Cyclomatic Complexity

Cyclomatic Complexity quantifies the number of linearly independent pathways through a section of programming code. Code can be represented by a Control Flow Graph with each command represented by a node (N). These nodes are joined by edges (E) which represent the flow of the code. The code may exit at one or more points (P) or connected components. This can be defined by Mc Cabe's formula:

$M = E – N + 2P$

Task: What is the cyclomatic complexity of the following piece of code?

```
public static string IntroducePerson(string name, int age)
{
    var response = $"Hi! My name is {name} and I'm {age} years old.";
    if (age >= 18)
        response += " I'm an adult.";
    if (name.Length > 7)
        response += " I have a long name.";
    return response;
}
```

Step 1: Nodes

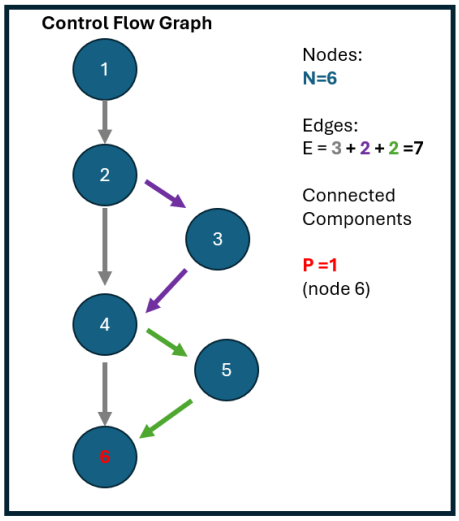
The code contains 6 nodes. They have been marked up in the code below:

Code

```
public static string IntroducePerson(string name, int age)
{
#Node 1
var response = $"Hi! My name is {name} and I'm {age}
years old.";
  #Node 2
  if (age >= 18)
    #Node 3
    response += " I'm an adult.";
  #Node 4
  if (name.Length > 7)
    #Node 5
    response += " I have a long name.";
  #Node 6
  return response;
}
```

Step 2: Control Flow Graph

The Control Flow Graph is shown below. There are 6 nodes (N=6). Nodes 2 and 4 represent the if statements which branch off and then return to the main stem of the code. Node 6 is the return response. There is only one return from the code, so P =1.



There are 7 edges (E=7) in the control flow diagram. They represent 3 independent pathways. The 3 grey edges show the flow where the both if statements are false (ie less than 18 and a short name). The purple pathway shows an independent flow which

is triggered by the if statement at node 2 (ie over 18). The 3rd independent pathway, shown in green is triggered by the second if statement at node 4 (ie a long name).

Step 3: McCabe's Formula

The cyclomatic complexity can be determined using McCabe's Formula:

M = E - N + 2P

= 7 – 6 + 2 x 1

= **3**

The code has a cyclomatic complexity of 3.

While it is possible to determine the cyclomatic complexity of the code using McCabe's Formula. It is also clear from the control flow graph that there are 3 linearly independent pathways (grey, purple and green).