
MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

VII SEMESTER B.TECH. (COMPUTER SCIENCE AND ENGINEERING)

END SEMESTER EXAMINATIONS, NOV 2019

SUBJECT: SOFTWARE TESTING AND ANALYSIS [CSE 4020]

REVISED CREDIT SYSTEM
(23/11/2019)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

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

1A. Differentiate between the following with an example for each. **3M**
a. Deliverables and Milestones
b. Quality Assurance and Quality Control.

1B. ***Commission Problem***: A desert cooler sales person sold cooler fans, pumps and bodies that were made by a cooler maker. Fans cost $45, pumps cost $30 and bodies cost $25. The salesperson had to sell at least one complete cooler per month, and the production limits were such that the most the sales person could sell in a month was 70 fans, 80 pumps and 90 bodies. The sales person used to send the details of sold items to the cooler maker. The cooler maker then computed the sales person's commission as follows: **2M**

- 10% on sales upto and including $1000.
- 15% of the next $800.
- And 20% on any sales in excess of $1800.

The commission program produced a monthly sales report that gave the total number of fans, pumps and bodies sold, the sales person's total dollar sales and finally, the commission."
Generate BVA Test Cases for this problem?

1C. For what kind of programs is the decision table based testing more appropriate? What is the relationship between decision tables and test cases? Design a decision table (show the various steps involved) to represent the following Hoosier Burger inventory reordering function. **5M**
Reordering depends on number of seasons
Academic Year: When there is an academic season check whether the item is perishable or nonperishable. If an item is perishable, such as meat, vegetables, milk etc., the inventory system has a standing daily order with a local supplier stating that a pre-specified amount of food is delivered each weekday for that day's use and a standing weekend order with a local supplier each Saturday for weekend use. If an item is nonperishable, such as straws, cups and napkins, a minimum order quantity is placed when the stock on hand reaches a certain predetermined minimum reorder quantity. Hoosier Burger's business is not as good during summer, Christmas and spring break holidays as it is during the Academic year.
Summer Season: The standing orders with all their suppliers are reduced by specific amounts during the summer i.e. summer reduction.
Holiday Season: The standing orders with all their suppliers are reduced by specific amounts during the holiday break i.e. Holiday reduction.

2A. Consider the given C function cgi_decode in Fig. 2A., which translates a cgi-encoded string to a plain ASCII string. The function translates a string from the CGI encoding to plain ascii text. If the input contains '+' then it becomes space, %xx becomes byte with hex value xx, other alphanumeric characters map to themselves and the function returns 0 for success, positive for erroneous input and 1 for bad hexadecimal digit. Identify the basic blocks and draw the control glow graph. **3M**

```
int cgi_decode(char *encoded, char *decoded) {
  char *eptr = encoded;
  char *dptr = decoded;
  int ok=0;
  while (*eptr) {
    char c;
    c = *eptr;
    /* Case 1: '+' maps to blank */
    if (c == '+') {
      *dptr = ' ';
    } else if (c == '%') {
      /* Case 2: '%xx' is hex for character xx */
      int digit_high = Hex_Values[*(++eptr)];
      int digit_low  = Hex_Values[*(++eptr)];
      /* Hex_Values maps illegal digits to -1 */
      if ( digit_high == -1 || digit_low == -1 ) {
        /* *dptr='?'; */
        ok=1; /* Bad return code */
      } else {
        *dptr = 16* digit_high + digit_low;
      }
      /* Case 3: All other characters map to themselves */
    } else {
      *dptr = *eptr;
    }
    ++dptr;
    ++eptr;
  }
  *dptr = '\0';              /* Null terminator for string */
  return ok;
}
```

Fig 2A.

2B. For the function given in Fig 2A, calculate the statement coverage for given the test suite Tp={" ","test","test+case%1Dadequacy"}. Design a test suite to achieve 100% statement coverage. **3M**

2C. Why do we undertake robustness testing? What are the additional benefits? Explain with an example. **4M**

3A. With the help of an example, explain the need of multiple condition coverage in control flow testing. **2M**

3B. Identify the valid and invalid equivalence classes for the commission problem in Q1B. Design the strong robust test cases for the same. **3M**

3C. Define test adequacy based on Linear Code Sequence and Jump (LCSAJ) coverage. Find all LCSAJ for the following program in the Fig. 3C. Generate test set T containing two test cases so that T is adequate with respect to LCSAJ coverage criteria **5M**

```
1    begin
2    // Compute x^y given non-negative integers x and y.
3      int x, y, p;
4      input (x, y);
5      p=1;
6      count=y;
7      while(count>0){
8        p=p*x;
9        count=count-1;
10     }
11     output(p);
12   end
```

Fig. 3C.

4A. Identify basic blocks and hence construct dataflow graph for the following program in Fig. 4A specifying **def**, **c-use** and **p-use** for each node. Show the table for **def**, **dcu set** and **dpu set** for all variables in the program. Generate test set T containing two test cases so that T is adequate with respect to P-use coverage criteria. **5M**

```
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. 4A.

4B. Define mutation score for the test set T. Consider the following program P in Fig. 4B. and four mutants **5M**

```
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. 4B.

M1: Change line 5 to for i = 1 to 3 do
M2: Change line 6 to if (i > atoi(argv[r])) r = i;
M3: Change line 6 to if (atoi(argv[i]) ≥ atoi(argv[r])) r = i;
M4: 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)} adequate w.r.t mutation testing criterion? If not, design an additional test case. With a neat table, show the output of P and mutants for each test case.

5A. With neat figure explain Regression Testing Selection (RTS) problem. Give one suitable example for RTS problem. **3M**

5B. With a neat figure, explain the four types of integration errors. **3M**

5C. Define stubs with suitable example. Give two important advantages of stubs. **4M**