Test Report for wrap\_it.py Program

\*\*Tester:\*\* [Your Name]

\*\*Date:\*\* [Insert Date]

\*\*Program:\*\* wrap\_it.py

\*\*Description:\*\* This program accepts a string of text and an integer as the maximum wrap limit. It outputs the input text wrapped so that no line exceeds the specified limit, followed by the marker --END OF FILE--.

## Example Test

\*\*Test Purpose:\*\* To confirm the program behaves as shown in the specification example.

\*\*Input:\*\*

- Text: "a quick brown fox jumps over the lazy dog"

- Wrap limit: 26

\*\*Expected Output:\*\*

a quick brown fox jumps  
over the lazy dog  
--END OF FILE--

\*\*Outcome:\*\* ✅ Passed

\*\*Notes:\*\* Output matched expectations. The line break correctly occurs before the word 'over'.

## Test 1: Empty String

\*\*Test Purpose:\*\* To verify how the program handles an empty input string.

\*\*Input:\*\*

- Text: ""

- Wrap limit: 10

\*\*Expected Output:\*\*

--END OF FILE--

\*\*Outcome:\*\* ✅ Passed

\*\*Notes:\*\* Works consistently for wrap limits of 10, 100, and 1000. Also handles multiple spaces gracefully.

## Test 2: Long Word

\*\*Test Purpose:\*\* To observe behavior when a single word exceeds the wrap limit.

\*\*Input:\*\*

- Text: "antidisestablishmentarianism"

- Wrap limit: 10

\*\*Expected Output:\*\*

antidisestablishmentarianism  
--END OF FILE--

\*\*Outcome:\*\* ❌ Failed

\*\*Notes:\*\* Program crashes with a ValueError. Suggests the code tries to find a space for breaking, but none exist in a single long word.

## Test 3: Punctuation and Spacing

\*\*Test Purpose:\*\* To test how the program handles punctuation.

\*\*Input:\*\*

- Text: "Hello, Python! This is a test."

- Wrap limit: 15

\*\*Expected Output:\*\*

Hello, Python!  
This is a test.  
--END OF FILE--

\*\*Outcome:\*\* ✅ Passed

\*\*Notes:\*\* Correct line breaks and punctuation preserved. Behavior remains consistent with lower wrap limits.

## Test 4: Special Characters

\*\*Test Purpose:\*\* To verify whether special characters impact wrapping.

\*\*Input:\*\*

- Text: "Hello, &!-$., this is a test."

- Wrap limit: 20

\*\*Expected Output:\*\*

Hello, &!-$., this  
is a test.  
--END OF FILE--

\*\*Outcome:\*\* ✅ Passed

\*\*Notes:\*\* Special characters treated as normal characters. No issues observed.

## Test 5: Double Spaces

\*\*Test Purpose:\*\* To observe how the program handles multiple consecutive spaces.

\*\*Input:\*\*

- Text: " a quick brown fox jumps over the lazy dog"

- Wrap limit: 26

\*\*Expected Output:\*\*

a quick brown fox  
jumps over the lazy  
dog.  
--END OF FILE--

\*\*Outcome:\*\* ✅ Passed

\*\*Notes:\*\* Program preserves double spacing, as expected.

# Bug Report: Long Word Crash

\*\*Summary:\*\* Program crashes when processing input consisting of a single long word with no spaces, where the word length exceeds the wrap limit.

\*\*Steps to Reproduce:\*\* Run the command:  
python3 wrap\_it.py 10 "antidisestablishmentarianism"

\*\*Observed Result:\*\*  
ValueError: substring not found

\*\*Expected Result:\*\*  
The entire word should be printed on a single line (even if it exceeds the limit), followed by --END OF FILE--.

\*\*Impact:\*\* High – this causes the program to crash entirely and could affect usability with technical text or identifiers.

\*\*Severity:\*\* High

\*\*Likelihood:\*\* Medium

\*\*Suggested Fix:\*\* Add logic to handle long words by printing them as-is when no space is found before the wrap limit.