Software Testing and Quality Assurance

Lab Assignment - 6

PRN: 15070121170

NAME: Yash Saboo

BRANCH: CS (C4)

Question

Write a program that accepts marks of 20 students of 5 subjects and outputs the average and pass/fail message.
Compute the following:
a. Draw flow graph
b. Draw DD graph and connection matrix
c. Compute cyclomatic complexity
d. Identify independent paths

Code

```
1    import csv
2    marks = read_csv("/marks.csv")
3    res = []
4    resDummy = []
5    avg = []
6    avgDummy = 0
7    for i in range(20):
8      for j in range(5):
9        avgDummy = avgDummy + marks[i][j]
10       if marks[i][j] < 40:
11         resDummy.append("F")
12       else:    resDummy.append("P")
13     #For Loop End
14     avg.append(avgDummy/5)
15     res.append(resDummy)
16     resDummy = []
17     avgDummy = 0
18   end()
```

Other Information

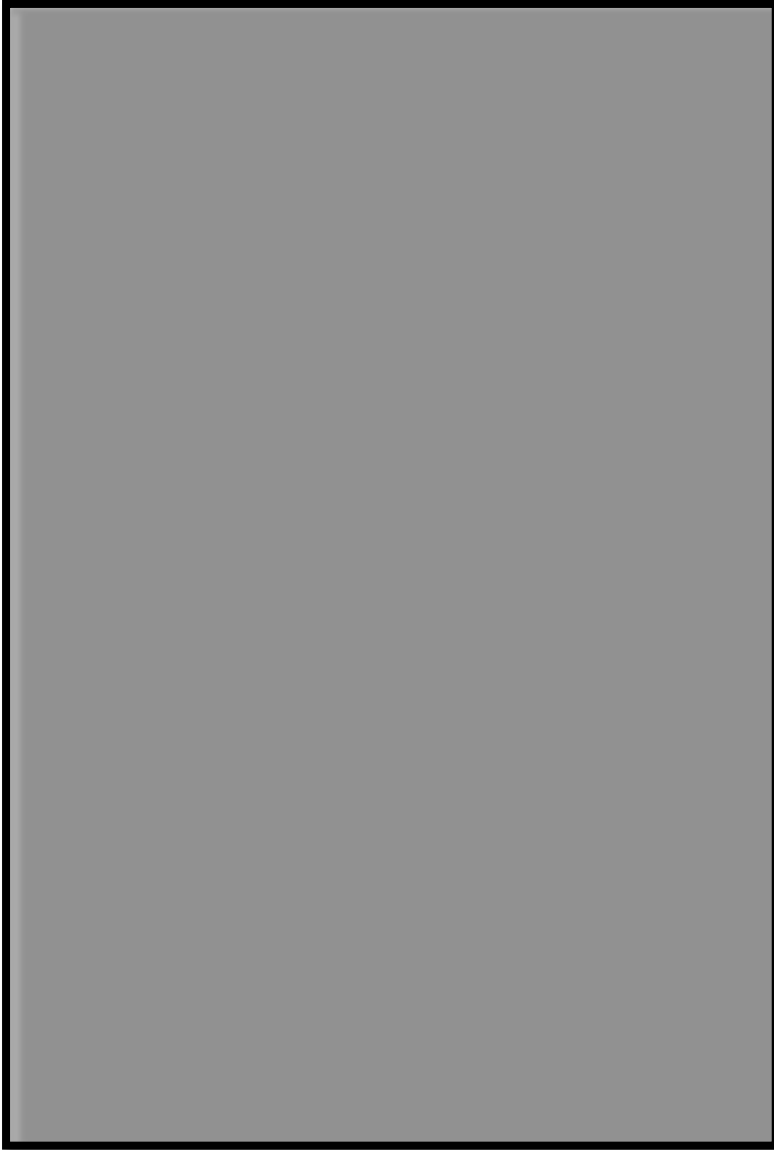|	
Marks	[[8, 59, 24, 57, 41], [79, 62, 14, 66, 40], [48, 49, 5, 35, 33], [97, 19, 52, 71, 3], [92, 20, 58, 21, 46], [84, 56, 84, 25, 90], [55, 90, 15, 59, 17], [52, 22, 86, 36, 9], [62, 52, 5, 16, 58], [95, 55, 1, 9, 99], [11, 26, 53, 62, 34], [8, 43, 8, 47, 59], [51, 61, 46, 17, 50], [60, 1, 25, 85, 36], [28, 89, 99, 80, 88], [66, 90, 37, 84, 41], [86, 13, 87, 64, 32], [15, 16, 69, 83, 59], [44, 37, 55, 25, 15], [69, 57, 13, 78, 41]]
Avg. Marks	[37.8, 52.2, 34.0, 48.4, 47.4, 67.8, 47.2, 41.0, 38.6, 51.8, 37.2, 33.0, 45.0, 41.4, 76.8, 63.6, 56.4, 48.4, 35.2, 51.6]
Result:Pass/Fail	[['F', 'P', 'F', 'P', 'P'], ['P', 'P', 'F', 'P', 'P'], ['P', 'P', 'F', 'F', 'F'], ['P', 'F', 'P', 'P', 'F'], ['P', 'F', 'P', 'F', 'P'], ['P', 'P', 'P', 'F', 'P'], ['P', 'P', 'F', 'P', 'F'], ['P', 'F', 'P', 'F', 'F'], ['P', 'P', 'F', 'F', 'P'], ['P', 'P', 'F', 'F', 'P'], ['F', 'F', 'P', 'P', 'F'], ['F', 'P', 'F', 'P', 'P'], ['P', 'P', 'P', 'F', 'P'], ['P', 'F', 'F', 'P', 'F'], ['F', 'P', 'P', 'P', 'P'], ['P', 'P', 'F', 'P', 'P'], ['P', 'F', 'P', 'P', 'F'], ['F', 'F', 'P', 'P', 'P'], ['P', 'F', 'P', 'F', 'F'], ['P', 'P', 'F', 'P', 'P']]

Screenshot


```python
1  import csv
2  marks = read_csv("/marks.csv")
3  res = []
4  resDummy = []
5  avg = []
6  avgDummy = 0
7  for i in range(20):
8      for j in range(5):
9          avgDummy = avgDummy + marks[i][j]
10         if marks[i][j] < 40:
11             resDummy.append("F")
12         else:    resDummy.append("P")
13     #For Loop End
14     avg.append(avgDummy/5)
15     res.append(resDummy)
16     resDummy = []
17     avgDummy = 0
18 end()
```
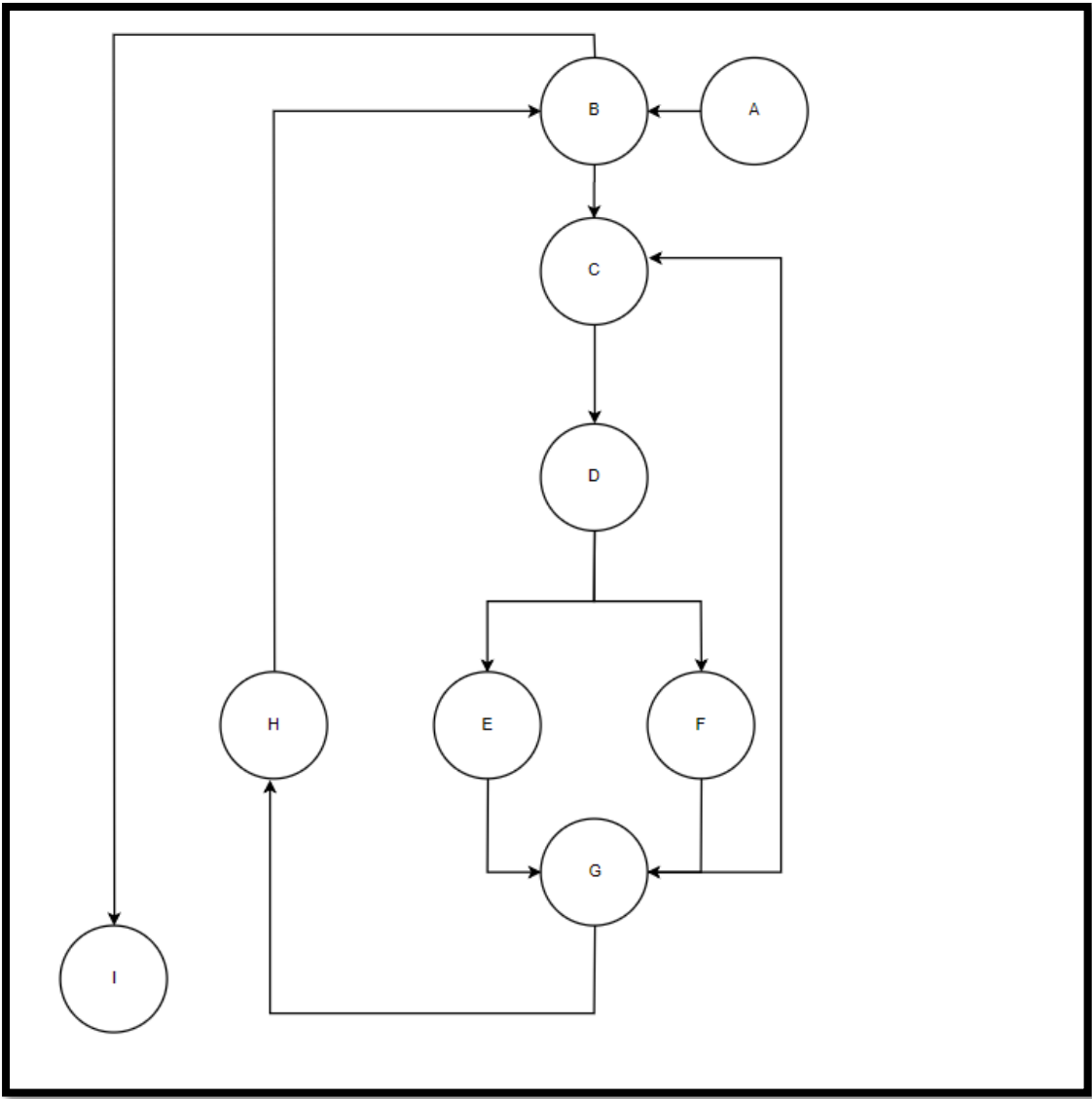
```python
1  res
```

```
[['F', 'P', 'F', 'P', 'P'],
 ['P', 'P', 'F', 'P', 'P'],
 ['P', 'P', 'F', 'F', 'F'],
 ['P', 'F', 'P', 'P', 'F'],
 ['P', 'F', 'P', 'F', 'P'],
 ['P', 'P', 'P', 'F', 'P'],
 ['P', 'P', 'F', 'P', 'F'],
 ['P', 'F', 'P', 'F', 'F'],
 ['P', 'P', 'F', 'F', 'P'],
 ['P', 'P', 'F', 'F', 'P'],
 ['F', 'F', 'P', 'P', 'F'],
 ['F', 'P', 'F', 'P', 'P'],
 ['P', 'P', 'P', 'F', 'P'],
 ['P', 'F', 'F', 'P', 'F'],
 ['F', 'P', 'P', 'P', 'P'],
 ['P', 'F', 'F', 'P', 'P'],
 ['P', 'F', 'P', 'P', 'F'],
 ['F', 'F', 'P', 'P', 'P'],
 ['P', 'F', 'P', 'F', 'F']]
```

Flow Graph

Decision Directed Graph

Connection Matrix

	A	B	C	D	E	F	G	H	I
A		1							
B			1						1
C				1					
D					1	1			
E							1		
F		1					1		
G			1					1	
H		1							
I									

Cyclomatic Complexity

The complexity is Number of Simple decisions + 1 = 3+1 = 4

Independent Paths

1. ABI
2. ABCDEGHBI
3. ABCDFGHBI