Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

II SEMESTER M.TECH. (COMPUTER SCIENCE & ENGINEERING)

END SEMESTER EXAMINATIONS, MAY 2019

SUBJECT: SOFTWARE TESTING & ANALYSIS [CSE 5243]

REVISED CREDIT SYSTEM
(02/05/2019)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL** the questions.
- Missing data may be suitable assumed.

1A. Distinguish between static testing and dynamic testing. **3**

1B. Construct the control flow graph (CFG) for the following program in Fig. 1B.

```
begin
int num, product;
bool done;
product = 1;
input (done);
while (!done) {
    input (num);
    product = product * num;
    Input (done);
 }
output (product);
end
```

Fig. 1B

Define dominators and post-dominators for the CFG G = (N, E). Construct the dominator and post-dominator trees for the CFG of the above program in Fig. 1B. **4**

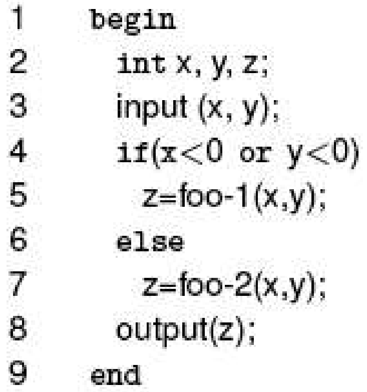
1C. With a neat diagram, explain equivalence partitioning technique. **3**

2A. Consider a program for classification of a triangle. Input to the program are three positive integers (say a,b,c) and the input parameters lies between 1 and 100. The triangle is classified according to the following rules:
Right angled triangle: <sum of squares of two sides is equal to square of other>
obtuse angled triangle : <sum of squares of two sides is less than square of other>
acute angled triangle : <sum of squares of two sides is greater than square of other>
Output can be one of the following:
Right angled triangle, obtuse angled triangle, acute angled triangle, Invalid triangle.
Design efficient Boundary value test suite based on input domain. **5**

2B. When should black box testing and white box testing be used during the software lifecycle respectively? Why? **2**

2C. Why do we go for equivalence class testing? Give an example where Equivalence class testing is better than BVA. **3**

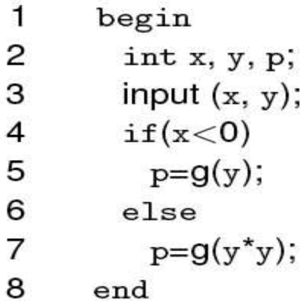
3A. Define condition/decision coverage criteria. Design a test set T for the program in Fig. 3A so that it is adequate with respect to condition/decision coverage criteria.



```
1   begin
2     int x, y, z;
3     input (x, y);
4     if(x<0 or y<0)
5       z=foo-1(x,y);
6     else
7       z=foo-2(x,y);
8     output(z);
9   end
```

Fig. 3A **2**

3B. Find all linear code sequence and jump (LCSAJ) for the following program in the Fig. 3B.



```
1   begin
2     int x, y, p;
3     input (x, y);
4     if(x<0)
5       p=g(y);
6     else
7       p=g(y*y);
8   end
```

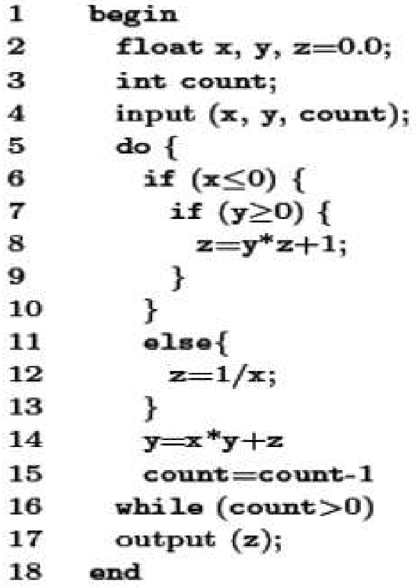
Fig. 3B

Is the following test set T covers all LCSAJs.

$$T = \left\{ \begin{array}{lll} t_1: & < x = -5 & y = 2 > \\ t_2: & < x = 9 & y = 2 > \end{array} \right\}$$

3

3C. Construct data flow graph for the following program in Fig. 3C. Compute **dcu** and **dpu** sets for all variables in the program showing its def, c-use and p-use.



```
1    begin
2      float x, y, z=0.0;
3      int count;
4      input (x, y, count);
5      do {
6        if (x≤0) {
7          if (y≥0) {
8            z=y*z+1;
9          }
10       }
11       else{
12         z=1/x;
13       }
14       y=x*y+z
15       count=count-1
16     while (count>0)
17     output (z);
18   end
```

Fig. 3C **5**

4A. Consider the following program P in Fig. 4A and four mutants

```
1. main(argc, argv)
2. int argc, r, i;
3. char *argv[ ];
4. { r = 1;
5. for i = 2 to 3 do
6. if (atoi(argv[i]) > atoi(argv[r])) r = i;
7. printf("Value of the rank is %d \n", r);
8. exit(0); }
```

Fig. 4A

Mutant 1: Change line 5 to for i = 1 to 3 do

Mutant 2: Change line 6 to if (i > atoi(argv[r])) r = i;

Mutant 3: Change line 6 to if (atoi(argv[i]) ≥ atoi(argv[r])) r = i;

Mutant 4: Change line 6 to if (atoi(argv[r]) > atoi(argv[r])) r = i;

Is test set $T = \{t_1 = (1, 2, 3)\ t_2 = (1, 2, 1), t_3 = (3, 1, 2)$ is adequate with respect to mutation testing? If not design an additional test case. **5**

4B. Explain Test Minimization procedure for regression testing. **3**

4C. With a neat diagram, explain regression testing selection problem. **2**

5A. Give Four differences between unit testing and integration testing. **4**

5B. Explain the FOUR Integration Testing strategies. **4**

5C. Discuss four types of integration errors. **2**