Consider the following source code and perform the following tasks:

a) Calculate cyclomatic complexity for the below program using all the three methods discussed in the class. [4]
b) Draw the control flow graph, identify all the independent paths. [3]
c) Generate the testing cases using equivalence partitioning for at-least one value from valid and invalid classes, show your test cases in grid form. [3]

```
1. begin int x, y, power;
2. float z;
3. input(x, y);
4. if(y<0)
5. power = -y;
6. else power = y;
7. z=1;
8. while(power!=0)
9. { z=z*x;
10.     power=power-1;
11.     } if(y<0)
12.     z=1/z;
13.     output(z);
14.     end
```

Solution:

Part a)

Method-01:

CC = Total number of closed regions in the control flow graph + 1
= 3 + 1
= 4

Method-02:

Cyclomatic Complexity = E – N + 2P
= 16 – 14 + 2(1)
= 4

Method-03:

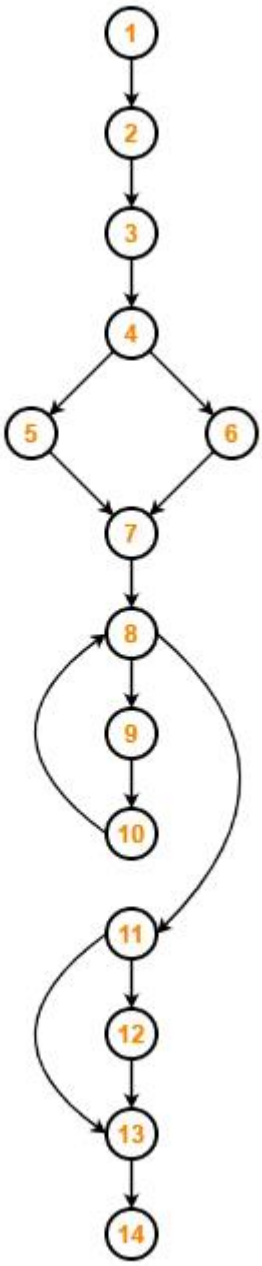
Cyclomatic Complexity = P + 1

= 3 + 1

= 4

Part b)

i) Control Flow Graph:



ii) Independent Paths:

Path 1: input(x, y) → if(y<0) → power = -y → z=1 → while(power!=0) → z=z*x → power=power-1 → if(y<0) → z=1/z → output(z)

Path 2: input(x, y) → if(y<0) → power = -y → z=1 → while(power!=0) → z=z*x → power=power-1 → output(z)

Path 3: input(x, y) → if(y<0) → power = -y → z=1 → output(z)

Path 4: input(x, y) → if(y>=0) → power = y → z=1 → while(power!=0) → z=z*x → power=power-1 → output(z)

Path 5: input(x, y) → if(y>=0) → power = y → z=1 → output(z)

Part3:

Test Case	Input Values	Expected Output
1	x = 5	125
	y = 3	
2	x = 2	16
	y = 4	
3	x = "a"	Invalid Input
	y = "b"	
4	x = NULL	Invalid Input
	y = 5	