***Systems Programming***

***Final Project – SIC/XE Assembler***

***Phase 2***

**Prepared by:**

1. Eman Rafik ID:11
2. Toka Alaa ID:14
3. Nada Salama ID:55
4. Yomna Gamal ID:60

* **Requirements Specifications:**

Implementing pass 2 of two pass SIC/XE assembler by building on the previous phase:

The output of this phase is:

* The Object-Code file.
* The program code with each line ended with its object code.
* A report with all errors in both pass1 and pass2.

The program supports:

* EQU and ORG statements.
* Simple expressions evaluation.
* **Extra Features:**

1. The program deals with literals.

* **Design:**

**In addition to previously mentioned classes and the overall design and flow of source code, there are 4 other classes:**

* **Class Parser\_phase2:**

To parse each line of pass 1 output and get the operation codes and operands and send these parameters to ObjectCode class

* **Class ObjectCode:**

Generate object code for each line by calculating addresses.

* **Class ObjectFile:**

Generate different records for the object file and print them as an output of pass 2.

* **Class Expressions:**

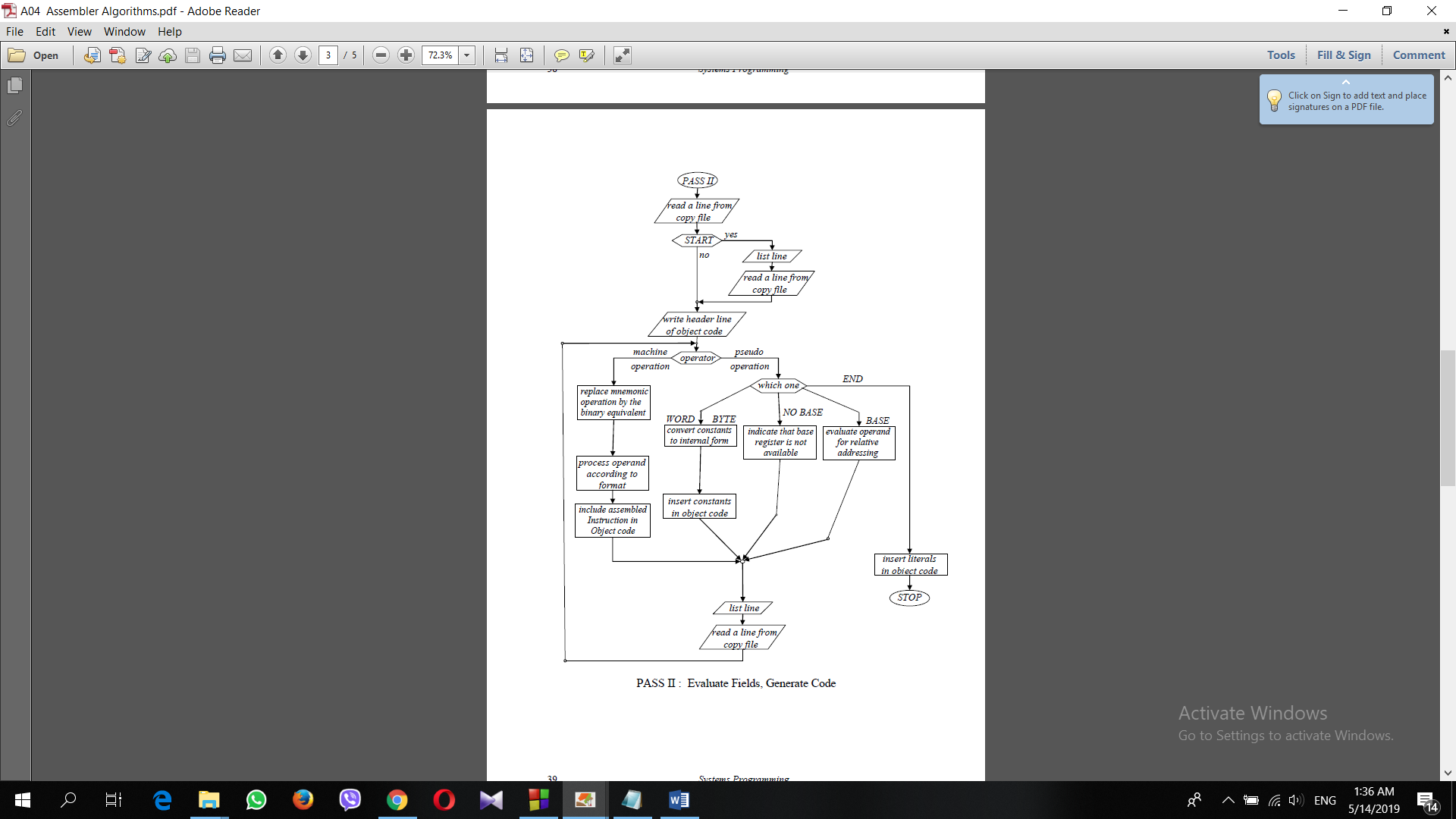
To deal with simple expressions.

* **Main Data Structures:**

Map to store object code of all operations and directives.

Vector to store modification records.

* **Algorithms Description:**



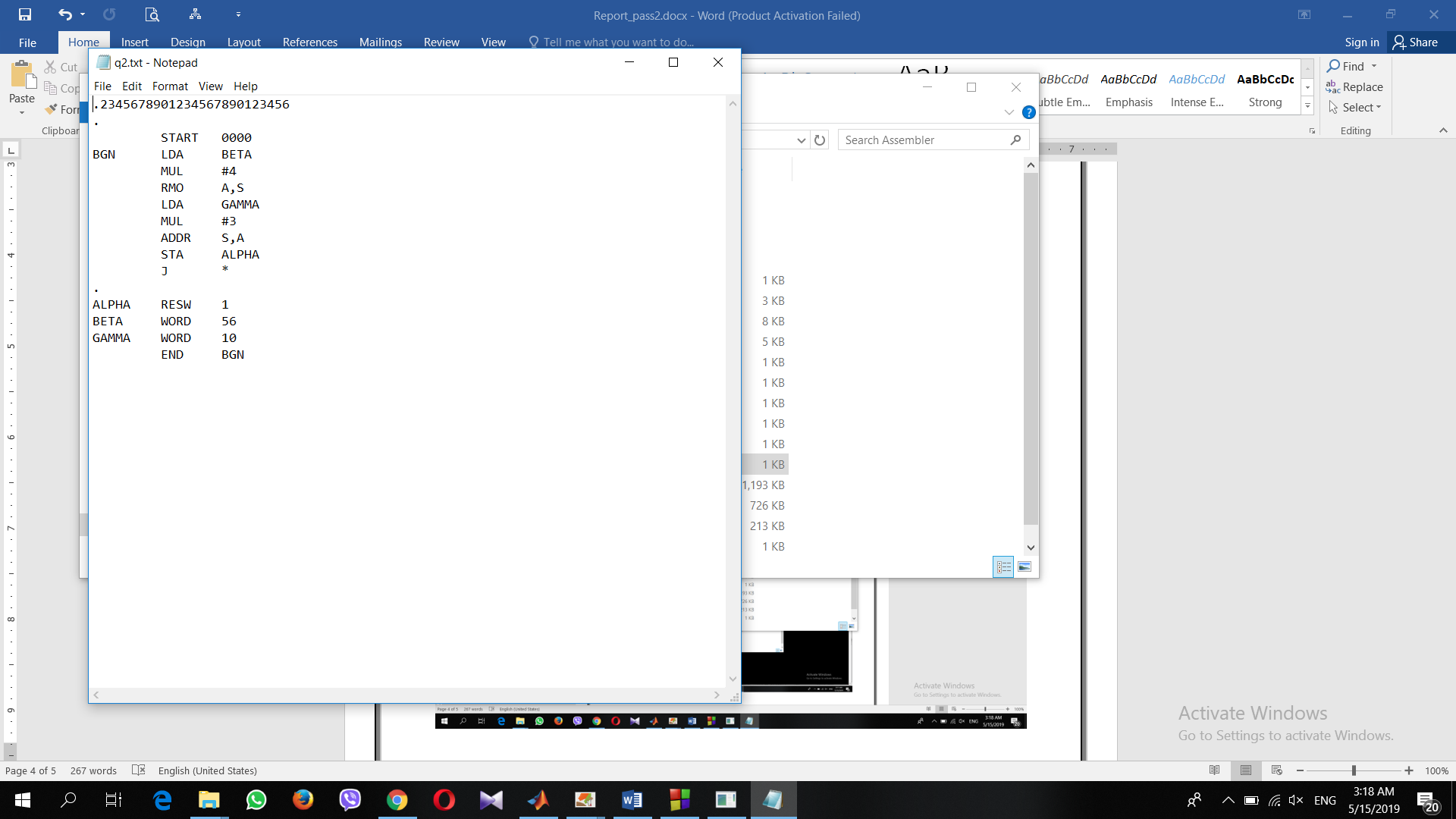
* **Assumptions:**

1. If an error was found in pass 1, assembler would not start pass 2.
2. The program code with lines ended with object code and object code file are supposed to be displayed in two different text files at the end of pass 2.
3. All output files are saved in pass relative to the running program.

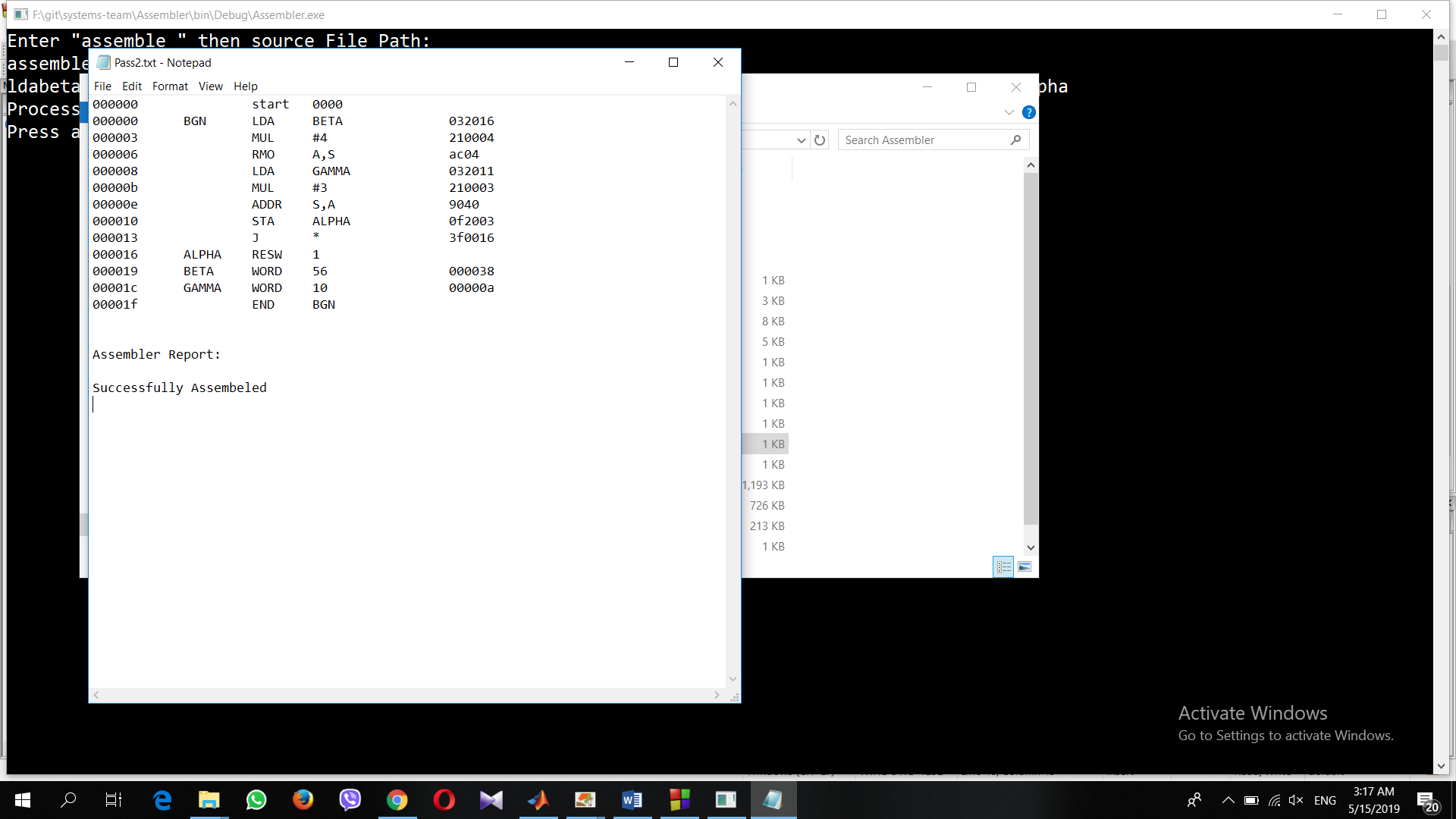
* **Sample Runs:**

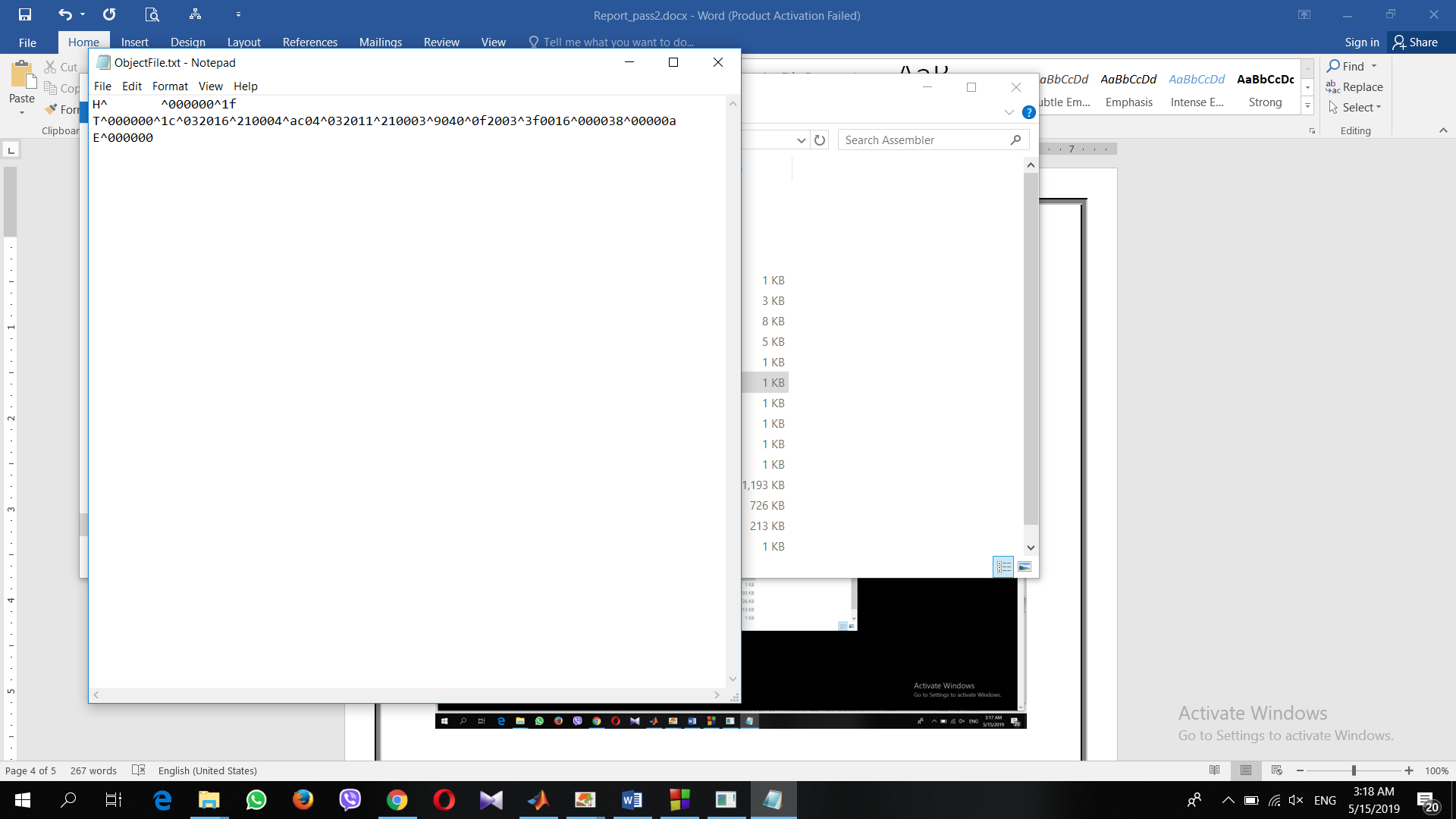
Sample 1:

* Input:



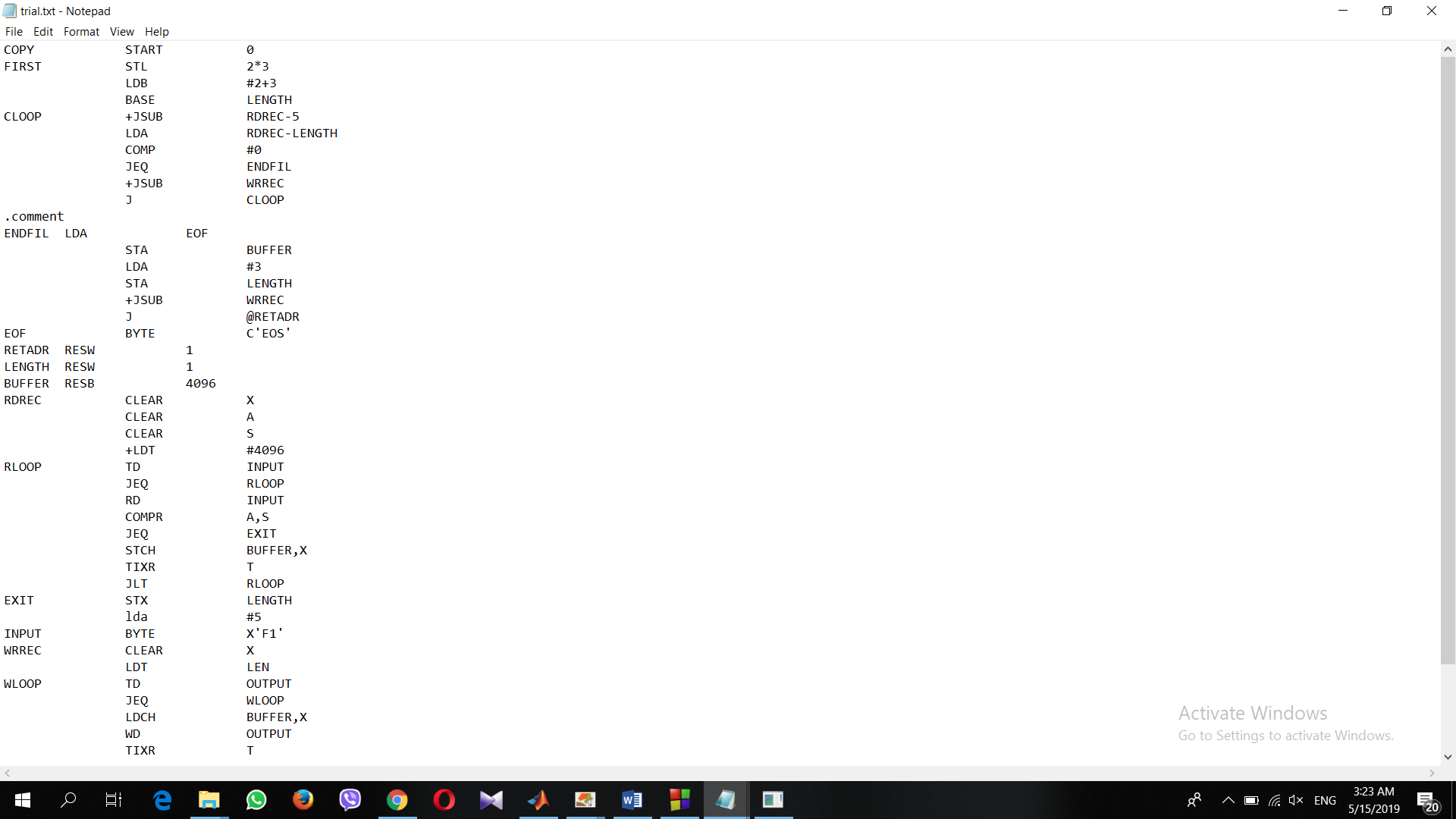
* Output:

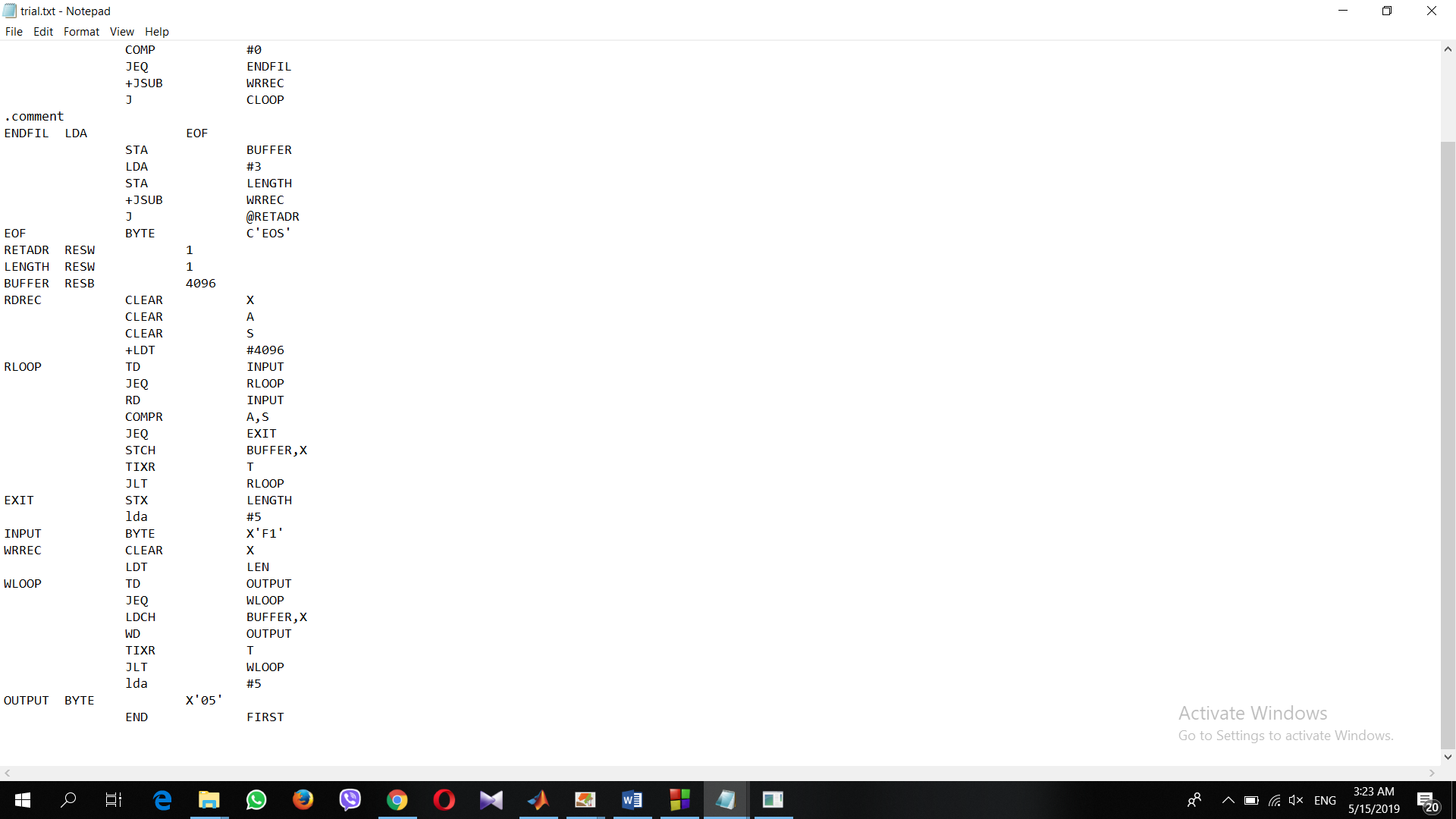




Sample 2:

* Input:





* Output:

