[cover sheet, phase 1 and full assignment]

3

ASSIGNMENT 3

**CYCLOMATIC COMPLEXITY, PHASE 1**

CSE 6329 -- SOFTWARE MEASUREMENT AND QUALITY ENGINEERING

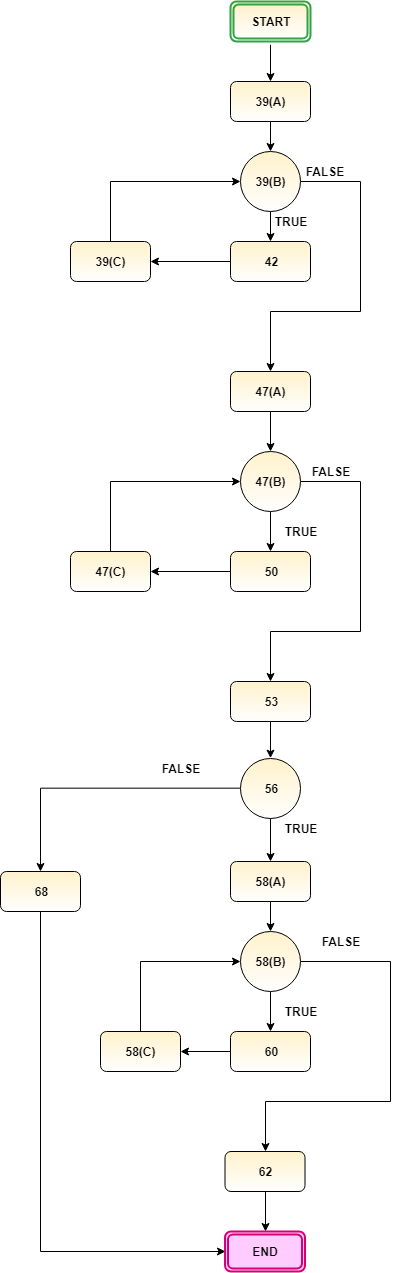
Professor Dennis J. Frailey

**Fall, 2018**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Cyclomatic Complexity Calculation** | | | | |
| **Program** | **Arcs** | | **Nodes** | **C (Number of Separate Flowgraphs)** | **Arcs - Nodes** | **Arcs – Nodes + 2C (Cyclomatic Complexity)** |
| Main | 21 | | 18 | 1 | 3 | 5 |
| Function | 24 | | 20 | 1 | 4 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grading Comments (student – do not write inside this box)** | | | | |
| **Phase 1** | | | | |
| **<total goes here>** | **/10** | **Flowgraph (main)** | | **Comments from Grader** |
| **/5** | **Complexity (main)** | |
| **/10** | **Program (function)** | |
| **/10** | **Flowgraph (function)** | |
| **/ 5** | **Complexity (function)** | |
| **Phase 2** | | | | |
| **<total goes here>** | **<name>** | | First Program Reviewed | |
| **/ 5** | | **Errors** |  |
| **/ 5** | | **Cyclomatic Complexity** |
| **/ 10** | | **Flowgraph** |
| **<total goes here>** | **<name>** | | Second Program Reviewed | |
| **/ 5** | | **Errors** |  |
| **/ 5** | | **Cyclomatic Complexity** |
| **/ 10** | | **Flowgraph** |
| **<total goes here>** | **<name>** | | Third Program Reviewed | |
| **/ 5** | | **Errors** |  |
| **/ 5** | | **Cyclomatic Complexity** |
| **/ 10** | | **Flowgraph** |
| **<total>** | **Grand Total** | | |  |

****

1. **Flow graph of Main Program:**

**Note:**

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)*

**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)

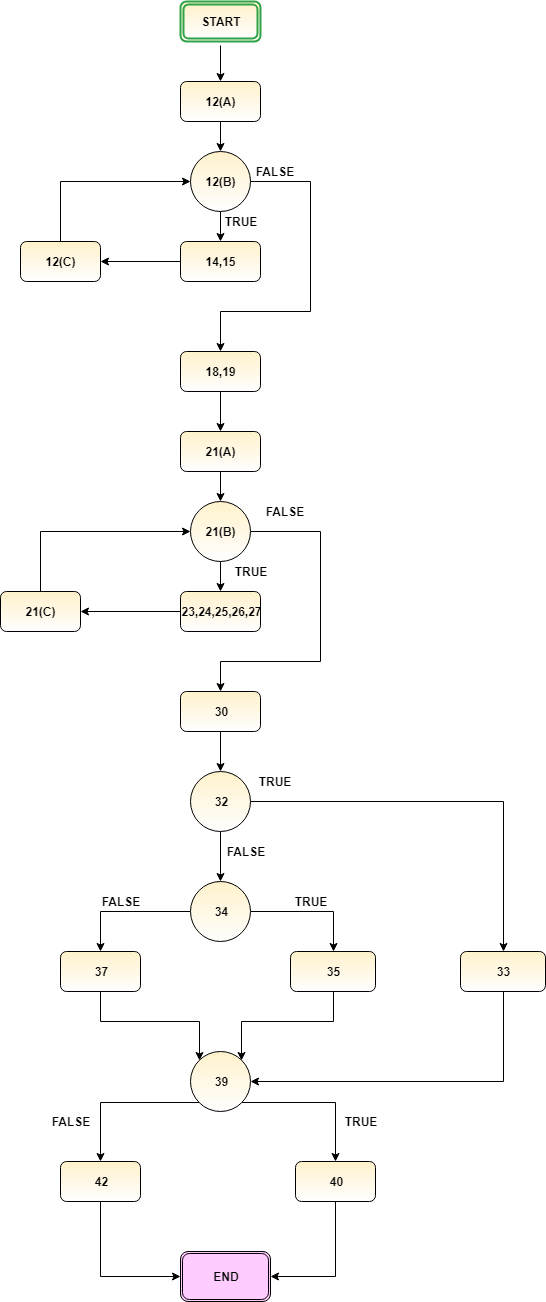
- n is the number of nodes

* + - From the above graph, we find

Number of edges (e) = 21

Number of nodes (n)= 18

**Cyclomatic complexity = e-n+2 = 21- 18+ 2 = 5**

****

1. **Flow graph of the function:**

**Note:**

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)*

**Cyclomatic Complexity of the function:**

* + - 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) = 24

Number of nodes (n)= 20

**Cyclomatic complexity = e-n+2 = 24- 20+ 2 = 6**