Paths taken & Taken?: #4 1 0 0 0 | 1
#6 1 1 0 0 | 0
#8 0 1 0 0
#10 0 0 1 0
#12 1 1 1 0
#14 0 0 0 1
1 2 2 0 min is 1
2 1 2 0 min is 2
3 3 1 0 min is 1

VU Programm- und Systemverifikation

Assignment 3: Coverage Metrics

231

Name: Danny MILOSAVLJEVIC Matr. number: 0826039

Due: May 6, 4pm

Consider the following program fragment and test suite:

```
bool minmax (int i, int j, int k,
             int &least, int &most) {
#1   least = i;
#2   most = i;
#3   if (most < j)
#4     most = j;
#5   if (most < k)
#6     most = k;
#7   if (least > j)
#8     least = j;
#9   if (least > k)
#10    least = k;
#11  if (least < most)
#12    return true;
#13  if (least == most)
#14    return false;
#15 }
```

most := max {i, j, k}
least := min {i, j, k}

Inputs			Outputs		
i	j	k	least	most	result
1	2	3	1	3	true
2	1	3	1	3	true
2	2	1	1	2	true
0	0	0	0	0	false

#least > most impossible, but coverage doesn't care about "impossible".

Task 1 (3P): Control-Flow-Based Coverage Criteria Indicate (✓) which of the following coverage criteria are satisfied by the test-suite above.

probably each branch separately?

	satisfied	
Criterion	yes	no
path coverage		✓
statement coverage	✓	
branch coverage		✓
decision coverage		✓
condition/decision coverage		✓
MC/DC		✓

path through #4 but not #6 missing, see left-side markings, near code

no tests visits #15 (exit point)

branch coverage ⇒ decision coverage

Task 2 (7P): Data-Flow-Based Coverage Criteria Which of the following coverage criteria are satisfied by the test-suite above (don't count the parameters of the function as definitions):

	satisfied		reason
Criterion	yes	no	
all-defs	✓		all defs are used (by tests) (indirectly), vacously true. There are no c-uses.
all-c-uses	✓		
all-p-uses	✓		all uses in predicates are exercised.
all-c-uses/some-p-uses	✓		vacously all-c-uses, and all-p-uses.
all-p-uses/some-c-uses	✓		all-p-uses.
all-uses	✓		all-p-uses, and all-c-uses.
all-du-paths		✓	path through #4 but not through #6, eventually using most at #11 is not exercised.

Task 3 (1P): Cyclomatic Complexity What is the cyclomatic complexity of the program?

$2^6 = 64$

Task 4 (3P): Complete the Test Suite

- *If* the test-suite from above does not satisfy the coverage criteria listed below, augment it with test-cases such that these criteria are satisfied.
- If full coverage cannot be achieved for one or more of these criteria, explain why.
- If a coverage criterion is already satisfied, briefly explain why this is the case.

vacously. There are only uses of least and most in predicates.

all-c-uses

Inputs			Outputs		
i	j	k	least	most	result

all-p-uses/some-c-uses

Inputs			Outputs		
i	j	k	least	most	result

MC/DC

Inputs			Outputs		
i	j	k	least	most	result

it's impossible to reach #15, so MC/DC coverage cannot be achieved.

Task 5 (1P): Modified Condition/Decision Coverage Provide sufficiently many test-cases to guarantee modified condition/decision coverage for the following program fragment:

```
bool foo(int x, int y) {
  return ((x < y) || (y % 2 >= 1));
}
```

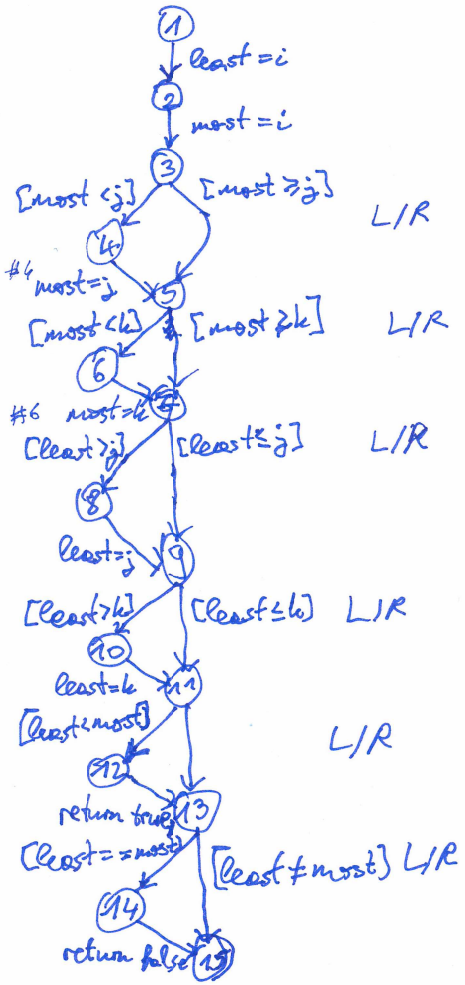
F
T
F

F
F
T

F
T
T

Input		Output
x	y	result
5	4	F
3	4	T
7	7	T

Please hand in your assignment via TUWEL (as a single PDF file) by May 6, 2015, 4pm.



L/R

L/R

L/R

L/R

L/R

$CC = 2^6 = 64$