NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA-769008
MID-SEMESTER EXAMINATION, 2020
SESSION: 2019 – 2020 (Spring)

M.Tech.2nd Semester

Subject code: CS6404 **Subject Name:** Software Testing **Dept. Code:** CS
No. of pages: 2 **Full Marks:** 60 **Duration:** 2h*rs*

Figures at the right hand margin indicate marks.
All parts of a question should be answered at one place.
Answer all questions.

Q. No.	Particulars	Marks
1.		2x10=20
(a)	What do you understand by dynamic slice?	
(b)	Give a positive and a negative test scenario for the “Mobile Number” field of Student Registration Page.	
(c)	What do you mean by “dd” anomaly? What is its effect in data flow testing?	
(d)	A software was tested using the error seeding strategy in which 20 errors were seeded in the code. When the code was tested using the complete test suite, 16 of the seeded errors were detected. The same test suite also detected 200 non-seeded errors. What is the estimated number of latent errors in the software after this testing?	
(e)	Which testing techniques take into account the possible combinations of input conditions, while generating test cases?	
(f)	What is a stub? How is it associated with driver?	
(g)	What is the difference between verification and validation of software?	
(h)	What do you mean by equivalent mutants? Give an example of equivalent mutants.	
(i)	Give an example where it is not possible to determine the state of data variable by just static analysis of the code.	
(j)	Suppose a program contains 4 decision inputs, if there are 3 choices at each decision point, how many test cases are necessary for branch testing?	
2.		
(a)	Design the black-box test suite for a program that accepts two strings and checks if the first string is a substring of the second string and displays the number of times the first string occurs in the second string. Assume that each of the two strings has size less than twenty characters.	5
(b)	Consider a system having an FSM for a gaming software named Gumball with the following states and transitions: **States:** NoQuarter, HasQuarter, Sold, SoldOut, Error **Transitions:** NoQuarter to HasQuarter: addQuarter() NoQuarter to NoQuarter: turnCrank() HasQuarter to Sold: turnCrank() Sold to SoldOut: takeGumball() SoldOut to Error: turnCrank(), addQuarter() HasQuarter to Error: addQuarter() Sold to Sold: turnCrank() Sold to NoQuarter: takeGumball() Design test cases for this FSM using state table based testing.	8

(c) Consider a scenario in which you need to derive the test cases for a web page of a research paper, that has four different sections as follows: (i) Introduction, (ii) Basic Concepts (iii) Methodology (iv) Summary. These four sections can individually be shown or hidden to the user or show error message. Design the test conditions using orthogonal array to test the interaction between the different sections. 7

3.

(a) Write a function for finding the prime numbers in between 1-100. . 10

i. Construct the CFG of the above function.
ii. Determine its cyclometic complexity.
iii. Identify the linearly independent paths in the CFG.
iv. Design a test suite for the above function that satisfies path coverage.

(b) Consider the following program to find the greatest number among three numbers: 10

```
#include <stdio.h>
main()
{
        float x,y,z;
        clrscr();
        printf("enter the three variables x,y,z");
        scanf("%f %f %f",&x,&y,&z);
        if(x > y)
        {
                if(x > z)
                printf("greatest number =%f", x);
                else
                printf("("greatest number =%f", z");
        }
        else
                {
                if(y > z)
                printf("("greatest number =%f", y");
                else
                printf("("greatest number =%f", z");
        }
        getch();
}
```

Find out five possible mutants in the above program and check how many of them are killed by a set of test data. Add new test data if required. Show the mutation score achieved.