

# Software Testing and Reliability

Xiaoyuan Xie 谢晓园

[xxie@whu.edu.cn](mailto:xxie@whu.edu.cn)

计算机学院E301

---

# Lecture 6

---



**Lecture on 2019.3.12 (Wed)**

**18:30- 20:05     4-601**

**Remains unchanged!!!**

**Lecture on 2019.3.13 (Wed)**

**14:04 – 15:40    4-601**



**Lecture on 2019.3.21 (Thu)**

**18:30- 20:05    4-401**



# **White-box Testing (control-flow coverage)**

## Control-flow Coverage (continued)

- Path coverage
  - Execute every path at least once
  - **Imply all the coverage criteria**
  - Problem
    - When the software contains loops, it has infinite number of paths – then 100% path coverage becomes infeasible.
  - Solution
    - Group paths into equivalence classes (see the next slide)



## Equivalence classes of paths

- Two paths are considered equivalent if they differ only in the number of iterations
- Two classes
  - **Zero** loop traversal
  - **At least one** loop traversals

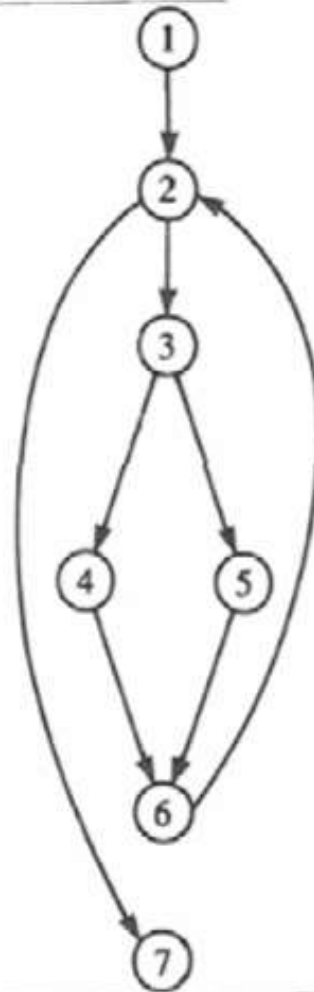
# Coverage for equivalence classes of paths (continued)

while loop example

3 paths for the above while loop example

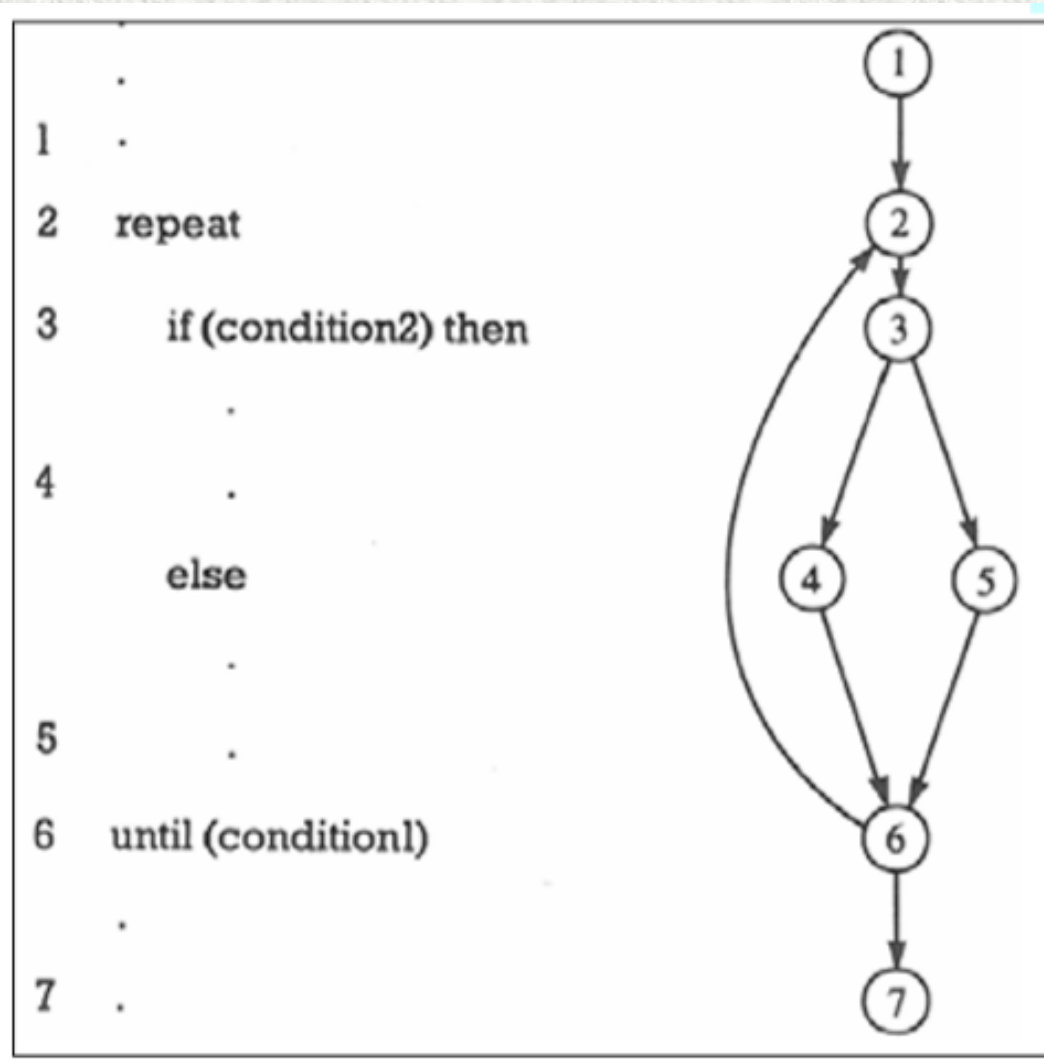
1. 1-2-7
2. 1-2-3-4-6-2-7
3. 1-2-3-5-6-2-7

```
1 .  
2 while (condition1) do  
  begin  
3   if (condition2) then  
4     .  
  else  
5     .  
6 end  
7 .
```





# Coverage for equivalence classes of paths (continued)



repeat loop example

6 paths for the above repeat loop example

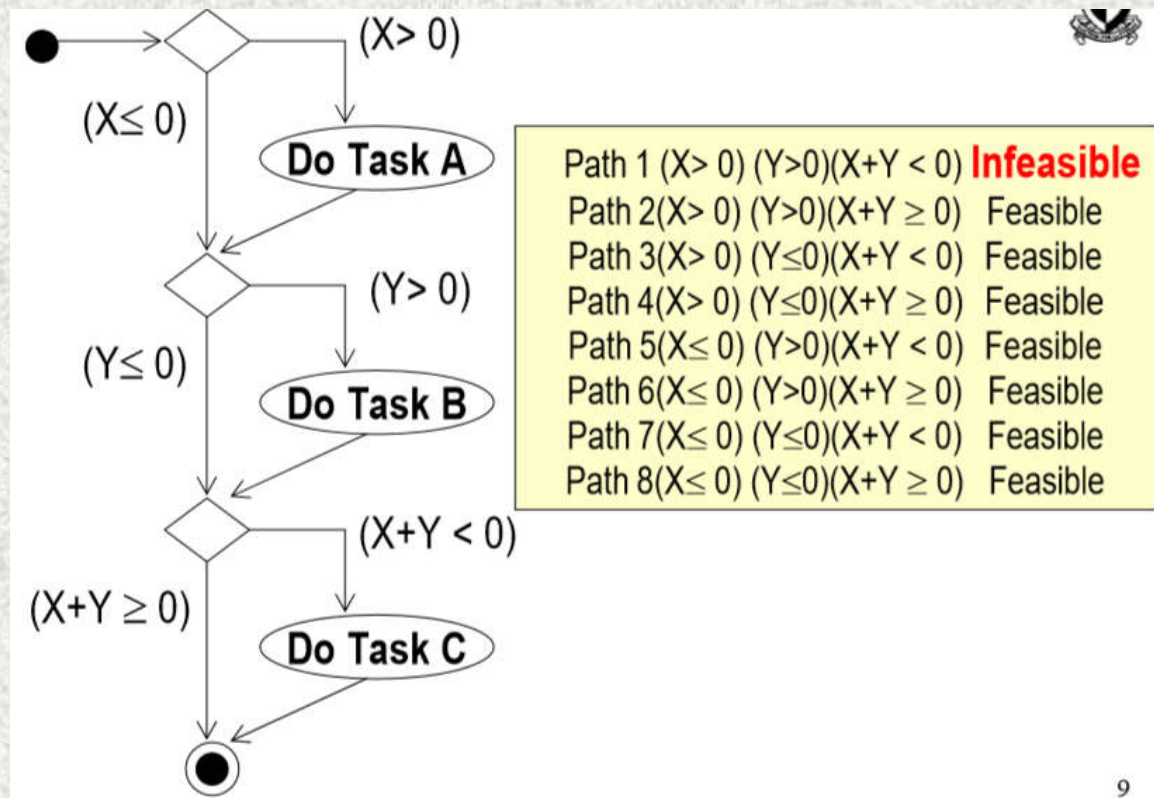
1. 1-2-3-4-6-7
2. 1-2-3-5-6-7
3. 1-2-3-4-6-2-3-4-6-7
4. 1-2-3-4-6-2-3-5-6-7
5. 1-2-3-5-6-2-3-4-6-7
6. 1-2-3-5-6-2-3-5-6-7

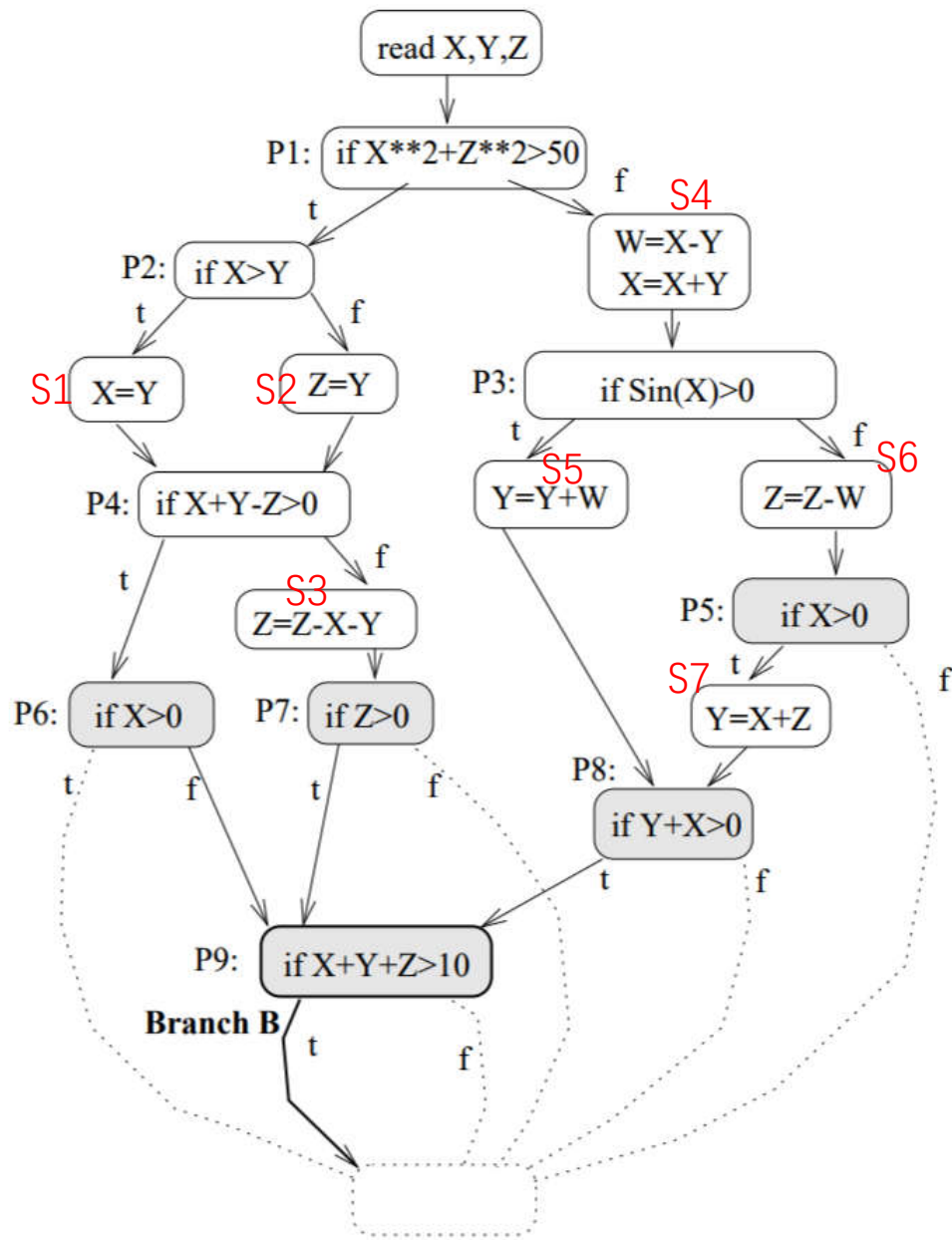
# How to generate a test case that executes an item?

- It is a search problem
  - Normally, there are more than one paths that will reach the item
- The task is to solve ONE of the path conditions related to the item
  - Random approach
  - Search algorithms, such as the genetic algorithm
  - Constraint solver



# Or, just solve the path conditions manually





Statement coverage?

Branch coverage?

Path coverage?

If each branch has statements, then branch coverage has no difference from statement coverage



# How do we know that an item has been executed?

- By program instrumentation
  - Insert some printing statements to trace the program execution

Can be done either after or in parallel with compilation

# Applications

- To measure the thoroughness of a test
- To control the behavior of a test
- To generate symbolic traces
- To be used in dynamic data flow analysis



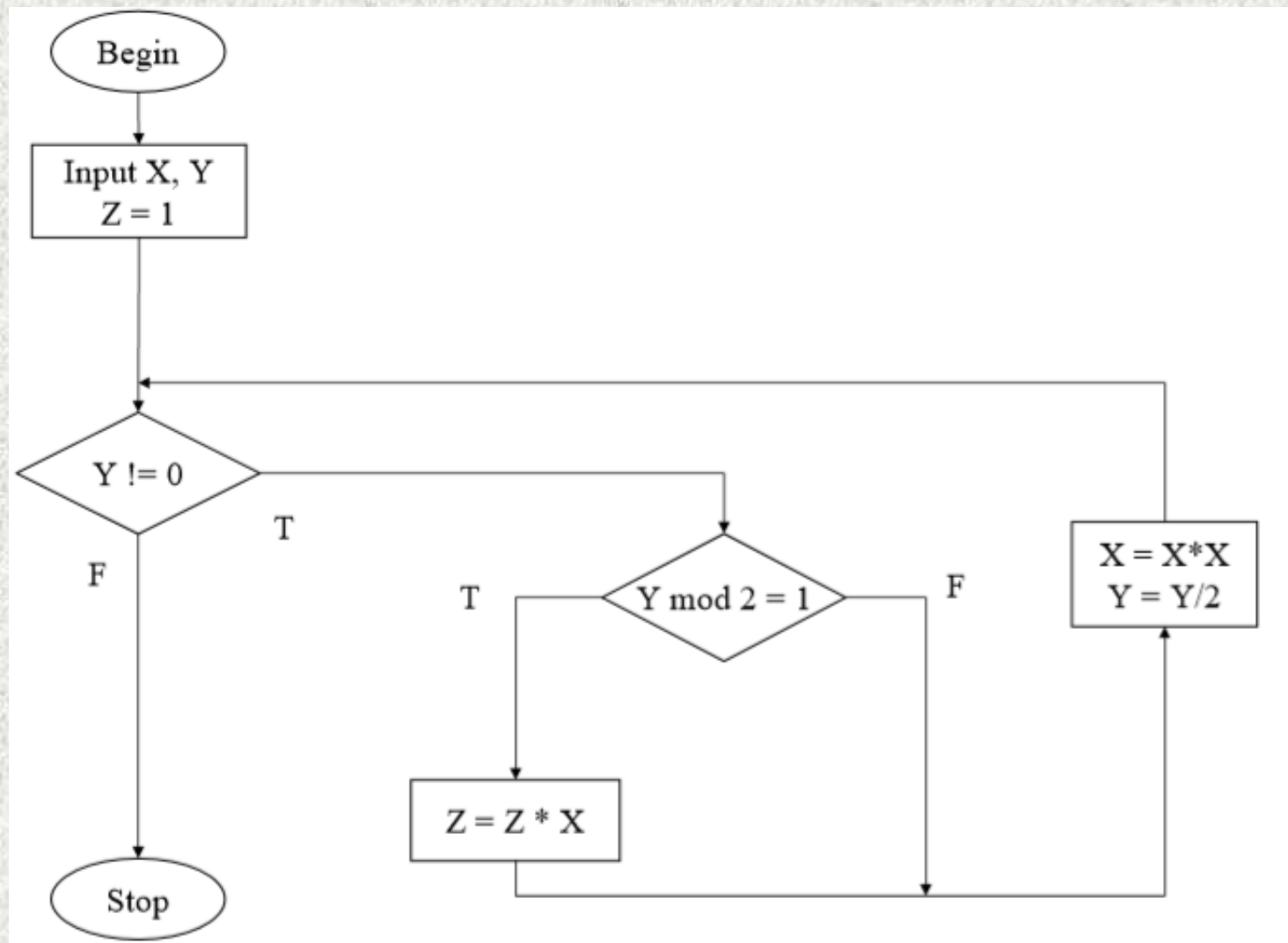
## Example: Coverage Testing

- Intuition?
- Coverage criteria
  - Control-flow coverage
  - Data-flow coverage --- talk later

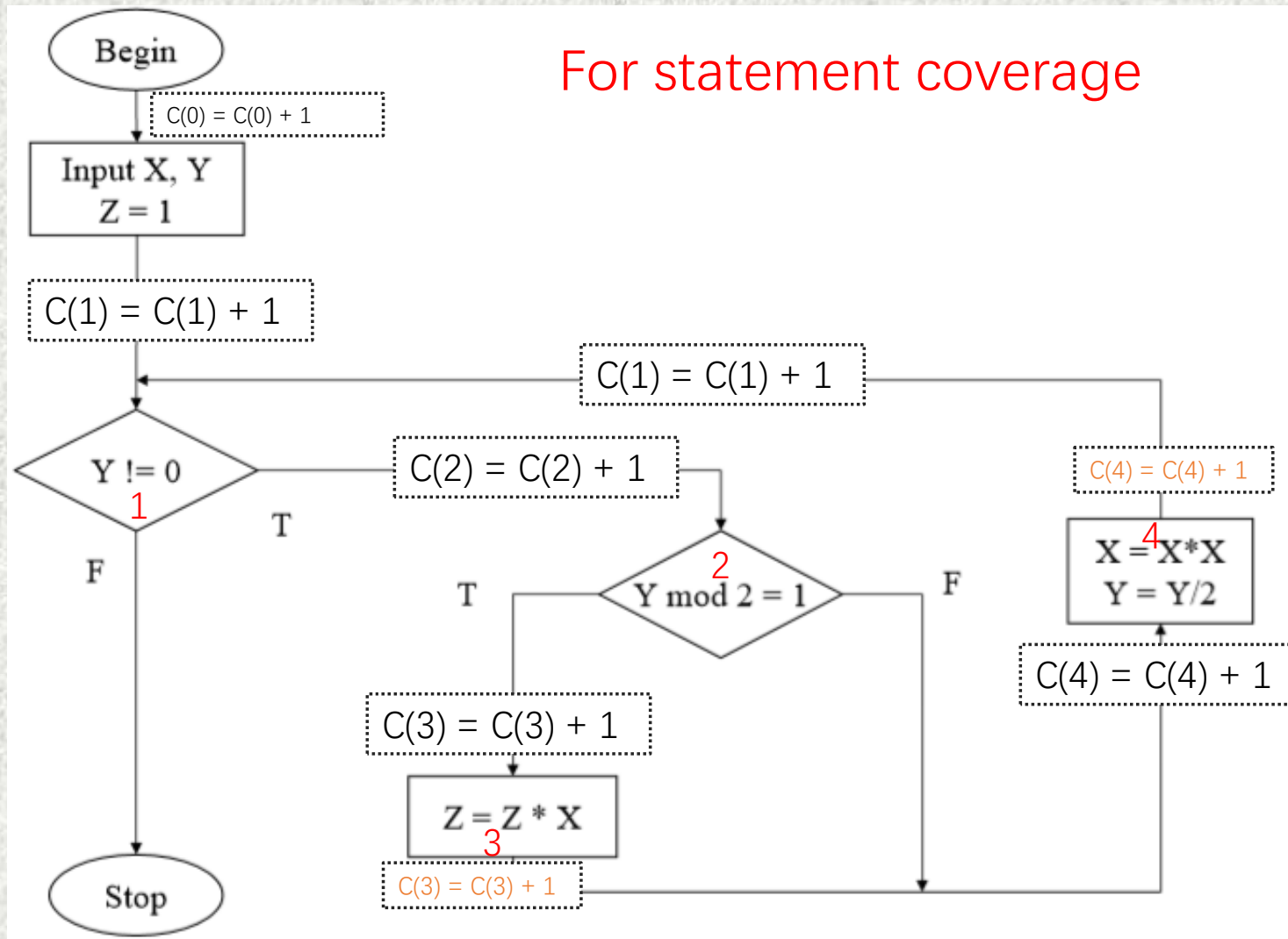
# Control-flow Coverage

- Every statement be executed at least once.
- Every branch be executed at least once.
- .....
- ....
- Every path be executed at least once.



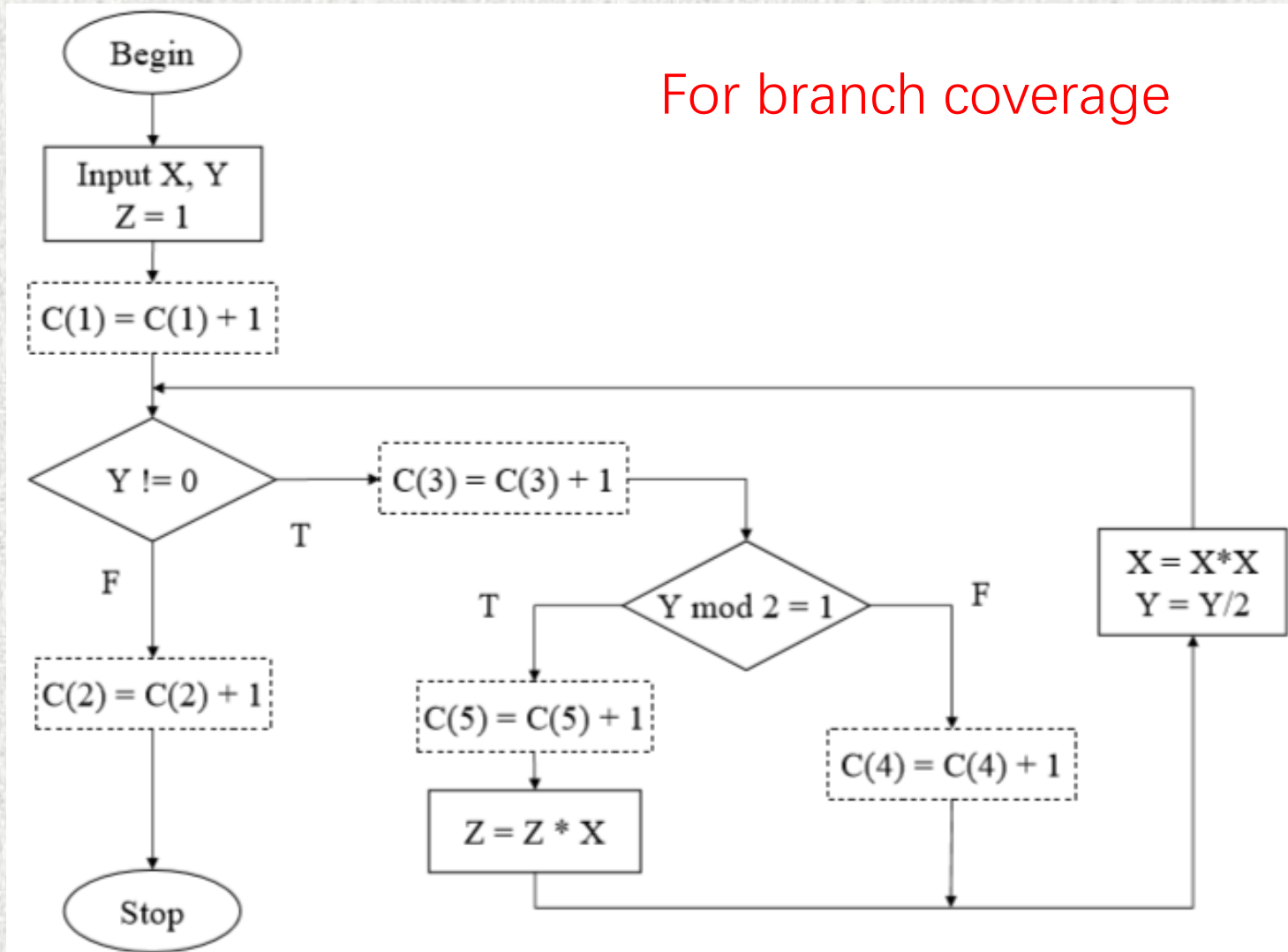


For statement coverage





For branch coverage



TEST CASES						
X	Y	C[1]	C[2]	C[3]	C[4]	C[5]
5	0	1	1	0	0	0
10	1	1	1	1	0	1
15	2	1	1	2	1	1
20	3	1	1	3	1	2
25	4	1	1	3	2	1

How about data flow analysis?

Two special examples for loop and if structure



# Quiz on March 20

- Control flow coverage;
- Data flow coverage