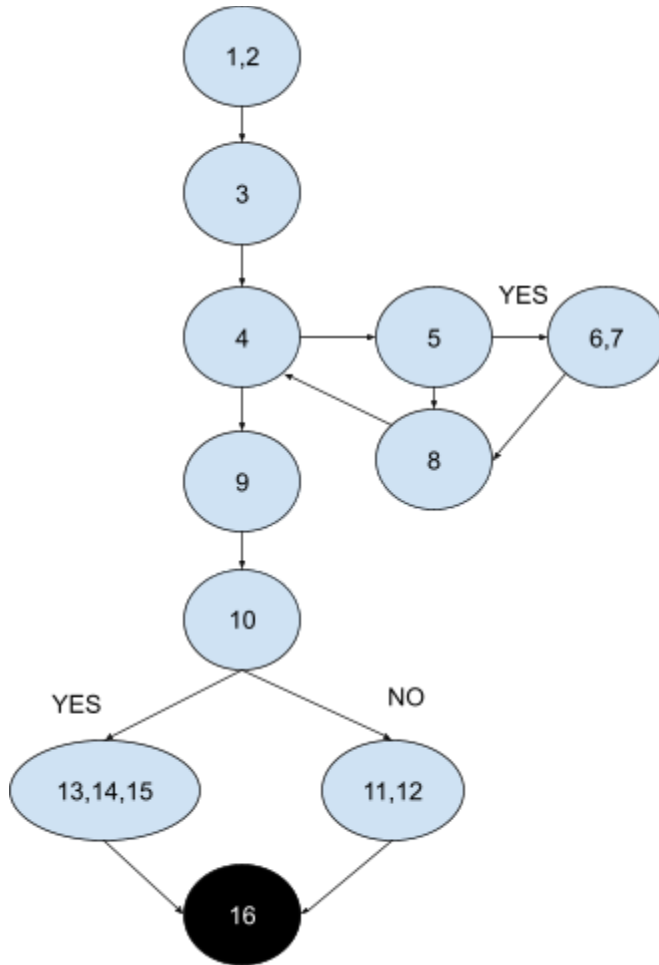


<b>Subject :</b>	MCAL35 Software Testing and Quality Assurance Lab
<b>Assignment no:</b>	2
<b>Assignment Date:</b>	08-09-2022
<b>Submission deadline:</b>	16-09-2022

Q.1 Draw CFG, find Cyclomatic complexity

```
#include <iostream>
1 void prime(int n)
2 {
3     int i, flag = 0;
4     for (i = 2; i <= n/2; ++i) {
5         if (n%i == 0) {
6             flag = 1;
7             break;
8         }
9     }
10    if (flag == 1) {
11        cout << n << " is not a prime number.";
12    }
13    else {
14        cout << n << " is a prime number.";
15    }
16 }
```

**OUTPUT:**



Cyclomatic Complexity =  $E - N + 2$   
 where, E = Number of Edges  
 N = Number of Nodes

$$\text{Cyclomatic Complexity} = 13 - 11 + 2 = 4$$

Q.2 Draw CFG, find Cyclomatic complexity

```

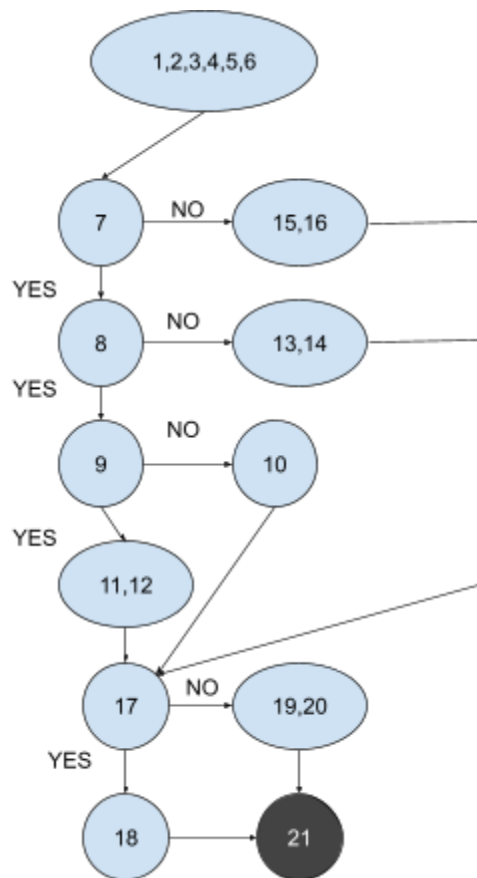
public class Demo {
    public static void main(String[] args) {
        1    int year;
        2    Scanner scan = new Scanner(System.in);
        3    System.out.println("Enter any Year:");
        4    year = scan.nextInt();
        5    scan.close();
        6    boolean isLeap = false;
        7    if(year % 4 == 0) {
  
```

```

8      if( year % 100 == 0)
9          if ( year % 400 == 0)
10             isLeap = true;
11         else
12             isLeap = false; }
13     Else
14         isLeap = true; }
15     else {
16         isLeap = false; }
17     if(isLeap==true)
18         System.out.println(year + " is a Leap Year.");
19     Else
20         System.out.println(year + " is not a Leap Year.");
21     }
    }

```

## OUTPUT:



Cyclomatic Complexity =  $E - N + 2$   
 where, E = Number of Edges

N = Number of Nodes

$$\text{Cyclomatic Complexity} = 15 - 12 + 2 \\ = 5$$

Q.3 For the following code, find statement coverage, decision coverage, branch coverage.

```
main()
{
1   int work;
2   int payment=0;
3   cin>> work;
4   if(work > 0){
5   payment=40;
6   if(work > 20)
7   {
8   if (work <=30)
9   payment = payment + (work -25) * 0.5;
10  else
11  {
12      payment = payment + 50 + (work -30) * 0.1;
13      if (payment >= 3000)
14          payment = payment * 0.9;
15  }
16  }
17  }
18  cout<< "Final payment"<<payment;
19  }
```

### 1. Statement Coverage

Test Case id	Input	Statement Executed	Statement Coverage
1	25	1-9,16-19	$(13/19)*100=68.42\%$
2	50	1-8,10-13,15-19	$(17/19)*100=89.47\%$
3	30000	1-8,10-19	$(18/19)*100=94.73\%$

### 2. Branch Coverage/Decision Coverage

Test case Id	Condition Evaluated	Input :Work	Payment	Branch Coverage
1	if(work>0) [invalid]	0	0	4,17
2	if(work>0) [valid] if(work>20) [invalid]	18	40	4-6,17
3	if(work>0) [valid] if(work>20) [valid] if(work<=30) [valid]	22	38.5	4-9,16,17
4	if(work>0) [valid] if(work>20) [valid] if(work<=30) [invalid] ->else if (payment >= 3000) [invalid]	60	93	4-8,10-13,15-17
5	if(work>0) [valid] if(work>20) [valid] if(work<=30) [invalid]->else if (payment >= 3000) [valid]	30000	3087	4-8,10-17