1. Converting Module - String to Integer Conversion (atoi)

Test Case 1:

Input: 12345678901234567890

Expected Output: Large integer 12345678901234567890

---

Test Case 2:

Input: 12345

Expected Output: Integer 12345

---

Test Case 3:

Input: abcd

Expected Output: 0 (Invalid string)

---

Test Case 4:

Input: 1 2 3 4 5

Expected Output: 12345 (Converts valid numbers)

---

Test Case 5:

Input: "q"

Expected Output: Ends program or returns 0 (quit condition)

---

2. Fundamentals Module - Indexing Strings

Test Case 1:

Input: buffer1 = "Hello World", numInput = "4"

Expected Output: Character at index 4: 'o'

---

Test Case 2:

Input: buffer1 = "Hello World", numInput = "10"

Expected Output: Too big... Position reduced to max. available (Last character 'd')

---

Test Case 3:

Input: buffer1 = "Hello World", numInput = "11"

Expected Output: Invalid index or error message for out-of-range input

---

Test Case 4:

Input: buffer1 = "Hello World", numInput = "-1"

Expected Output: Invalid index or error for negative index

---

Test Case 5:

Input: buffer1 = "Hello World", numInput = ""

Expected Output: Empty string input, error message

---

Test Case 6:

Input: buffer1 = "Hello World", numInput = "abc"

Expected Output: Non-numeric input, should return 0 or handle error gracefully

---

Test Case 7:

Input: buffer1 = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz", numInput = "81"

Expected Output: Character at position 81: 'b' (for large strings)

---

Test Case 8:

Input: buffer1 = "Hello World", numInput = "12345678901"

Expected Output: Out of range, max position adjusted

---

Test Case 9:

Input: buffer1 = "Q", numInput = "0"

Expected Output: Character at index 0: 'Q'

---

Test Case 10:

Input: buffer1 = "q", numInput = "0"

Expected Output: Character at index 0: 'q'

---

3. Manipulating Module - String Concatenation

Test Case 1:

Input: string1 = "a", string2 = "b"

Expected Output: "ab"

---

Test Case 2:

Input: string1 = "ThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10ln", string2 = "ThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10lnThisIs10ln"

Expected Output: Long concatenated string

---

Test Case 3:

Input: string1 = "My Name Is", string2 = "ddd"

Expected Output: "My Name Isddd"

---

Test Case 4:

Input: string1 = "My Name Is", string2 = " ddd"

Expected Output: "My Name Is ddd"

---

Test Case 5:

Input: string1 = "Q", string2 = "q"

Expected Output: "Qq"

---

4. Tokenizing Module - Tokenizing Words

Test Case 1:

Input: "q"

Expected Output: Ends program or tokenizes "q"

---

Test Case 2:

Input: "aa"

Expected Output: Tokenizes "aa"

---

Test Case 3:

Input: "The quick brown fox"

Expected Output: "The", "quick", "brown", "fox"

---

Test Case 4:

Input: "Hello world"

Expected Output: "Hello", "world"

---

Test Case 5:

Input: " "

Expected Output: Empty tokens due to double spaces

---

Test Case 6:

Input: "123 456"

Expected Output: "123", "456" (Tokens separated by tab)

---

Edge Case Testing:

- Converting Module: Handle invalid strings (e.g., "abcd", "12 34") and large numbers (e.g., 12345678901234567890).

- Fundamentals Module: Test with large strings (over 100 characters) and negative indices.

- Manipulating Module: Test string concatenation with very long strings.

- Tokenizing Module: Test tokenization with multiple spaces and tabs as delimiters.