[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** |
| Purple 4 | 1. The Correlate() function returns 1 if size is greater than zero and returns 0 if size if less than or equal to zero. The criteria of size should be added in the function. 2. The Pearson coefficient range to compute the correlationFlag is declared incorrectly. Only the lower bound (0.9) is mentioned. The condition should be:   if (\*pearsonCoefficient >= 0.9 && \*pearsonCoefficient <=1)  \*correlationFlag = 1.0; |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cyclomatic Complexity Calculation** | | | | |
| **Arcs** | **Nodes** | **C (Number of Separate Flowgraphs)** | **Arcs - Nodes** | **Arcs – Nodes + 2C**  **(Cyclomatic Complexity)** |
| 16 | 14 | 1 | 2 | 4 |

**Flow chart for Purple 4 Program:**

****

**Note:**

1. The flow chart for Purple 4 has been attached as a separate file along with the submissions.

Filename : A3 CSE6329 2018fa PURPLE4 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:***

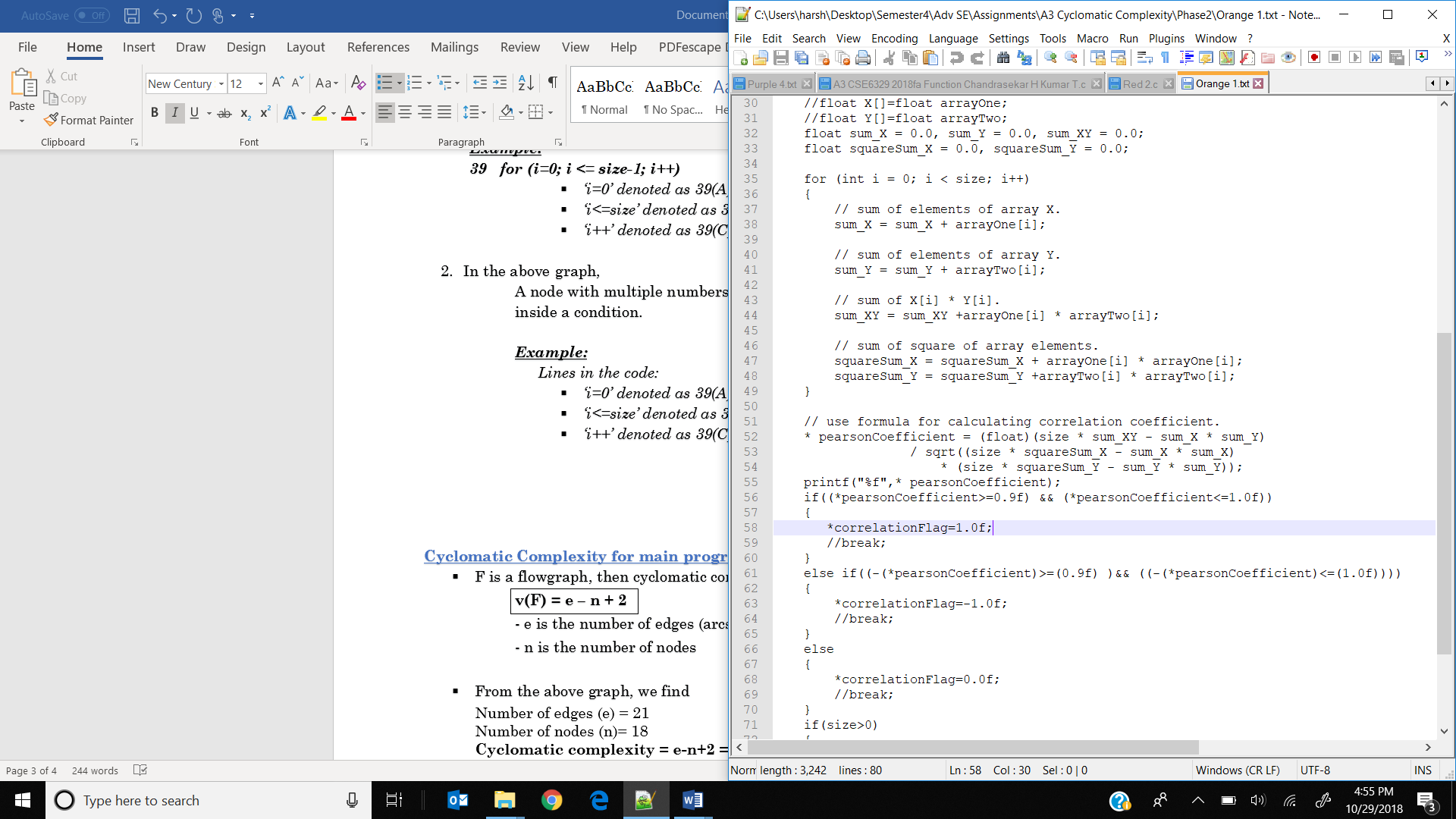
***39 for (i=0; i <= size-1; i++)***

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

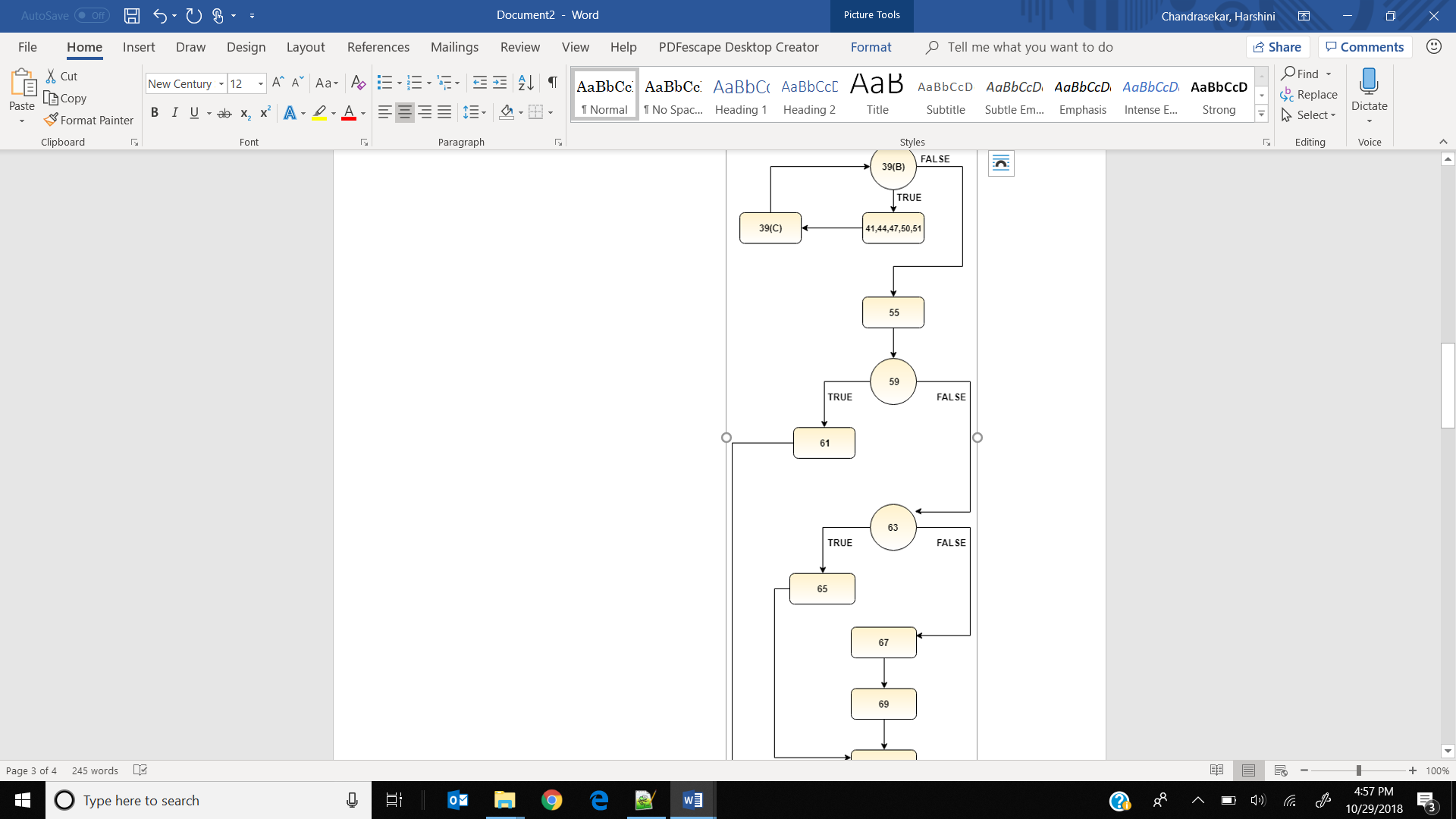
1. In the above graph,

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) = 16

Number of nodes (n)= 14

**Cyclomatic complexity = e-n+2 = 16- 14+ 2 = 4**