[cover sheet – phase 2 (one for each program evaluated)]

ASSIGNMENT 3

**CYCLOMATIC COMPLEXITY, PHASE 2**

CSE 6329 -- SOFTWARE MEASUREMENT AND QUALITY ENGINEERING

Professor Dennis J. Frailey

**Fall, 2018**

|  |  |
| --- | --- |
| NAME | STUDENT ID NUMBER |
| **Harshini Chandrasekar, Tharuna Kumar** | **1001586563, 1001537450** |

|  |  |
| --- | --- |
| **Name of Program Evaluated** | **Discussion of Any Errors Found and How to Correct Them** |
| Orange 1 | No errors found |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cyclomatic Complexity Calculation** | | | | |
| **Arcs** | **Nodes** | **C (Number of Separate Flowgraphs)** | **Arcs - Nodes** | **Arcs – Nodes + 2C**  **(Cyclomatic Complexity)** |
| 19 | 16 | 1 | 3 | 5 |

**Flow graph of Orange 1 Program:**

****

**Note:**

1. The flow chart for Orange 1 has been attached as a separate file along with the submissions.

Filename : A3 CSE6329 2018fa ORANGE1 Flowgraph Chandrasekar H Kumar T.png

1. In the above graph,

* ‘X’(A) denotes initialization in a for loop.
* ‘X’(B) denotes condition in a for loop.
* ‘X’(C) denotes incrementation in a for loop.

Where ‘X’ -> corresponding line of C code.

***Example:***

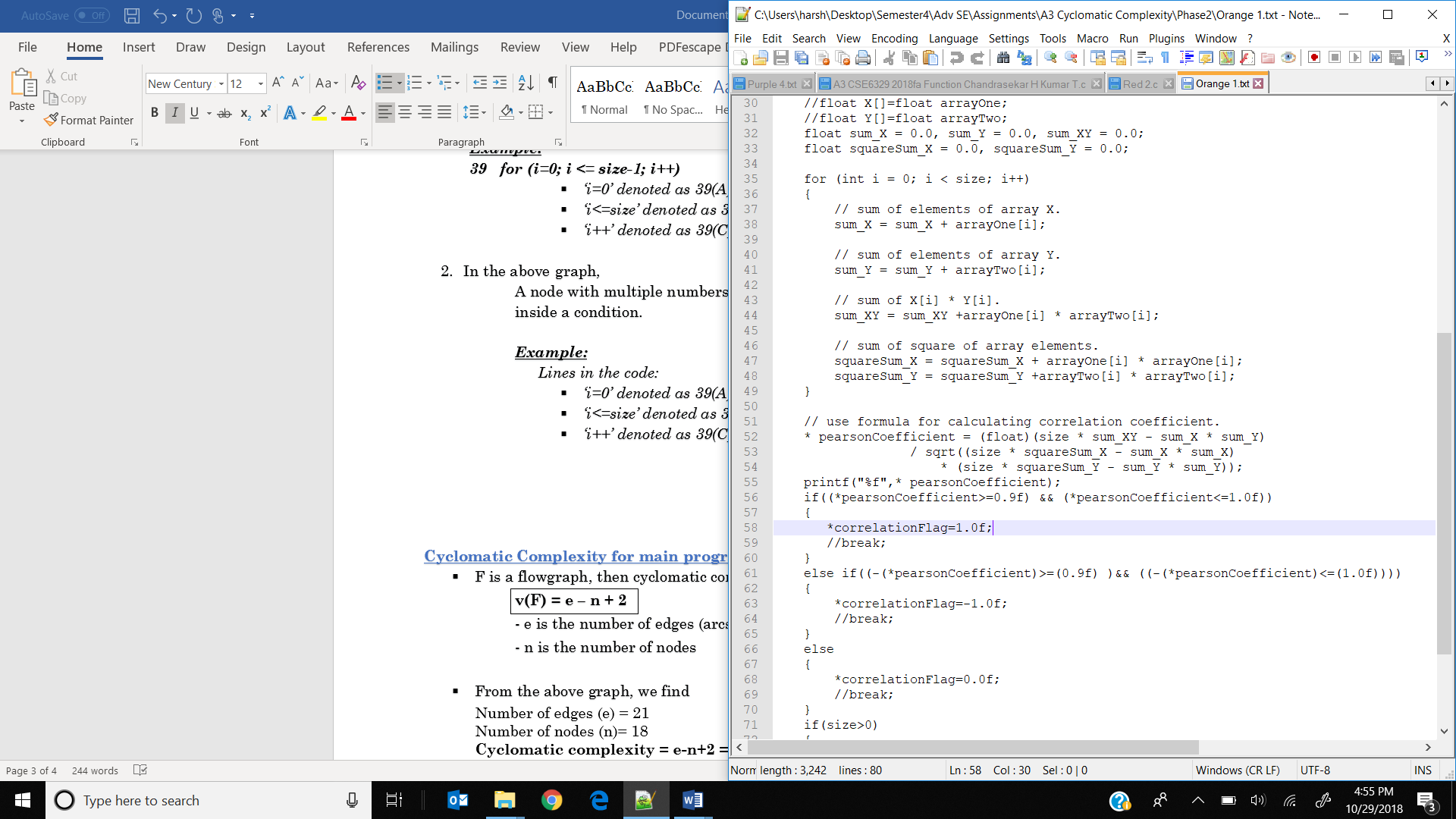
***39 for (i=0; i <= size-1; i++)***

* *‘i=0’ denoted as 39(A)*
* *‘i<=size’ denoted as 39(B)*
* *‘i++’ denoted as 39(C)*

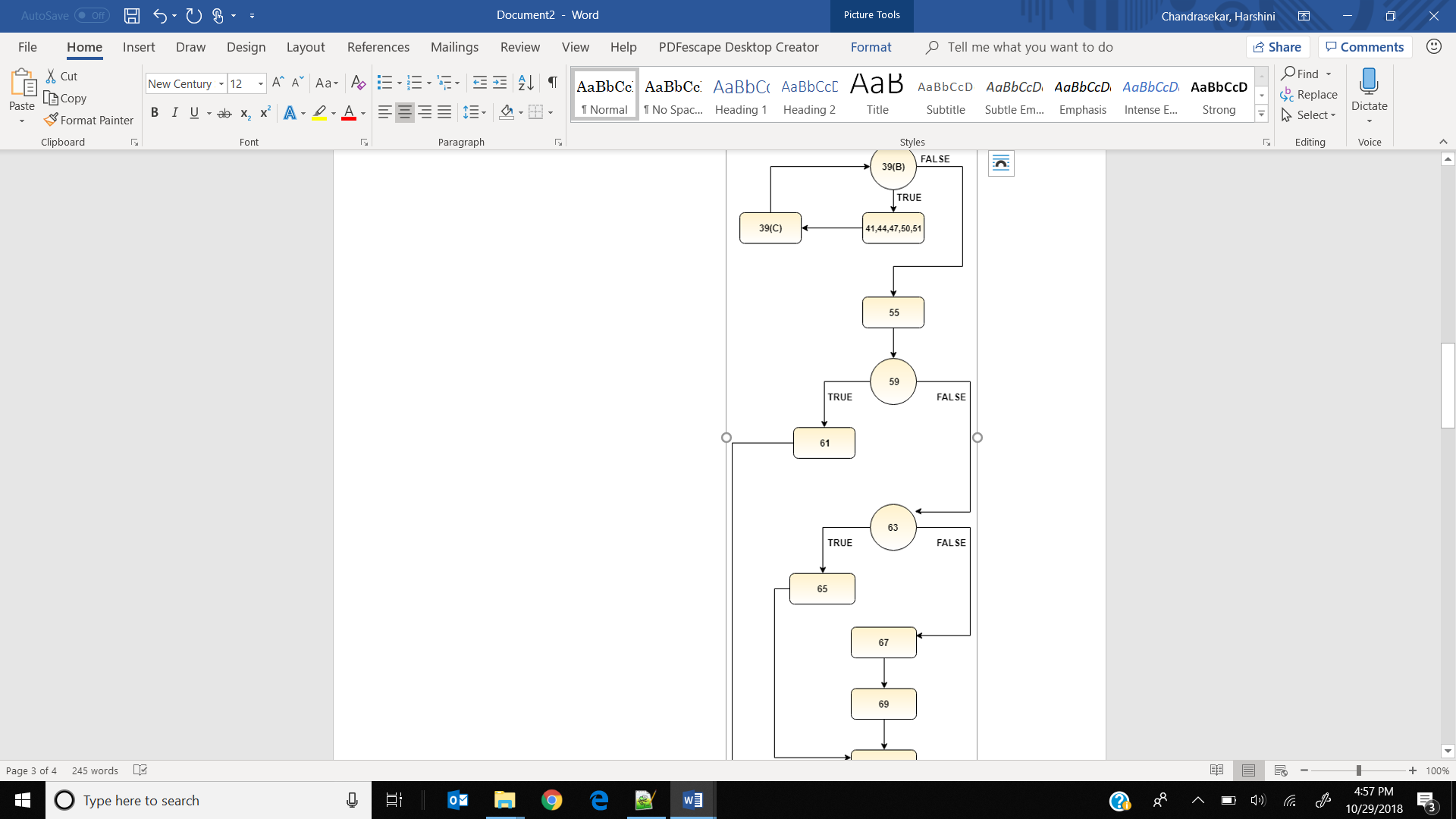
1. In the above graph,

A node with multiple numbers denote set of statements executed inside a condition.

***Example:***



Lines within the ‘for’ loop are denoted in a single node representing the line numbers separated by commas as follows:



**Cyclomatic Complexity for main program:**

* + - F is a flowgraph, then cyclomatic complexity v(F) is calculated by

**v(F) = e – n + 2**

- e is the number of edges (arcs)

- n is the number of nodes

* + - From the above graph, we find

Number of edges (e) = 19

Number of nodes (n)= 16

**Cyclomatic complexity = e-n+2 = 19 – 16 + 2 = 5**