Minister of Education, Culture and Research of Moldova

Technical University of The Republic of Moldova

Software Engineering and Automatics

**REPORT**

Laboratory Work N.4

*Software Testing*

Performed by:

st. gr. FAF-161 G.Zaharia

Checked by: M.Catruc

a.univ.

Chisinau, 2019

**Prerequisites:**

* Projection of Test Cases.
* McCabe algorithm.
* Coverage criteria.
* Data flux control.
* White-Box technique.

**Objectives:**

* Understand the notion about test cases.
* Understand the methodology of development the application:
  + Test-Driven development
  + Code-Driven development
* Forming the partitioning skills in the coverage criteria.
* Use data flux graph.

**Tasks:**

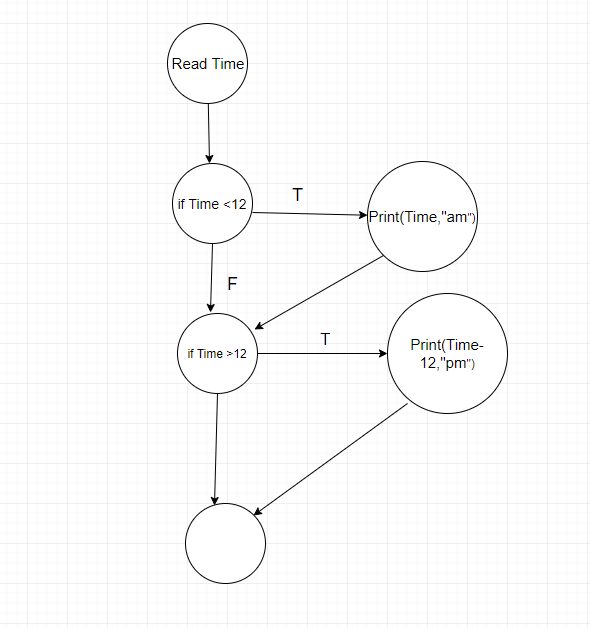
1. Draw the flow control graph of the program.
2. Test the program according to the coverage criteria
3. Evidence the sample where the erroneous results can be obtained for different coverage criteria.
4. Comment the special cases and the elaborated tests.
5. Use the McCabe technique to test the base ways.

**Implementation of tasks:**

1: **Begin**  
2: **Read Time**  
3: **If Time < 12 Then**  
4: **Print(Time, ”am”)**  
5: **Endif**  
6: **If Time > 12 Then**  
7: **Print(Time-12, “pm”)**  
8: **Endif**  
9: **End**

**Task 1:**

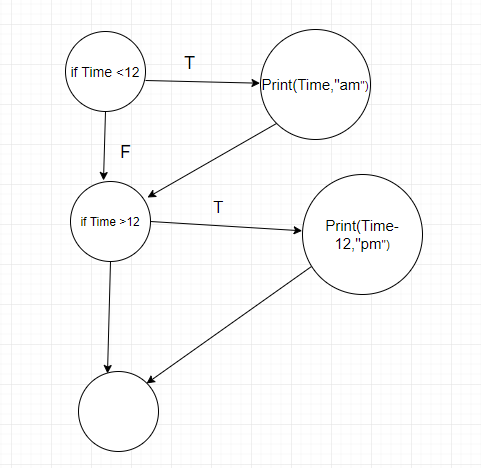
The Flow Control graph is one of 2 techniques of testing the transparent box.

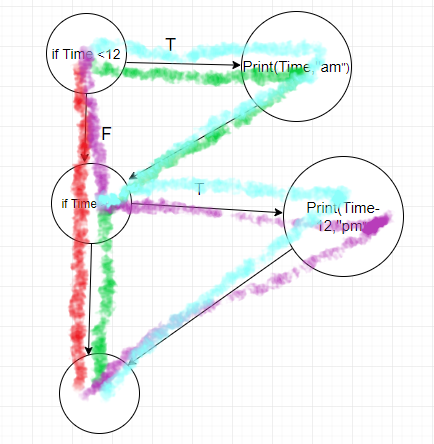


**Task 2 :**

So, in this task we have to use the coverage criteria for the our program. Also, we know that there are several types of covering.

**Level 0** : “100% covering the instructions”.



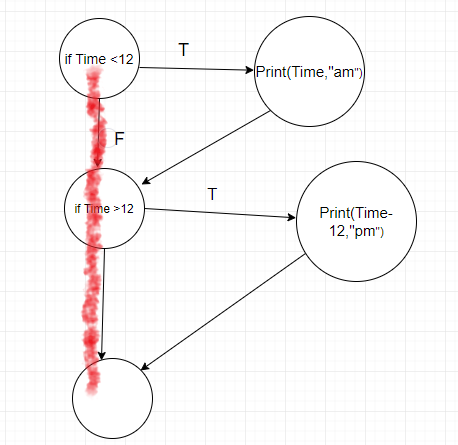
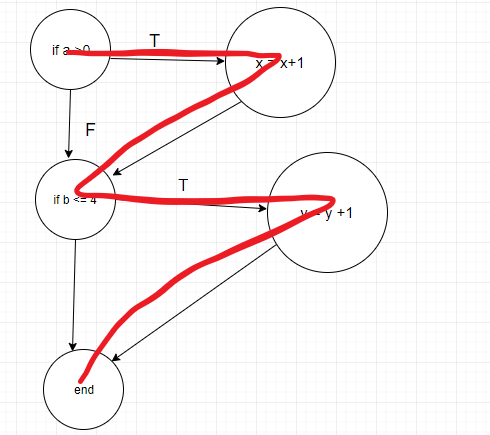


We can see that we can generate different ways of execution, in this case there are 4 ways. Let’s see if we can apply just one case on this type of covering criteria, as we can that is the lower that rest types of covering.

**Answer** : we can have just one test case, **Time = 12**. We obtained that this criteria will lose a lot of ways untested.

**Level 1 :** “100% covering of decisions or branch covering”.

In the previous example I can’t examine the both cases from this criteria, just the first, when the output will be **False**. For the second part , in case when the output must be **True** and all of the decisions should be coverage I will take another example.



a = 3 , b = 3.

**First case** : c1s1c2s3; c1c2s3 **Second case :** : c1s1c2s2s3

**Level 2:** “100% covering of conditions ”

This level also means that all conditions should be passed at least one time and one time the result should be **True** andsecond time should be **False.** So, I will give an example according to the second example of code.

**Cases :** a = 3; b = 3; a = 0; b = 5;

**Level 3:** “100% covering of decisions/conditions”

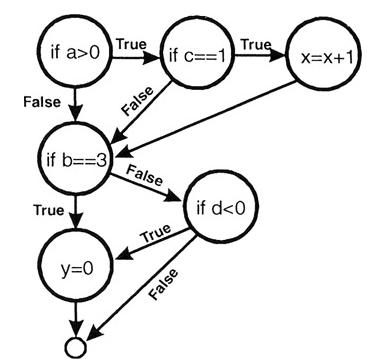
This level means that we can perform “and” logic between conditions, but this will not execute ever.

**Level 4:** “100% covering of multiple conditions”.

This level means to perform “or” logic between conditions. Let’s take another example of code for this type of level.

if (a>0 && c==1) {x=x+1;}

if (b==3 || d<0) {y=0;}

 a>0, c=1, b=3, d<0

a≤0, c=1, b=3, d≥0

a>0, c≠1, b≠3, d<0

a≤0, c≠1, b≠3, d≥0

**Task 4:**

We can’t end the testing and the process of performing the test cases, without testing the base ways. Structural testing is based on work of Tom McCabe, who used the typology of flow control graph for performing the test cases. So, for this process there are several steps:

* Extract the flow control graph to software.
* Calculate the cyclomatic number(C).
* Calculate a set of base ways(C).
* Create the test for each base way.
* Execute the tests.

**First code :** C = 6-5+2 = 3;  
**Second code:** C = 6-5+2 = 3;

|  |  |
| --- | --- |
| **Cyclomatic Complexity** | **Complexity level and Risk** |
| 1-10 | a simple program, without much risk |
| 11-20 | more complex, moderate risk |
| 21-50 | more complex, moderate risk |
| greater than 50 | untestable program , very high risk |

**Conclusion:**

In this laboratory work I obtained skills operating with White-Box testing and coverage criteria of this type of testing. Also, I learned about McCabe structural testing. Moreover, how to perform minimum tests for testing as more as possible ways of product.