Software Testing

Assignment- 3

TYPE OF QUESTION: MCQ/MSQ

Number of questions: **10**   **Total mark: 10 X 1 = 10**

For each of the following questions one or more of the given options are correct. Choose the correct option(s).

QUESTION 1:

Which of the following pairs of white-box test techniques are complementary test techniques?

      a. Statement-coverage and path coverage

      b. Multiple condition coverage and MC/DC testing

      c. MC/DC and Data flow testing

      d. Multiple condition coverage and decision coverage

      e. Path coverage and MC/DC testing

Correct Answer: c. MC/DC and Data flow testing

               e. Path coverage and MC/DC testing

Detailed Solution:

MC/DC is complementary to both dataflow and path coverage test techniques. Please refer slide 10 of week 3 lecture material for more information.

QUESTION 2:

Which of the following are the measures of the testing difficulty of a program segment?

      a. Number of statements in the program segment

      b. Number of decision statements in the program segment

      c. Complexity of the arithmetic expressions used in the program segment

      d. Mc Cabe's cyclometric complexity for the program segment

      e. Number of arithmetic expressions used in the program segment

Correct Answer: b. Number of decision statements in the program segment

               d. Mc Cabe's cyclometric complexity for the program segment

Detailed Solution:

Number of decision statement in the program measure the testing difficulty and since cyclomatic complexity of a program depends on number of decision statements, it also measure the testing difficulty of a program segment.

Questions 3 to 7 are based on the following "C" code segment.

```
int main (){
        int a,b=0;

        scanf("%d",&a);

        if( a < 10)    b=b+10;
        else           b=b+20;

        if( a == 30 )  b=b+30;
        else           b=b+40;

}
```

QUESTION 3:

At least how many test cases are needed for achieving decision coverage on the given C code?
   a) 2
   b) 3
   c) 5
   d) 6
   e) 7

 Correct Answer: a. 2

 Detailed Solution:

 Decision Coverage is a white box testing technique which reports the true or false outcomes of each boolean expression of the source code. a=5 will give TRUE and False. a=30 will give False and TRUE. So, two test cases required for achieving decision coverage.

QUESTION 4:

At least how many test cases are needed for the given C code for achieving basic condition coverage?
   a) 2
   b) 3
   c) 5
   d) 6
   e) 8

 Correct Answer: a. 2

 Detailed Solution:
In basic condition coverage, each atomic condition made to assume both T and F values. So, in this case using two test case we can achieve basic condition coverage.

QUESTION 5:

At least how many test cases are needed for the given C code for achieving multiple condition coverage?

        a) 2
        b) 4
        c) 8
        d) 16
        e) Multiple condition coverage is not achievable for the given code

Correct Answer: a. 2

Detailed Solution:
Atomic conditions made to assume all possible combinations of truth values. Here for both the condition only one variable 'a' used. So, for achieving multiple condition coverage we just need to check 'a' is TRUE and 'a' is false. Two test cases are sufficient to achieve this.

QUESTION 6:

At least how many test cases are needed for the given C code for achieving MC/DC coverage?

        a) 2
        b) 4
        c) 8
        d) 16
        e) MC/DC coverage is not achievable for the given code

Correct Answer: a. 2

Detailed Solution:
In our example there is no compound decision. So, in this case also we just need to check a=TRUE and a=False. Therefore, only two test cases are sufficient to achieve this.

QUESTION 7:

At least how many test cases are needed for the given C code for achieving basis path coverage?
- a) 2
- b) 3
- c) 5
- d) 7
- e) Basis path coverage is not achievable for the given code

Correct Answer: b. 3

Detailed Solution:
In our example, there are two decision. For two decision there are four combination: TT, TF, FT, FF. But in this example TT is cannot be possible at same time. We just need to cover TF, FT and FF paths. So, for three paths three test cases required. a=5 will give TF. a=30 will give FT and a=20 will give FF.

QUESTION 8:

At least how many test cases are required to achieve multiple condition coverage of the following code segment:

If((a>5) and (b<100) and (c>50)) x=x+1;
- a) 2
- b) 4
- c) 6
- d) 8
- e) 16

Correct Answer: d. 8

Detailed Solution:

Total independent condition =3. So, test cases required for multiple condition coverage:
2^3 =8.

QUESTION 9:

For a given function, the computed McCabe's Cyclometric complexity metric does not correlate with which of the following?
   a. Number of errors existing in the function after unit testing is complete
   b. Time required to find the location of an error after a failure.
   c. Number of statements in the program
   d. Difficulty of understanding the function.
   e. Number of test cases required to achieve MC/DC

Correct Answer: c. Number of statements in the program
                e. Number of test cases required to achieve MC/DC

Detailed Solution:
Cyclometric complexity metric indicate the number of errors existing in the code,
the time required to find and correct the errors, difficulty of understanding a program. It also
indicates number of test cases required for path testing but it neither indicate number of test cases
required to achieve MC/DC nor the number of statements present in the program.

QUESTION 10:

What is the McCabe's Cyclomatic complexity for the following code segment?

```
int partition (int arr[], int l, int h){
        int x = arr[h], i = (l - 1),t;

        for (int j = l; j <= h- 1; j++) {
                if (arr[j] <= x){
                        i++;
                        t=&arr[i]; &arr[i]= &arr[j]; &arr[j]=t;
                }
        }
        t=&arr[i + 1];
        &arr[i+1])= &arr[h];
        &arr[h]=t;
}
```
   a. 2
   b. 3
   c. 4
   d. 5
   e. 6

Correct Answer: b. 3

Detailed Solution:
McCabe's Cyclomatic Complexity = 2(i.e., no. of decision point) + 1 =3

_____

************END***********