**Consider an office email system which is designed based on the following specifications:**
**1. Send email when recipient address has been mentioned, subject is present, and time is before 5:00pm.**
**2. If recipient address is missing or subject is missing. give warning message.**
**3, If time is after 5:00pm, then put the email in outbox folder.**

**a) Identify the conditions and actions for the given case. Also, derive the decision table for the problem.**
**b) Design the test cases with values of inputs and outputs.**

Sol.

a)

Conditions:

1. Recipient address is present
2. Subject is present
3. Time is before 5:00pm

Actions:

1. Send email
2. Give warning message
3. Put email in outbox folder

Decision table:

| Condition 1 | Condition 2 | Condition 3 | Action |
|---|---|---|---|
| Present | Present | Before 5pm | Send email |

| | | | |
|---|---|---|---|
| Missing | Present | Before 5pm | Give warning |
| Present | Missing | Before 5pm | Give warning |
| Present | Present | After 5pm | Put in outbox |
| Present | Missing | After 5pm | Put in outbox |
| Missing | Present | After 5pm | Put in outbox |
| Missing | Missing | Before 5pm | Give warning |
| Missing | Missing | After 5pm | Put in outbox |

b)

Test cases:

1. Recipient address present, subject present, time before 5:00pm - Expect email to be sent
2. Recipient address missing, subject present, time before 5:00pm - Expect warning message

3. Recipient address present, subject missing, time before 5:00pm - Expect warning message
4. Recipient address present, subject present, time after 5:00pm - Expect email to be put in outbox
5. Recipient address present, subject missing, time after 5:00pm - Expect email to be put in outbox
6. Recipient address missing, subject present, time after 5:00pm - Expect email to be put in outbox
7. Recipient address missing, subject missing, time before 5:00pm - Expect warning message
8. Recipient address missing, subject missing, time after 5:00pm - Expect email to be put in outbox

**An electric geyser is to be installed with two-way switches (S1 and S2) . Switch S1 is to be installed on the ground floor of the house, while switch S2 is to be installed on the first floor of the house. The geyser will be ON only if both the switches S1 and S2 are in ON position or both are in OFF position. The geyser will be OFF for all other positions of S1 and S2.**

**a) Identify the conditions and actions for the given case. Also, derive the decision table for the problem.**
**b) Design the test cases with values of inputs and outputs.**
**c) Draw the cause-effect graph for the same.**

Sol.

a) Conditions:
1. If Switch S1 is ON and Switch S2 is ON => Geyser is ON
2. If Switch S1 is OFF and Switch S2 is OFF => Geyser is ON
3. For all other conditions (Switch S1 ON and Switch S2 OFF, or Switch S1 OFF and Switch S2 ON) => Geyser is OFF

Actions:

1. Geyser is ON
2. Geyser is OFF

Decision Table:

| Switch S1 | Switch S2 | Geyser |
|---|---|---|
| ON | ON | ON |
| OFF | OFF | ON |
| ON | OFF | OFF |
| OFF | ON | OFF |

b) Test Cases:

| Inputs | Outputs |
|---|---|
| Switch S1 ON, Switch S2 ON | Geyser ON |
| Switch S1 OFF, Switch S2 OFF | Geyser ON |
| Switch S1 ON, Switch S2 OFF | Geyser OFF |
| Switch S1 OFF, Switch S2 ON | Geyser OFF |

c) Cause-Effect Graph:

[Switch S1] → [Geyser]
↓
[Switch S2] → [Geyser]

**Q.**

**Function Point estimation is a technique used in the early stages of the project to determine the size of the project based on software requirements. The calculation of size is based on functions supported by the software and the data files used by the software. Data files are of 2 types : Internal Logical Files (ILF) and External Interface Files (EIF). Complexity of files is determined based on 2 characteristics of the file namely - Data Element Types (DET) and Record Element Types (RET), based on the matrix given below:**

| **ILF/EIF** | **1-19 DETs** | **20-50 DETs** | **51+DETs** |
|---|---|---|---|
| **1 RET** | **Low** | **Low** | **Average** |
| **2-5 RETs** | **Low** | **Average** | **High** |
| **6+ RET** | **Average** | **High** | **High** |

**Software is designed to automate the process of calculating the complexity of the ILF based on the number of DETs and RETs**

a) **Derive the valid and invalid sub-domains of the input variables. Justify any specific choice you make for the sub-domains**
b) **Calculate the number of test cases for Strong Robust Equivalence Class testing. State clearly the sub–domains chosen.**
c) **Write down all the test cases for Strong Robust Equivalence Class testing.**

Sol.

a) Valid sub-domains for the input variables are as follows:

- RET: 1, 2, 3, 4, 5, 6+
- DET: 1-19, 20-50, 51+

The invalid sub-domains for the input variables are as follows:

- RET: Negative values or values greater than 6 are not valid.
- DET: Negative values or values greater than 51 are not valid.

The sub-domains have been chosen based on the characteristics of the files as given in the matrix, which define the complexity of the files based on the number of RETs and DETs.

b) The number of test cases for Strong Robust Equivalence Class testing can be calculated as follows:

For RET:

- 3 valid sub-domains (1, 2, 3), 1 invalid sub-domain (negative values)
- Total 4 RET sub-domains

For DET:

- 3 valid sub-domains (1-19, 20-50, 51+), 2 invalid sub-domains (negative values and values greater than 51)
- Total 5 DET sub-domains

Total number of test cases = (number of valid RET sub-domains * number of valid DET sub-domains) + (number of invalid RET sub-domains * number of invalid DET sub-domains)
= (3 * 3) + (1 * 2)

= 11 test cases

c) The test cases for Strong Robust Equivalence Class testing are as follows:

1. RET = -1, DET = -1
2. RET = 1, DET = -1
3. RET = 2, DET = -1
4. RET = 3, DET = -1
5. RET = -1, DET = 1
6. RET = -1, DET = 20
7. RET = -1, DET = 51
8. RET = 1, DET = 1
9. RET = 1, DET = 20
10. RET = 1, DET = 51
11. RET = 6, DET = 51

These test cases cover all the valid and invalid sub-domains of the input variables for Strong Robust Equivalence Class testing.

Sol2.

a) Valid sub-domains for the input variables are as follows:

- RET: 1, 2, 3, 4, 5, 6+
- DET: 1-19, 20-50, 51+

The invalid sub-domains for the input variables are as follows:

- RET: Negative values or values greater than 6 are not valid.
- DET: Negative values or values greater than 51 are not valid.

The sub-domains have been chosen based on the characteristics of the files as given in the matrix, which define the complexity of the files based on the number of RETs and DETs.

b) The number of test cases for Strong Robust Equivalence Class testing can be calculated as follows:

For RET:

- 3 valid sub-domains (1, 2, 3), 1 invalid sub-domain (negative values)
- Total 4 RET sub-domains

For DET:

- 3 valid sub-domains (1-19, 20-50, 51+), 2 invalid sub-domains (negative values and values greater than 51)
- Total 5 DET sub-domains

Total number of test cases = (number of valid RET sub-domains * number of valid DET sub-domains) + (number of invalid RET sub-domains * number of invalid DET sub-domains)
= (3 * 3) + (1 * 2)

= 11 test cases

c) The test cases for Strong Robust Equivalence Class testing are as follows:

1. RET = -1, DET = -1
2. RET = 1, DET = -1
3. RET = 2, DET = -1
4. RET = 3, DET = -1
5. RET = -1, DET = 1
6. RET = -1, DET = 20
7. RET = -1, DET = 51
8. RET = 1, DET = 1
9. RET = 1, DET = 20
10. RET = 1, DET = 51
11. RET = 6, DET = 51

These test cases cover all the valid and invalid sub-domains of the input variables for Strong Robust Equivalence Class testing.

**Suppose a user can operate a typical washing machine motor in three different modes namely : Soft, Normal and Strong. Before starting the motor in any mode, the values of two parameters namely ?A? and ?B? are to be set. The range (maximum and minimum) of valus that parameter ?A? and ?B? can take in various modes is shown in the following table :**

**a) Mode : Soft, Possible ranges of A : 40 <= A <= 70, Possible ranges of B : 10 <= B <= 15**
**b) Mode : Normal, Possible ranges of A : 70 < A <= 80, Possible ranges of B : 15 < B <= 25**
**c) Mode : Strong, Possible ranges of A : 80 < A <= 90, Possible ranges of B : 25 < B <= 35**

**a) Derive the valid and invalid sub-domains for the input variables. Justify any specific choice you make for the sub-domains.**
**b) Calculate the number of test cases for Strong Robust Equivalence Class testing. State clearly the sub-domains chosen.**
**c) Write down all the test cases for Strong Robust Equivalence Class testing.**

**Sol.**

a)

Valid sub-domains:

- Mode: Soft, A: 40 <= A <= 70, B: 10 <= B <= 15
- Mode: Normal, A: 70 < A <= 80, B: 15 < B <= 25
- Mode: Strong, A: 80 < A <= 90, B: 25 < B <= 35

Invalid sub-domains:

- Mode: Soft, A < 40 or A > 70 or B < 10 or B > 15
- Mode: Normal, A <= 70 or A > 80 or B <= 15 or B > 25
- Mode: Strong, A <= 80 or A > 90 or B <= 25 or B > 35

The choice of sub-domains is based on the given ranges for each mode of operation. We have identified valid sub-domains based on the given ranges for each mode, and invalid sub-domains by considering values outside these ranges.

b)

For Strong Robust Equivalence Class testing, we need to choose one valid and one invalid value for each input variable in each sub-domain.

Valid sub-domains:

1. Mode: Soft
- A=40, B=10 (valid)
- A=71, B=16 (invalid)
2. Mode: Normal
- A=71, B=16 (valid)
- A=81, B=26 (invalid)
3. Mode: Strong
- A=81, B=26 (valid)
- A=91, B=36 (invalid)

Invalid sub-domains:

1. Mode: Soft
- A=39, B=9 (invalid)
- A=72, B=16 (valid)
2. Mode: Normal
- A=70, B=15 (invalid)
- A=81, B=26 (valid)
3. Mode: Strong
- A=80, B=25 (invalid)
- A=91, B=36 (valid)

c)

Test cases for Strong Robust Equivalence Class testing:

1. Mode: Soft
- A=40, B=10 (valid)
- A=71, B=16 (invalid)

2. Mode: Normal
- A=71, B=16 (valid)
- A=81, B=26 (invalid)
3. Mode: Strong
- A=81, B=26 (valid)
- A=91, B=36 (invalid)
4. Mode: Soft
- A=39, B=9 (invalid)
- A=72, B=16 (valid)
5. Mode: Normal
- A=70, B=15 (invalid)
- A=81, B=26 (valid)
6. Mode: Strong
- A=80, B=25 (invalid)
- A=91, B=36 (valid)

**An organization has following criteria for promotion for its employees :**
**a) Average working hours per day is between 9 to 10**
**b) His / her minimum experience years should be 10 and 20**
**c) His / her age should be in the range of 25 to 50 years**

**a) Derive the total number of test cases that will be needed for BVA, Robust BVA, Worst Case BVA and Robust Worst case BVA**
**b) Write all the Robust BVA test cases**

a)

To calculate the total number of test cases needed for Boundary Value Analysis (BVA), we need to consider the input ranges and their boundaries, and apply the following formulas:

- Total number of test cases for BVA = 2n + 1, where n is the number of input variables
- Total number of test cases for Robust BVA = 3n + 1
- Total number of test cases for Worst Case BVA = 2^n
- Total number of test cases for Robust Worst Case BVA = 4^n

In this case, we have three input variables: average working hours, experience years, and age.

For the average working hours, the input range is between 9 to 10 hours per day. The boundaries are 8, 9, 9.5, 10, and 11.

For the experience years, the input range is between 10 to 20 years. The boundaries are 9, 10, 15, 20, and 21.

For the age, the input range is between 25 to 50 years. The boundaries are 24, 25, 37.5, 50, and 51.

Therefore, we have:

- Total number of test cases for BVA = 2^3 + 1 = 9
- Total number of test cases for Robust BVA = 3^3 + 1 = 28
- Total number of test cases for Worst Case BVA = 2^3 = 8

- Total number of test cases for Robust Worst Case BVA = 4^3 = 64

b)

The Robust BVA test cases for this promotion criteria would be:

1. Test case 1: Average working hours = 8, Experience years = 9, Age = 24
2. Test case 2: Average working hours = 8, Experience years = 9, Age = 25
3. Test case 3: Average working hours = 8, Experience years = 9, Age = 37.5
4. Test case 4: Average working hours = 8, Experience years = 9, Age = 50
5. Test case 5: Average working hours = 8, Experience years = 9, Age = 51
6. Test case 6: Average working hours = 8, Experience years = 10, Age = 24
7. Test case 7: Average working hours = 8, Experience years = 10, Age = 25
8. Test case 8: Average working hours = 8, Experience years = 10, Age = 37.5
9. Test case 9: Average working hours = 8, Experience years = 10, Age = 50
10. Test case 10: Average working hours = 8, Experience years = 10, Age = 51
11. Test case 11: Average working hours = 8, Experience years = 15, Age = 24
12. Test case 12: Average working hours = 8, Experience years = 15, Age = 25
13. Test case 13: Average working hours = 8, Experience years = 15, Age = 37.5
14. Test case 14: Average working hours = 8, Experience years = 15, Age = 50
15. Test case 15: Average working hours = 8, Experience years = 15, Age = 51
16. Test case 16: Average working hours = 8, Experience years = 20, Age = 24