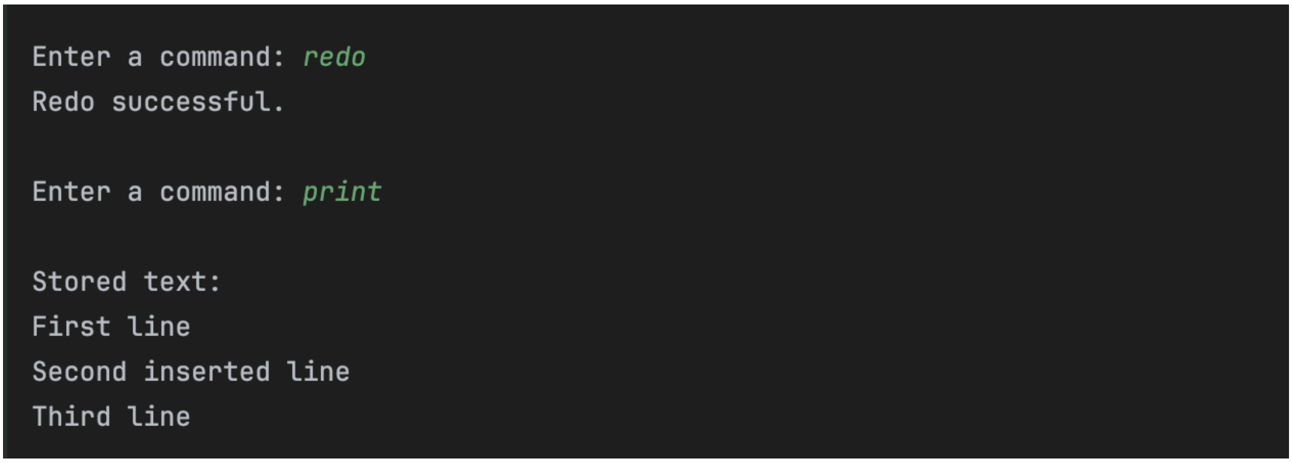
```
/Users/liz/CLionProjects/assignment_2/cmake-build-debug/assignment_2
Welcome to the command parser!

Enter a command: append First line
Text successfully appended.

Enter a command: newline
New line is started.

Enter a command: append Second line
Text successfully appended.

Enter a command: newline
New line is started.

Enter a command: append Third line
Text successfully appended.

Enter a command: insert 1 7 inserted
Text successfully inserted

Enter a command: print

Stored text:
First line
Second inserted line
Third line
```

2. Undo Insert and New Lines
   - Command: `undo`
   - Command: `undo`
   - Command: `undo`
   - Command: `undo`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     First line.
     ```



```
Enter a command: undo
Undo successful.

Enter a command: print

Stored text:
First line
```

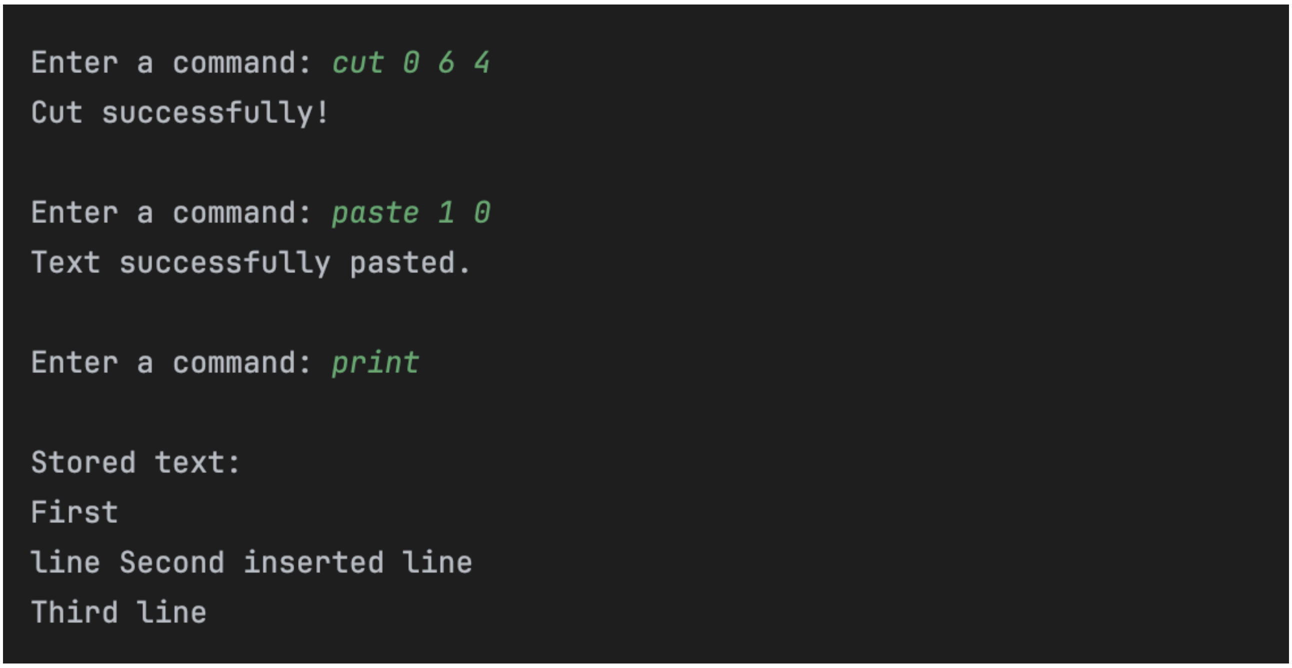
3. Redo New Lines and Insert
   - Command: `redo`
   - Command: `redo`
   - Command: `redo`
   - Command: `print`
   - Expected Result: Output shows:
     ```
     First line.
     Second inserted line.
     Third line.
     ```



```
Enter a command: redo
Redo successful.

Enter a command: print

Stored text:
First line
Second inserted line
Third line
```

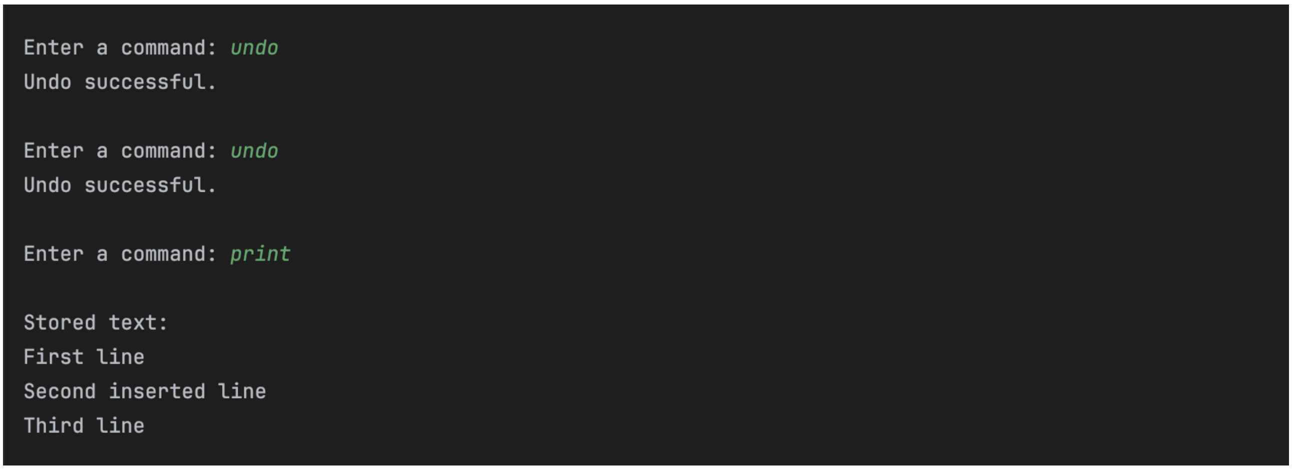
Scenario 7: Copy, Cut, Paste with Undo/Redo

1. Cut and Paste Text
    - Command: `cut 0 6 4`
    - Command: `paste 1 0`
    - Command: `print`
    - Expected Result: Output shows:
      ```
      First
      line.Second inserted line.
      Third line.
      ```



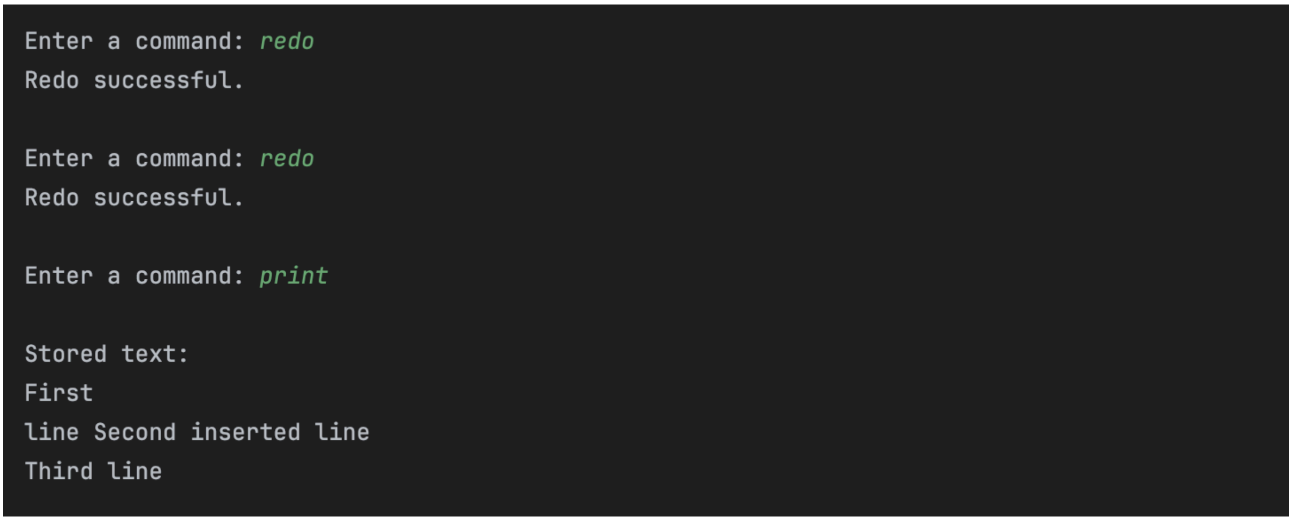
```
Enter a command: cut 0 6 4
Cut successfully!

Enter a command: paste 1 0
Text successfully pasted.

Enter a command: print

Stored text:
First
line Second inserted line
Third line
```

2. Undo Cut and Paste
    - Command: `undo`
    - Command: `undo`
    - Command: `print`
    - Expected Result: Output shows:
      ```
      First line.
      Second inserted line.
      Third line.
      ```



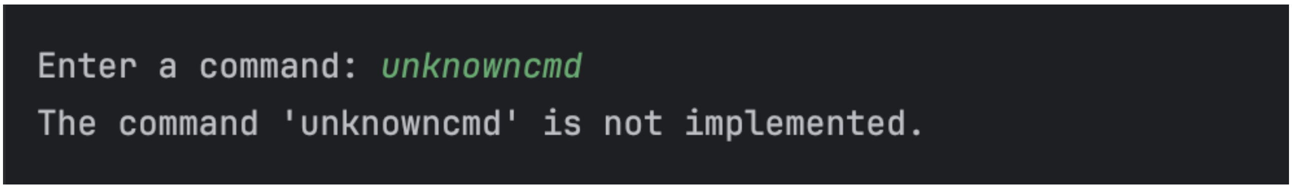
```
Enter a command: undo
Undo successful.

Enter a command: undo
Undo successful.

Enter a command: print

Stored text:
First line
Second inserted line
Third line
```

3. Redo Cut and Paste
   - Command: `redo`
   - Command: `redo`
   - Command: `print`
   - Expected Result: Output shows:
   ```
   First
   line.Second inserted line.
   Third line.
   ```



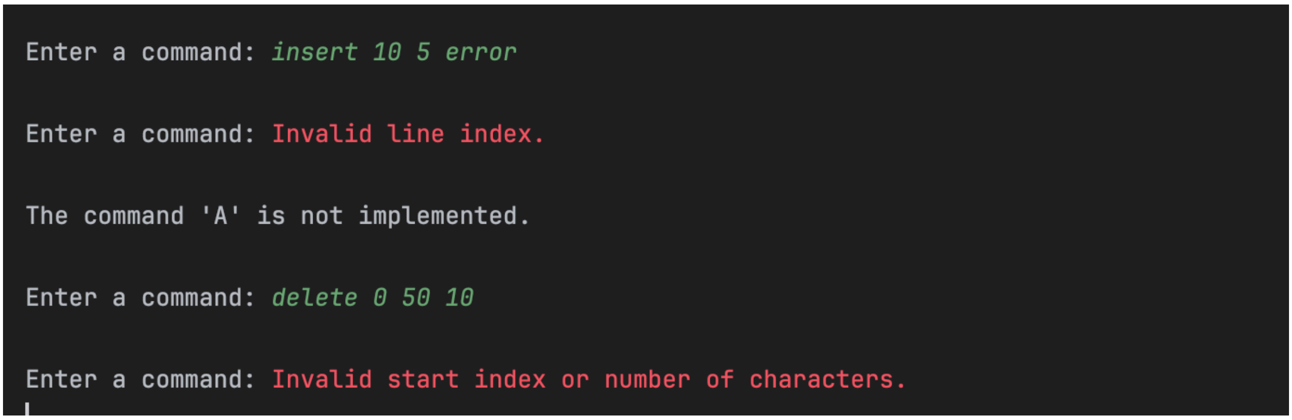
```
Enter a command: redo
Redo successful.

Enter a command: redo
Redo successful.

Enter a command: print

Stored text:
First
line Second inserted line
Third line
```

Scenario 8: Error Handling
1. Invalid Commands
   - Command: `unknowncmd`
   - Expected Result: Error message indicating the command is not implemented.



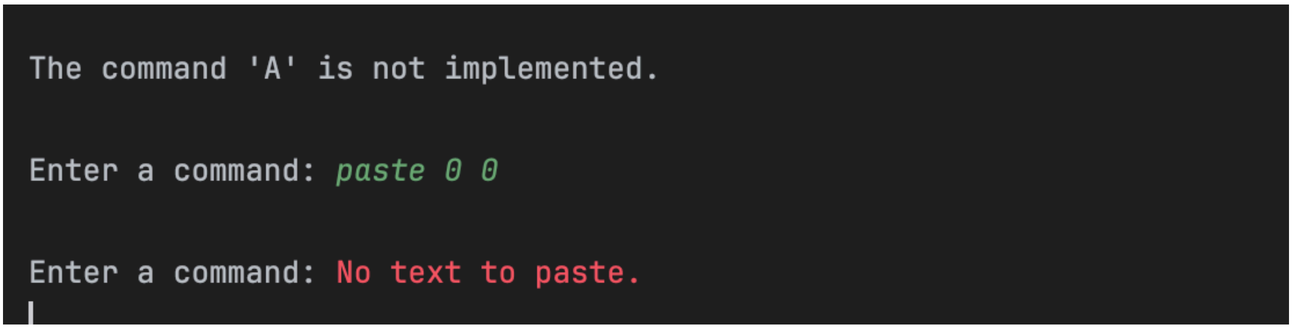
```
Enter a command: unknowncmd
The command 'unknowncmd' is not implemented.
```

2. Out of Bound Indices
   - Command: `insert 10 5 "error"`
   - Expected Result: Error message indicating invalid line index.
   - Command: `delete 0 50 10`
   - Expected Result: Error message indicating invalid character index.



```
Enter a command: insert 10 5 error

Enter a command: Invalid line index.

The command 'A' is not implemented.

Enter a command: delete 0 50 10

Enter a command: Invalid start index or number of characters.
```

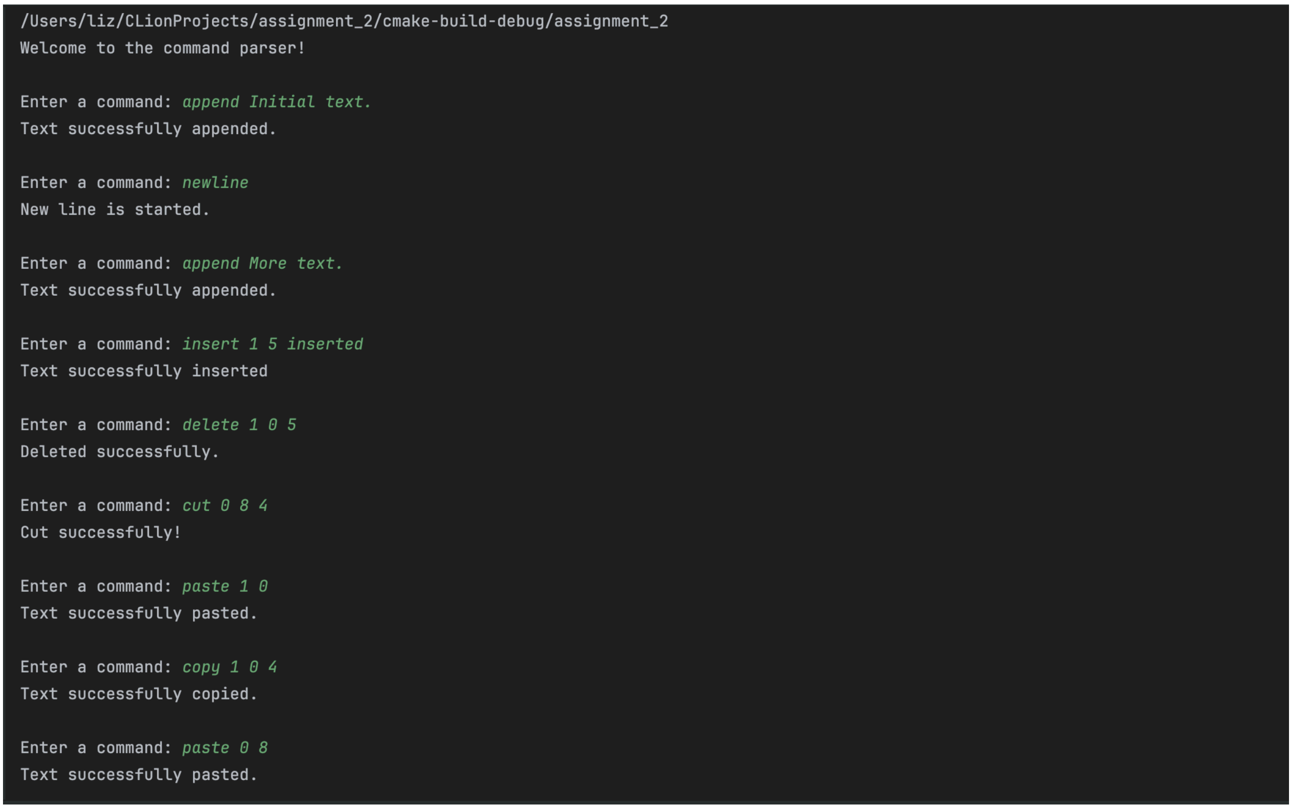
3. Empty Clipboard Operations
    - Command: `paste 0 0`
    - Expected Result: Error message indicating no text to paste.



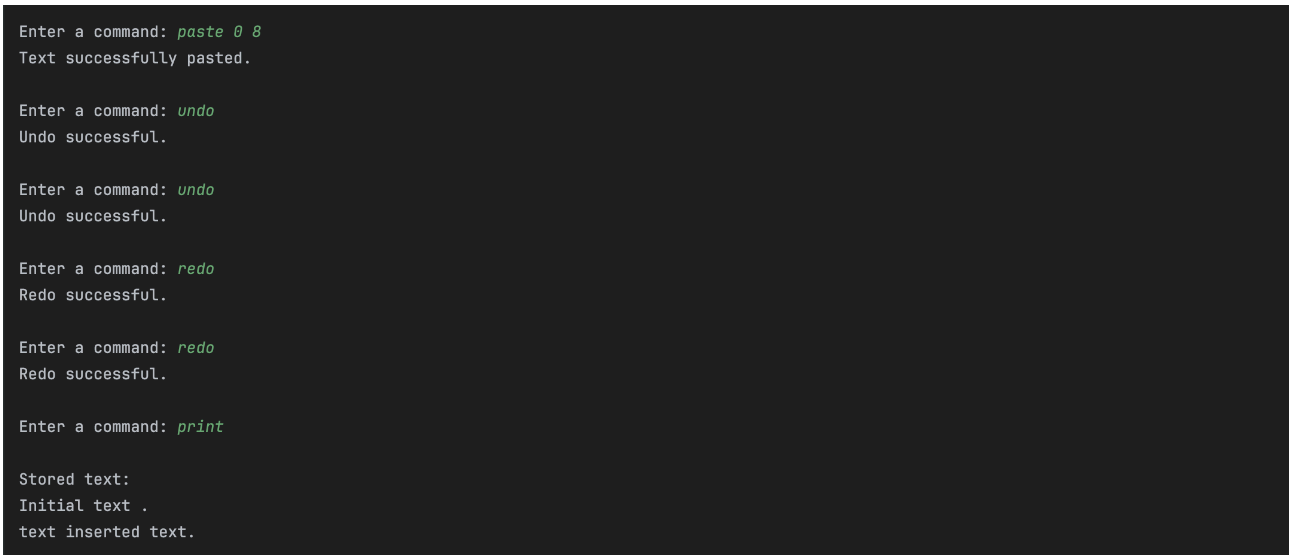
```
The command 'A' is not implemented.

Enter a command: paste 0 0

Enter a command: No text to paste.
```

Scenario 9: Complex Edit Sequence
1. Append, Insert, Delete, Cut, Copy, Paste, Undo, Redo**
    - Command: `append "Initial text."`
    - Command: `newline`
    - Command: `append "More text."`
    - Command: `insert 1 5 " inserted"`
    - Command: `delete 1 0 5`
    - Command: `cut 0 8 4`
    - Command: `paste 1 0`
    - Command: `copy 1 0 4`
    - Command: `paste 0 8`
    - Command: `undo`
    - Command: `undo`
    - Command: `redo`
    - Command: `redo`
    - Command: `print`
    - Expected Result: Output shows:

      ```
      Initial text.
      text inserted text.
      ```



```
/Users/liz/CLionProjects/assignment_2/cmake-build-debug/assignment_2
Welcome to the command parser!

Enter a command: append Initial text.
Text successfully appended.

Enter a command: newline
New line is started.

Enter a command: append More text.
Text successfully appended.

Enter a command: insert 1 5 inserted
Text successfully inserted

Enter a command: delete 1 0 5
Deleted successfully.

Enter a command: cut 0 8 4
Cut successfully!

Enter a command: paste 1 0
Text successfully pasted.

Enter a command: copy 1 0 4
Text successfully copied.

Enter a command: paste 0 8
Text successfully pasted.
```



```
Enter a command: paste 0 8
Text successfully pasted.

Enter a command: undo
Undo successful.

Enter a command: undo
Undo successful.

Enter a command: redo
Redo successful.

Enter a command: redo
Redo successful.

Enter a command: print

Stored text:
Initial text .
text inserted text.
```