Perspective on Software Testing : Venn Diagram Explanation

Specification (S – Specified Behavior) : Write a program that read numbers from the keyboard and add positive numbers and negative numbers separately. Also, count total inputs. Stop the program when user enter -0.

Program (P – Programmed Behavior):

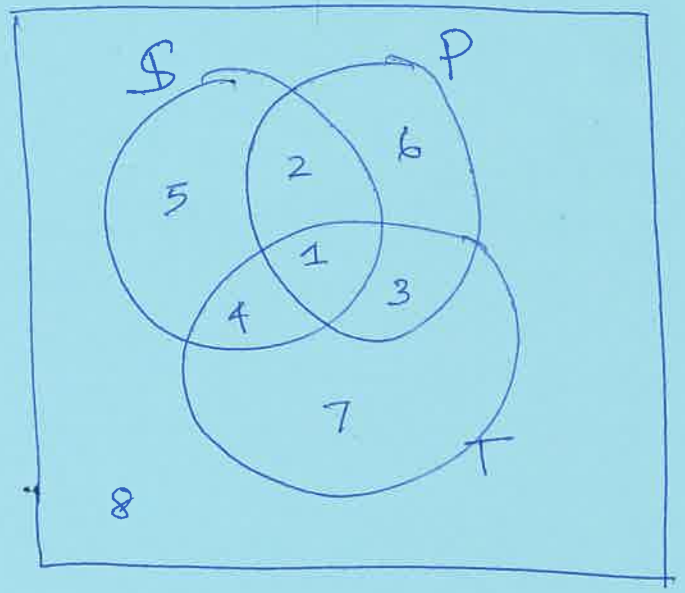
```
int pSum = 0, nSum = 0, num, totalNumbers = 0;
cin >> num;
while (num != -0)
{

       if (num < 0)
       {
              nSum = nSum + num;
              totalNumbers++;
       }
       else if (num > 0)
       {
              pSum = pSum + num;
              totalNumbers++;
       } else
       {
          cout<<"zero is neither positive nor negative"
       }


       cin >> num;
}

cout << pSum << endl;
cout << nSum << endl;
cout << totalNumbers << endl;
```

Test Cases (T – Tested Behavior):

Test Case #	Test Data	Expected Output	Actual Output
T1	2 56 -5 8 -7 -0	66 -12 5	
T2	2 6 -6 0 -4 -0	8 10 4	
T3	-0 2 6 -1	0 0 0	
T4	2 6 -1 -0 5 -5	8 -1 3	
T5	8 -2.5 -3 7.5 -0	8 -13.0 4	
T6	8 -2 “hello” -3 -0	Not specified, but should not crash the program	
-----	----------	-------------	-----------

Venn diagram:

Discuss during the lecture



Venn diagram example Test cases:

1: T_1, T_3, T_4

4: T 5

3: T_2

7: T_6

Region	Significance
1: S, P, T	Need to maximize.
2: S, P, ~T	unknown bugs possible
3: ~S, P, T	extra functionality, Tested good
4: S, ~P, T	known missing functionality
5: S, ~P, ~T	most probably unknown missing functionality
6: ~S, P, ~T	extra functinaluy, potential bugs as not tested
7: ~S, ~P, T	Tested unspecified behaviors that are expected or related to qualuy factors => good
8: ~S, ~P, ~T	all other possible behaviors. can ~~be~~ have bugs(defects)