Test case

Decision Tables (Cause-Effect table: Number of possible Combinations is given by 2 ^ n )

Cause effect graph

Equalence class testing (with all variation) -with test case- have format

- Equalence testing,

- week robust,

- strong robust,

- week normal,

- strong normal

- plotting graph

The number of test cases for Strong Robust Equivalence Class testing can be calculated using the formula:

N = ((k^2) x (p^m)) + ((k x q^m) x (p^(m-1)))

where:

k is the number of valid input values for each input variable

p is the number of modes (3 in this case)

m is the number of input variables (2 in this case)

q is the number of invalid input values for each input variable (1 in this case)

Boundary value analysis (BVA) (with all variation)- with test case - have format

- robust BVA

- worst case BVA

- robust worst case BVA

- Formula (very important)

- triangle example

- plotting graph

**BVA or Normal BVA= 4n + 1**

**Robust BVA = 6n +1**

**Worst case BVA= 5^n**  =

**Robust Worst case BVA= 7^n** =

* n = number of inputs
* Here n is 3 (conditions)

Code Based Techniques

-Control Flow

-Data Flow

Graph theory

McCabe

(optional)

Usage based testing

- Operational Profile

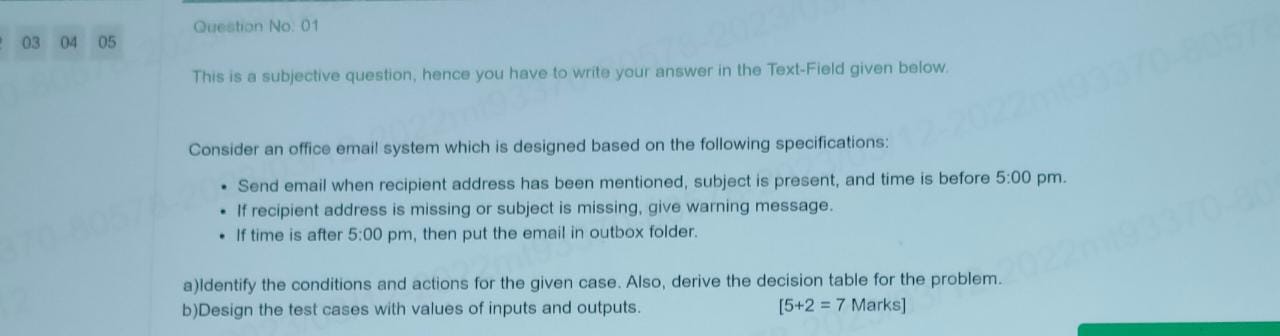
- Reliability Engineered Testing

Based on Engineers experience and intuition

- Exploratory

- Ad-hoc

Q1 :



A :

C1: recipient address present

C2: subject is present

C3: time < 5:00 PM

A1: send email

A2: warning message

A3: outbox folder

A4: impossible

**2^3(conditions) = 2\*2\*2= 8 (Rules)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions | Rule1 | Rule2 | Rule3 | Rule4 | Rule5 | Rule6 | Rule7 | Rule8 |
| C1 | F | T | T | T | T | F | F | F |
| C2 | F | T | F | F | T | T | F | T |
| C3 | F | T | F | T | F | T | T | F |
| Actions |  |  |  |  |  |  |  |  |
| A1 |  | T |  |  |  |  |  |  |
| A2 |  |  | T | T |  | T | T | T |
| A3 |  |  | T |  | T |  |  | T |
| A4 | T |  |  |  |  |  |  |  |

**2^n – n is rules**

2^3(conditions) = 2\*2\*2= 8 (Rules)

B:

TC 1: Email should be valid email address

TC 2: Mail should send before 5:00PM

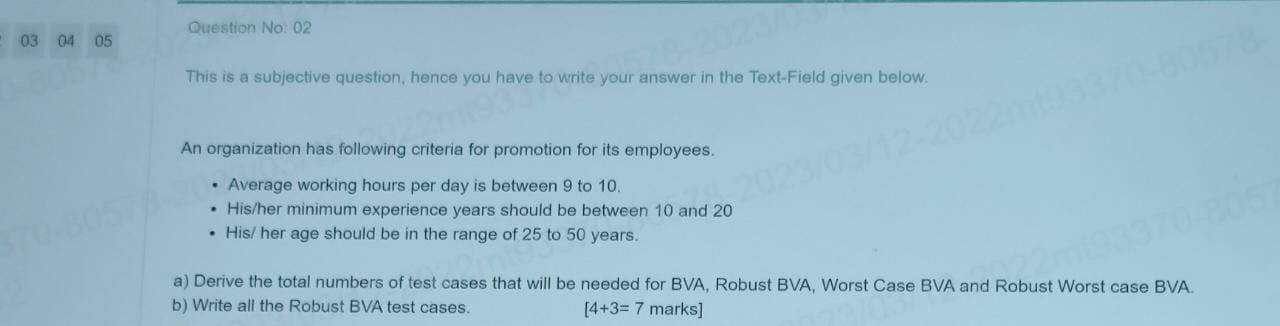
Tc 3: Subject should not be empty

Tc 4: If mail is sent after 5:00 PM, then mail should be stored in outbox

Tc 5:

…

Q2:



Sol:

a)

**BVA or Normal BVA=4n + 1** = 13

**Robust BVA = 6n +1 =** 19

**Worst case BVA= 5^n**  = 125

**Robust Worst case BVA= 7^n** = 343

* n = number of inputs
* Here n is 3 (conditions)

b)

Robust BVA

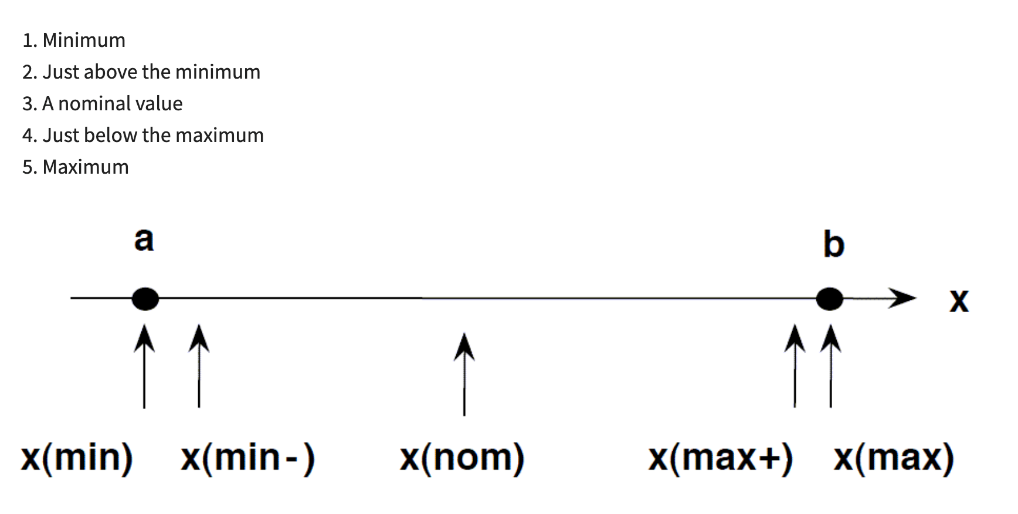
<https://t4tutorials.com/what-is-robust-case-testing-software-testing/#google_vignette>

BVA

<https://t4tutorials.com/what-is-simple-boundary-value-testing-software-testing/>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | X (working hours) | Y (experience) | Z (age) |  |
| Min- | 08:45 | 9 | 24 |  |
| Min | 09:00 | 10 | 25 |  |
| Min + | 09:15 | 11 | 26 |  |
| Nominal | 09:30 | 15 | 37 |  |
| Max- | 09:45 | 19 | 49 |  |
| Max | 10:00 | 20 | 50 |  |
| Max + | 10:15 | 21 | 51 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Test case | X (working hours) | Y (experience) | Z (age) | Expected output |
| 1 | 09:30 | 15 | 24 | Not promoted |
| 2 | 09:30 | 15 | 25 | Promoted |
| 3 | 09:30 | 15 | 26 | Promoted |
| 4 | 09:30 | 15 | 37 | Promoted |
| 5 | 09:30 | 15 | 49 | Promoted |
| 6 | 09:30 | 15 | 50 | Promoted |
| 7 | 09:30 | 15 | 51 | Not promoted |
| 8 | 09:30 | 9 | 37 | Not promoted |
| 9 | 09:30 | 10 | 37 | Promoted |
| 10 | 09:30 | 11 | 37 | Promoted |
| 11 | 09:30 | 19 | 37 | Promoted |
| 12 | 09:30 | 20 | 37 | Promoted |
| 13 | 09:30 | 21 | 37 | Not promoted |
| 14 | 08:45 | 15 | 37 | Not promoted |
| 15 | 09:00 | 15 | 37 | Promoted |
| 16 | 09:15 | 15 | 37 | Promoted |
| 17 | 09:45 | 15 | 37 | Promoted |
| 18 | 10:00 | 15 | 37 | Promoted |
| 19 | 10:15 | 15 | 37 | Not promoted |

Based on below diagram we have to write test case, for eg:



Q3. Graphical user interface, text

Description automatically generated

Ans:

Condition (C), Action (A)

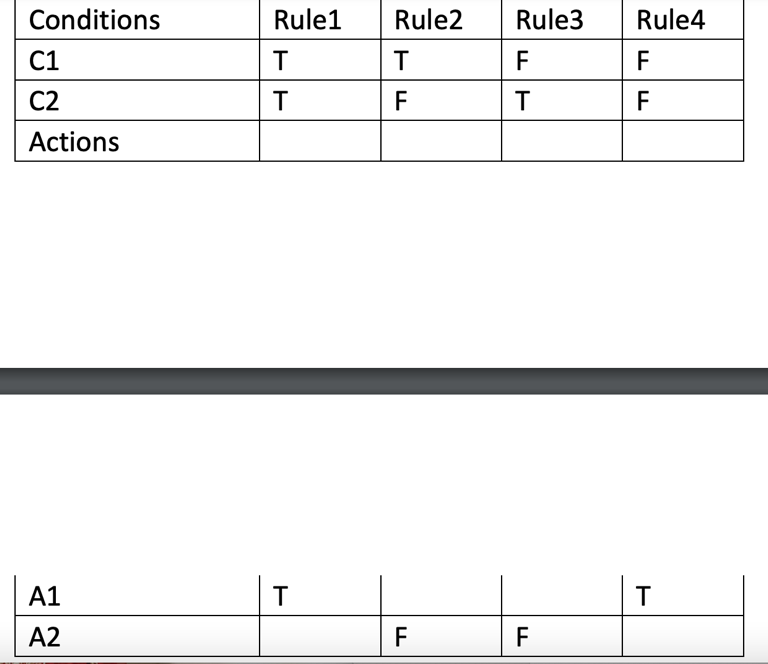
C1: s1

C2: S2

A1: Geyser is ON

A2: Geyser is OFF

**2^n – n is rules** 2^2(conditions) = 2\*2= 4 (Rules)



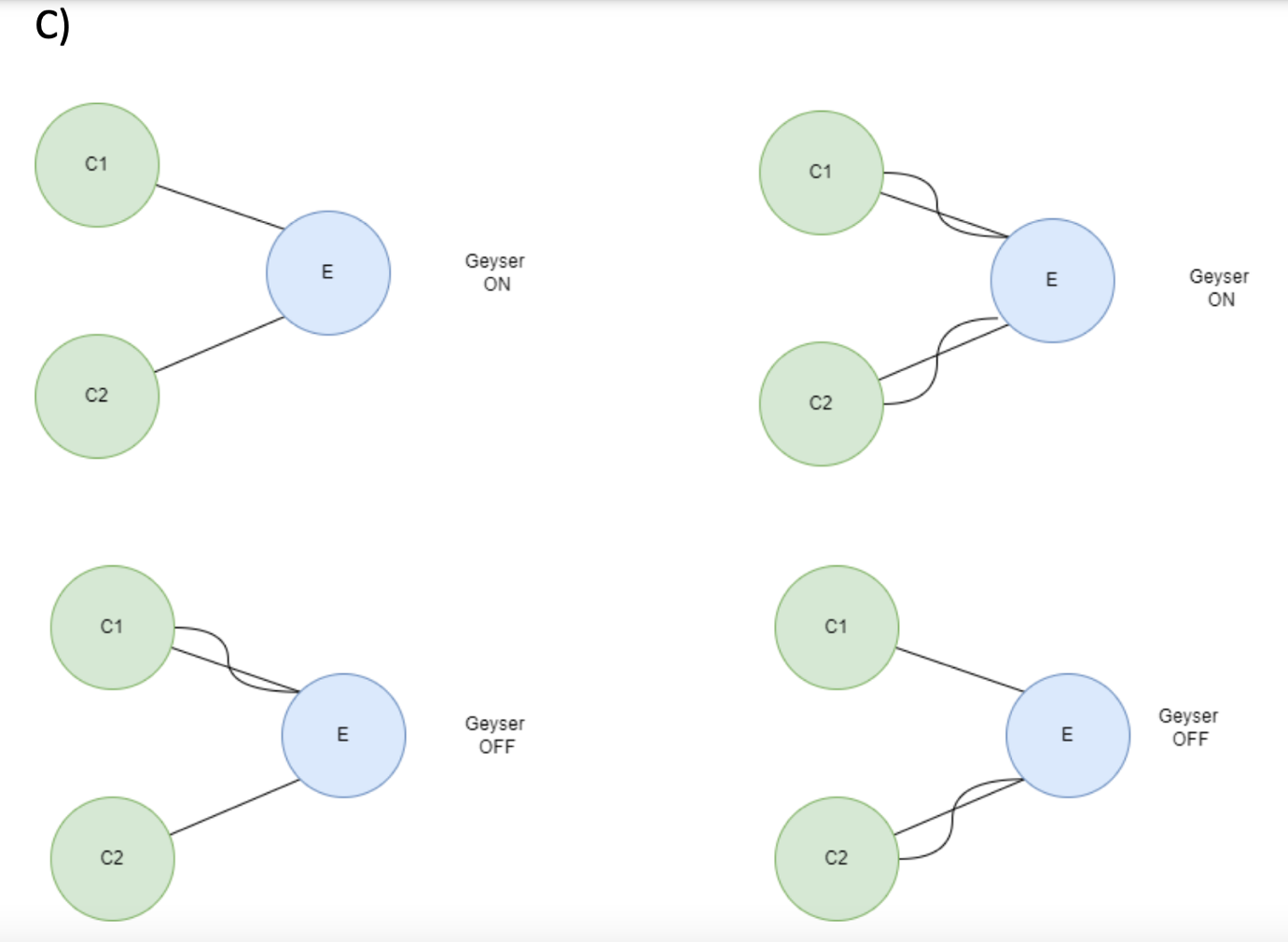
**B) TC-Test case**

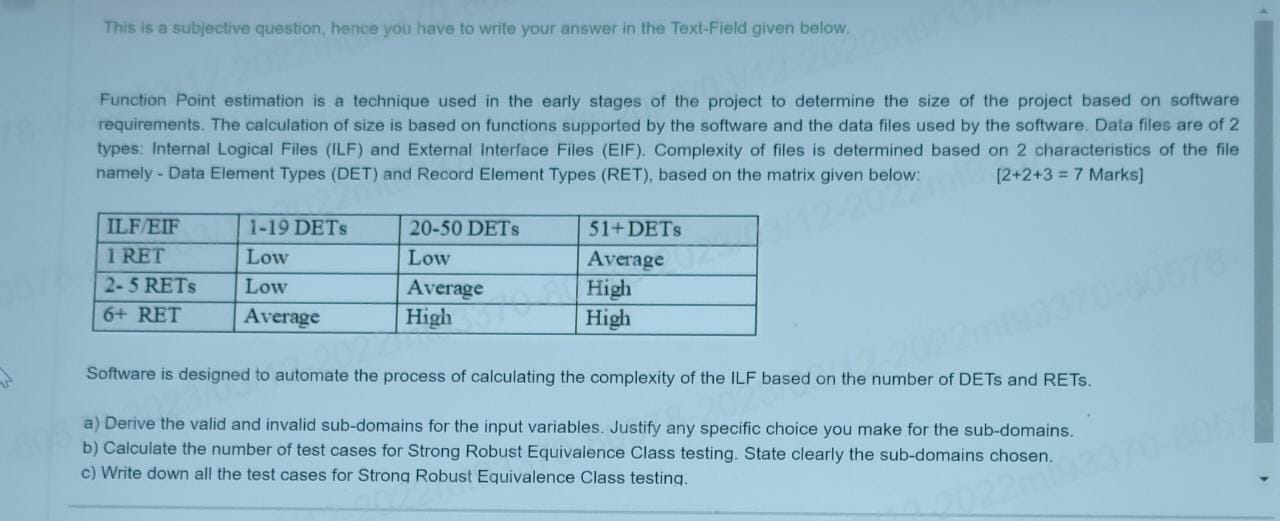
TC 1: S1 – on and S2 – off, then Geyser is OFF

TC 2: S1 – off and S2 – on, then Geyser is OFF

TC 3: S1 – on and S2 – on, then Geyser is ON

TC 4: S1 – off and S2 – off, then Geyser is ON



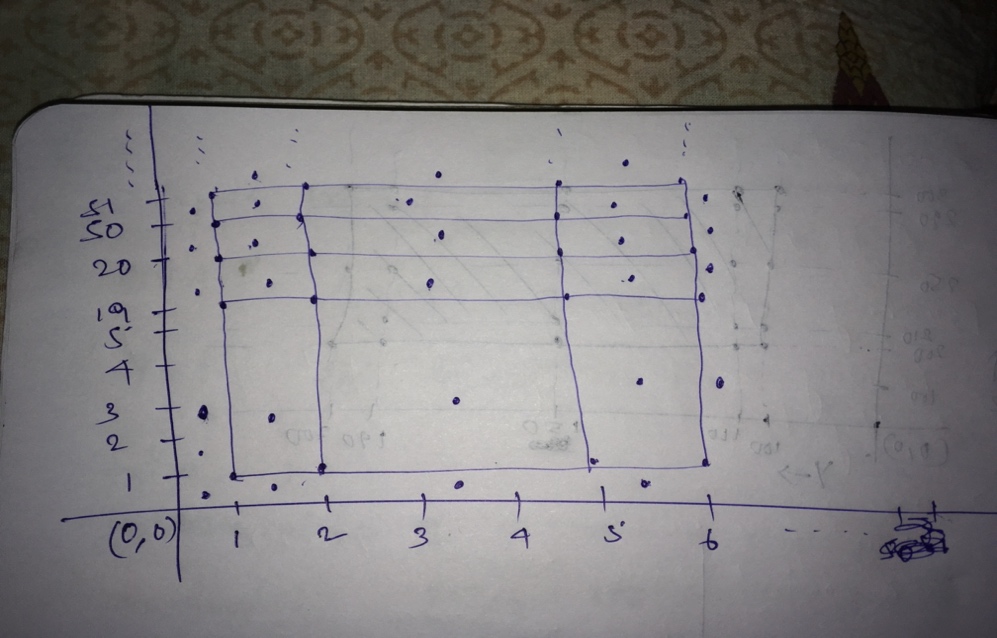
Q4.

1. **Valid subdomains:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no | Input parameters | Valid subdomains | Invalid Subdomains |
| 1 | RET | 1=RET | RET<1 |
|  |  | 2<=RET<=5 | 1<RET<2 |
|  |  | 6<=RET | 5<RET<6 |
| 2 | DET | 1<=DET<=19 | DET<1 |
|  |  | 20<=DET<=50 | 19<DET<20 |
|  |  | 51<=DET | 50<DET<51 |

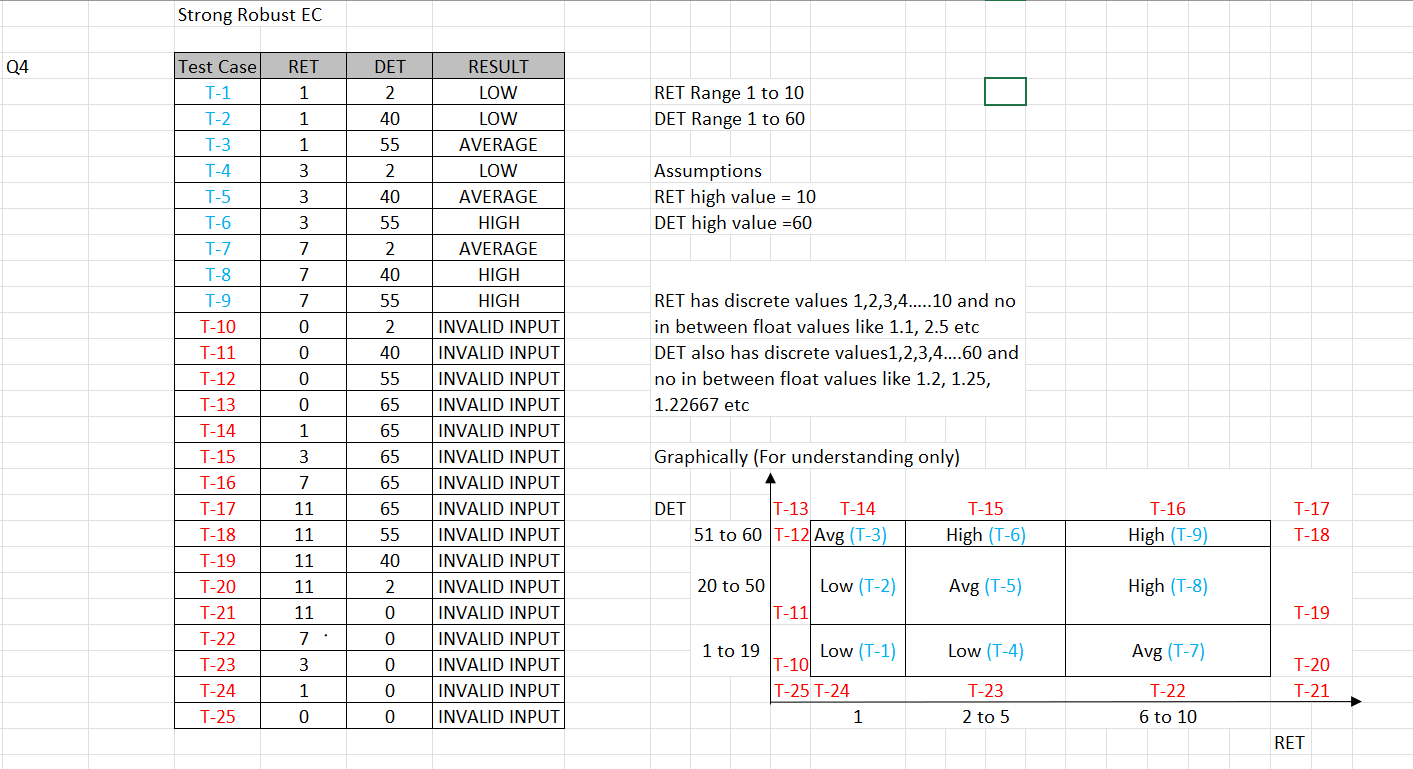
1. **Number of test cases for Strong robust:**

27

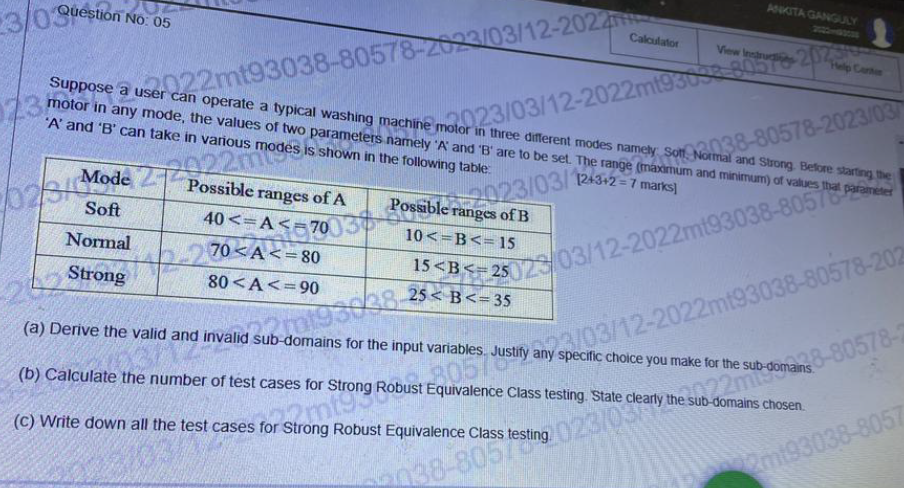


1. **Given my assumption i.e. considering discrete values of RET and DET.**

if values are continuous then we will have in between values also like between 1 and 2, 5 and 6 for RET and between 19 and 20, 50 and 51 for DET. these will be invalid input EC and will increase test cases by 14



Q. 05



Ans:

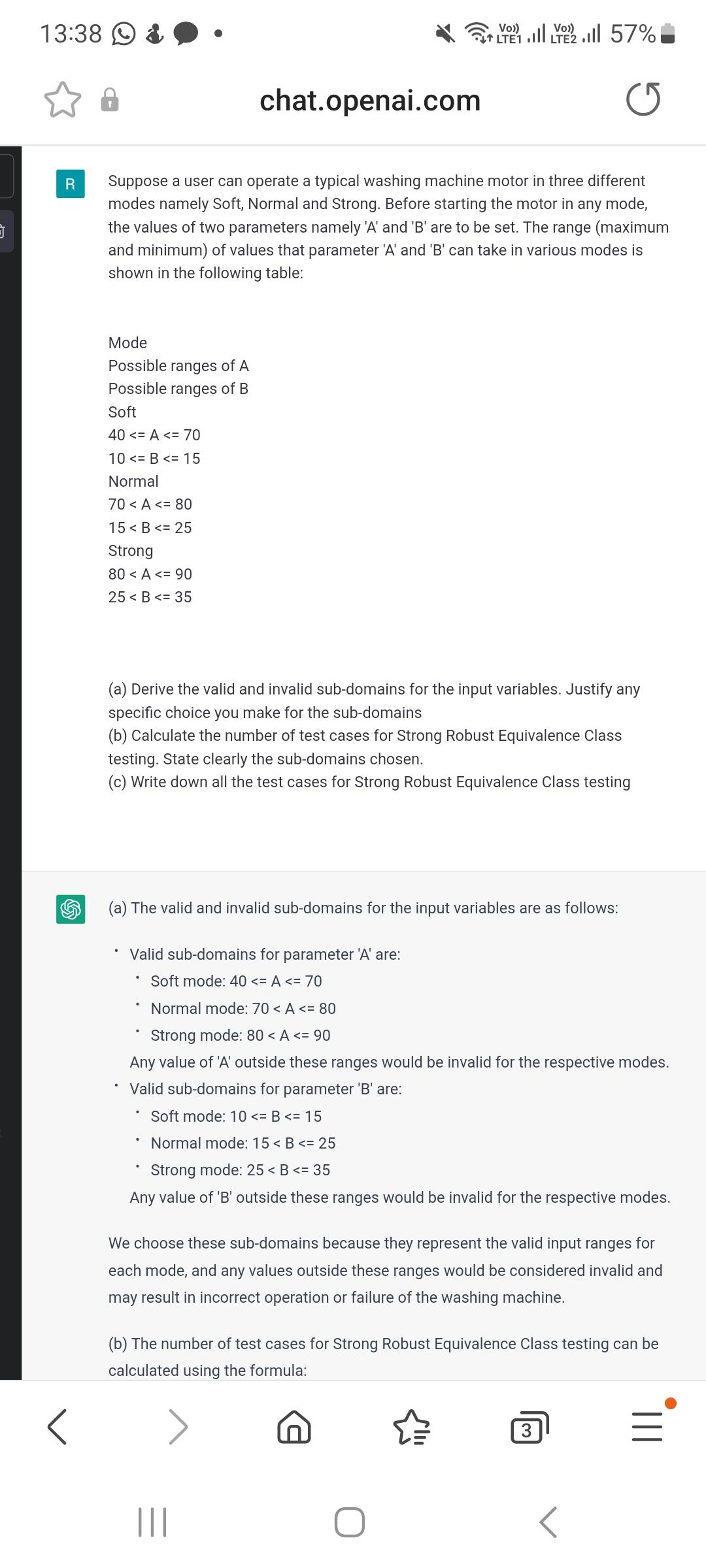
Number of test cases of Strong robust equivalence class testing as below.

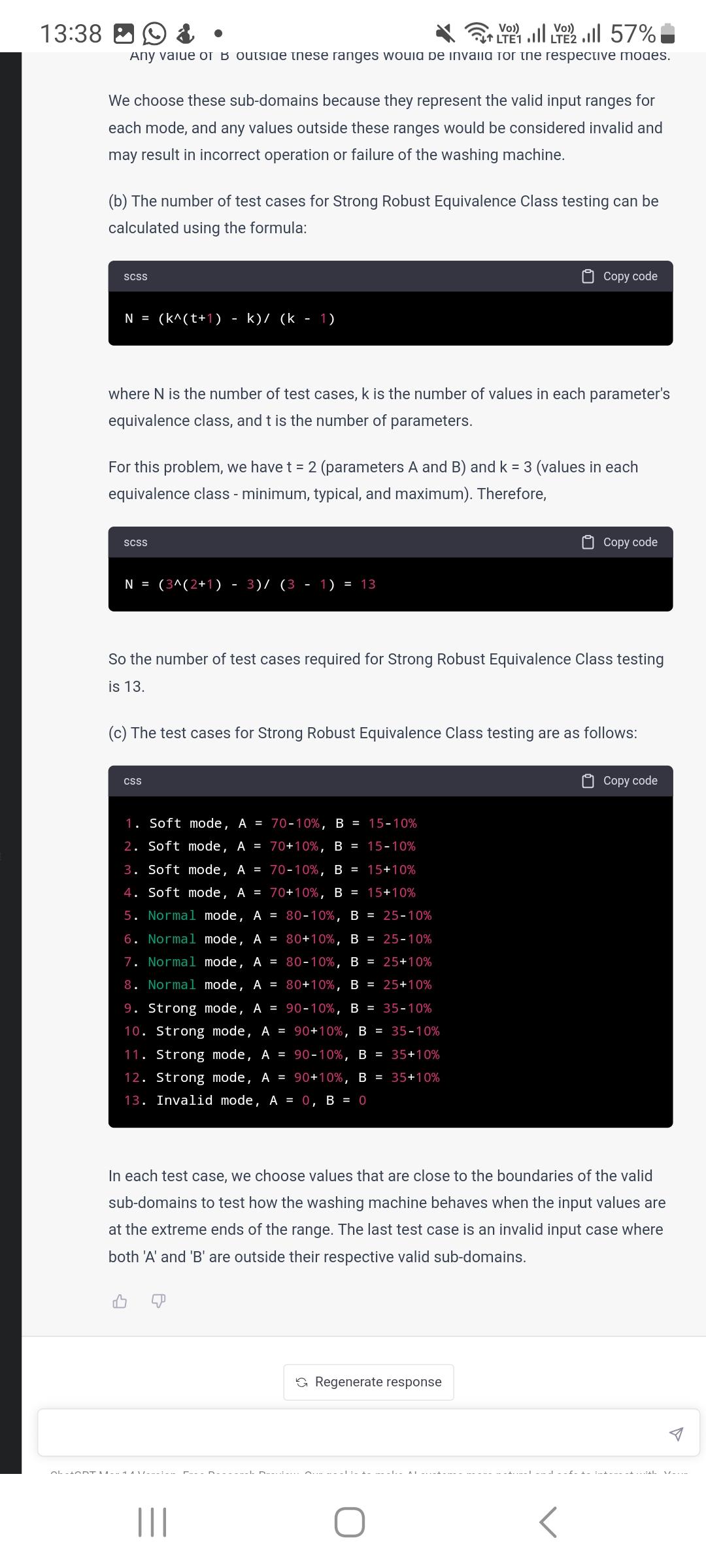
**N = (k^ (t+1) - k)/ (k-1)**

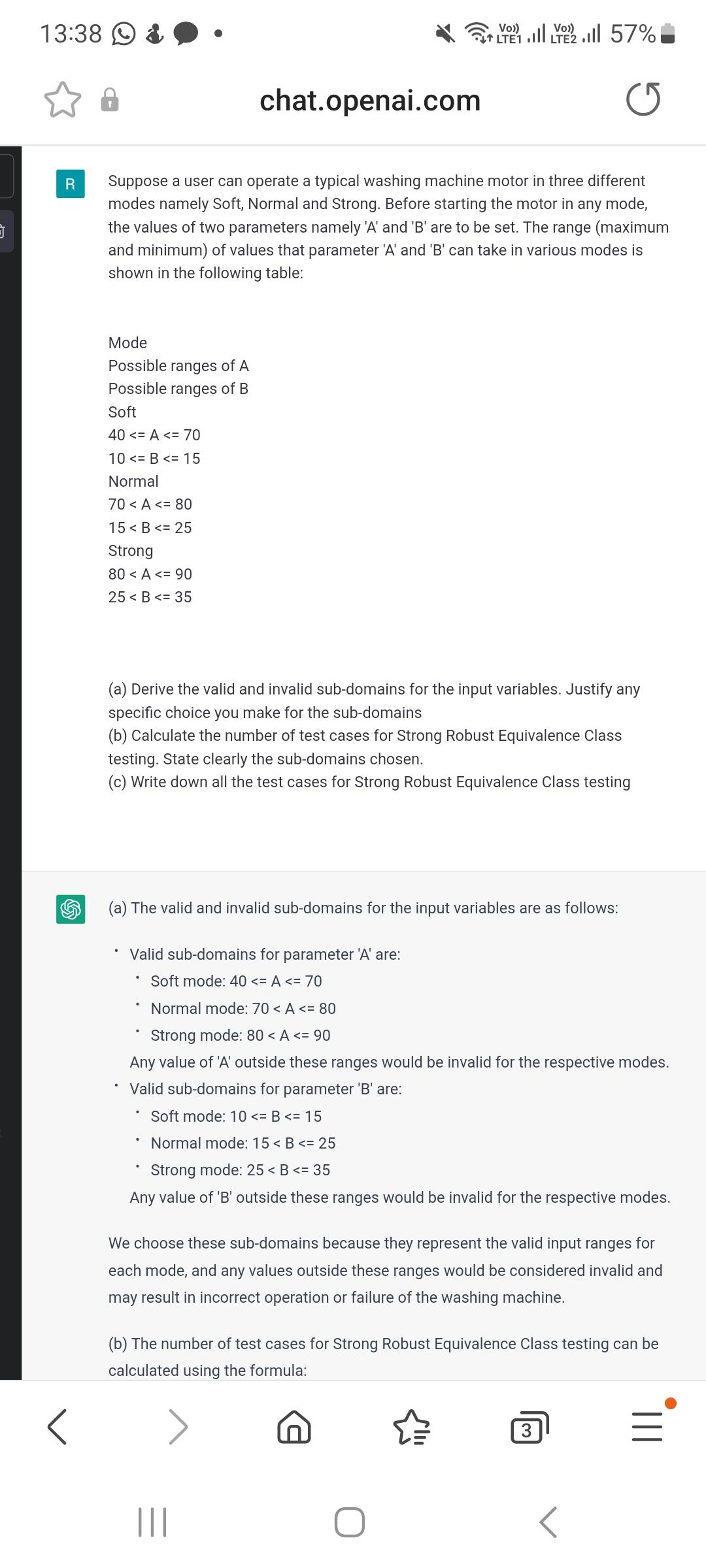
N- number of test cases

k-number of values in each parameter in equivalence class

t- number of parameters







<https://www.softwaretestinggenius.com/boundary-value-analysis-bva-a-black-box-testing-technique/#:~:text=If%20we%20have%20a%20function,4n%20%2B%201)%20test%20cases>.

<https://t4tutorials.com/what-is-robust-case-testing-software-testing/>

<https://t4tutorials.com/what-is-simple-boundary-value-testing-on-3-variables-software-testing/>

<https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html>

<https://www.geeksforgeeks.org/boundary-value-test-cases-robust-cases-and-worst-case-test-cases/>

Q.2 Draw state diagram to find out 110101 binary sequence for the binary input. List the output sequence for the given input Din = 110101110101 1110101 11010101101011010101

Ans:

Here's a state diagram to find the sequence "110101" in the binary input:

┌─────┐ 0 ┌─────┐ 1 ┌─────┐ 0 ┌─────┐ 1 ┌─────┐ 0 ┌─────┐

───► │ S0 │───────►│ S1 │───────►│ S2 │───────►│ S3 │───────►│ S4 │───────►│ S5 │

└─────┘ └─────┘ └─────┘ └─────┘ └─────┘ └─────┘

Here's the output sequence for the given input **Din = 1101011101011110101110101011010110101**:

**Output: 110101 110101 110101 110101 110101**

The state diagram consists of six states, labeled S0 to S5. Each state represents a partial match of the desired sequence "110101".

* Starting from the initial state S0, if the input is 0, we transition to state S1.
* From S1, if the next input is 1, we transition to S2.
* From S2, if the next input is 0, we transition to S3.
* From S3, if the next input is 1, we transition to S4.
* From S4, if the next input is 0, we transition to S5.
* From S5, if the next input is 1, we stay in S5 and output the desired sequence "110101".

If at any point the input deviates from the expected sequence, we return to the initial state S0.

The output sequence is shown above the state diagram, indicating the occurrences of the desired sequence "110101" in the input sequence.

**Q3. Explain at least 5 important factors to be considered while going for test automation. List at least 5 prominent metrics used for measuring automation success.**

**Ans:**

* Test Planning
* Test case generation
* Test environment development
* Test Execution
* Test results evaluation
* Problem reporting/Test log
* Defect tracking

**Factors to be considered while going for test automation:**

* **Purpose and scope:** Before starting test automation, it is essential to understand the purpose and scope of the testing. The team should determine what areas of the application need to be tested, what types of tests need to be executed, and what the expected outcomes of the tests should be.
* **Test coverage:** The team should ensure that the test suite covers all the important features and functionalities of the application. The test cases should be comprehensive, cover all possible scenarios, and be designed to find defects and bugs
* **Test data management:** The test data should be managed properly to ensure that the automation tests are consistent and reliable. The test data should be valid, up-to-date, and accurate.
* **Test Suitability:** Assess the suitability of the tests for automation. Not all tests are ideal candidates for automation. Tests that are repetitive, stable, and have a high probability of regression are generally good candidates for automation.
* **ROI Analysis:** Evaluate the Return on Investment (ROI) for test automation. Consider the effort required to automate tests, the cost of automation tools, and the expected benefits such as increased test coverage, faster test execution, and improved software quality.
* **Test Environment:** Ensure that the test environment is suitable for automation. The test environment should be stable, reliable, and capable of supporting automation tools and frameworks. It should also closely resemble the production environment to ensure accurate test results.
* **Test Maintenance:** Consider the effort required for test maintenance. Automated tests need to be regularly updated and maintained as the application undergoes changes. Analyze the frequency of changes and the impact on existing tests to determine the feasibility of automation.
* **Skill and Knowledge:** Assess the skills and knowledge required for test automation. Ensure that the team possesses the necessary technical skills to design, develop, and maintain automated tests. Training and upskilling may be required to bridge any gaps in knowledge.

**Prominent metrics for measuring automation success:**

1. **Test Coverage:** Measure the percentage of test coverage achieved through automation. It indicates how much of the application is tested automatically and provides insights into the effectiveness of automation efforts.
2. **Test Execution Time:** Measure the time taken to execute automated tests compared to manual tests. Automation should significantly reduce the time required to execute tests, enabling faster feedback and quicker releases.
3. **Defect Detection:** Track the number of defects caught by automated tests. A higher number of defects detected by automation indicates its effectiveness in identifying issues and preventing them from reaching the production environment.
4. **Test Stability:** Measure the stability of automated tests by tracking the number of false positives and false negatives. A stable test suite ensures reliable and accurate results, reducing false alarms and wasted effort.
5. **Return on Investment (ROI):** Evaluate the ROI of test automation by measuring the cost savings achieved through reduced manual testing efforts, increased efficiency, and improved software quality. It helps justify the investment in automation and assess its overall success.

These factors and metrics provide valuable insights into the effectiveness, efficiency, and value of test automation efforts, helping teams make informed decisions and continuously improve their automation practices.

**Q3. Take an example of any process that you are aware and list the various failure modes in that particular process. Conduct FMEA (Failure modes and effects analysis) and calculate RPN (Risk priority Number) for each identified failure modes. Also explain how these failure modes can be avoided which helps in designing the process well**

**Ans:**

Let's consider the process of online order fulfillment for an e-commerce platform as an example. In this process, customers place orders online, and the company fulfills and delivers those orders. Here are some failure modes that can occur in this process:

1. **Failure Mode:** Incorrect order picking

Potential Effects: Customer receives the wrong product, leading to customer dissatisfaction and potential returns.

Severity: 8 (on a scale of 1-10, where 10 represents the highest severity) Occurrence: 4 (on a scale of 1-10, where 10 represents the highest occurrence) Detection: 6 (on a scale of 1-10, where 10 represents the highest detection) RPN (Risk Priority Number): 8 x 4 x 6 = 192

1. **Failure Mode:** Shipping delay

Potential Effects: Customer receives the order later than expected, leading to dissatisfaction and potential negative reviews.

Severity: 7

Occurrence: 6

Detection: 5

RPN: 7 x 6 x 5 = 210

1. **Failure Mode:** Damaged package during transit

Potential Effects: Customer receives a damaged product, leading to customer dissatisfaction and potential returns.

Severity: 9

Occurrence: 3

Detection: 7 RPN: 9 x 3 x 7 = 189

To avoid these failure modes and design the process effectively, the following steps can be taken:

1. **Implement barcode or RFID scanning:** Use automated scanning technology during order picking to ensure accurate product identification and reduce the risk of incorrect order picking.
2. **Implement order verification checkpoints:** Introduce multiple verification checkpoints throughout the process, such as order confirmation, picking confirmation, and packing confirmation, to minimize the occurrence of shipping delays and ensure the correct products are being shipped.
3. **Enhance packaging quality:** Use sturdy packaging materials and proper cushioning techniques to protect products during transit and minimize the risk of package damage.
4. **Implement quality control checks:** Conduct regular quality control checks on the packaging process to ensure that products are being packaged properly and minimize the risk of damaged packages.
5. **Implement robust logistics and tracking systems:** Use advanced logistics and tracking systems to monitor shipments, identify potential delays, and proactively address any issues to minimize shipping delays.

By implementing these measures, the occurrence and impact of failure modes can be significantly reduced, improving the overall efficiency and customer satisfaction in the online order fulfillment process.

Top of Form

Bottom of Form

1. **Assume that you are going to start a company which is going to design and produce smart phones to capture the untapped market and to grab the market share.**
2. **List and explain the importance of various Components of House of Quality? (2.5 marks)**
3. **Using QFD (Quality Function Deployment) tool, capture all the functional and technical requirements and prioritize them. (2.5 marks)**
4. **Explain how you will tackle the competing market using QFD and explain how your product differentiates it from the existing competitors. (2.5 marks)**
5. **Explain the relation between the different technical characteristics? Explain how these technical characteristics helps in building the product features as per the customers requirement? (2.5 marks)**

**Ans:**   
a) The Components of House of Quality (HOQ) in Quality Function Deployment (QFD) are:

1. **Customer Requirements:** This component captures the needs and expectations of the customers. It includes gathering data from market research, surveys, and feedback to understand what customers want from the product.
2. **Technical Requirements:** These are the engineering specifications and technical features that the product should possess to meet the customer requirements. They are derived from the customer requirements and serve as the basis for product design and development.
3. **Importance Rating:** This component assigns a relative importance or weightage to each customer requirement. It helps prioritize the requirements based on their significance to the customer and the business goals.
4. **Competitive Assessment:** This component involves evaluating the performance of existing competitors' products in relation to the customer requirements. It helps identify areas where the company can differentiate its product and gain a competitive advantage.
5. **Relationship Matrix:** The relationship matrix establishes the correlation between the customer requirements and the technical requirements. It helps identify which technical requirements are critical in satisfying the customer needs and which ones are not as crucial.
6. **Target Values:** This component specifies the desired values or targets for each technical requirement. It sets the benchmarks for product development and guides the design and engineering teams.
7. **Design Parameters:** These are the specific parameters or variables that influence the technical requirements. They are controlled and manipulated during the product development process to achieve the desired outcomes.

The importance of these components lies in their ability to align customer needs with the technical capabilities of the product. They facilitate the development of a customer-focused product that meets or exceeds customer expectations.

b) Using QFD, functional and technical requirements can be captured and prioritized through the following steps:

1. **Identify Customer Requirements:** Gather customer feedback, conduct market research, and define the desired functionalities and features for the smartphone.
2. **Translate Customer Requirements into Technical Requirements:** Determine the engineering specifications and technical features needed to fulfill the identified customer requirements. These technical requirements should be measurable and achievable.
3. **Prioritize Technical Requirements:** Assign importance ratings to each technical requirement based on customer needs and business objectives. This can be done using methods such as Analytical Hierarchy Process (AHP) or Relative Importance Weighting (RIW).
4. **Develop Design Parameters:** Identify the design parameters that will influence the achievement of the technical requirements. These parameters can include factors such as battery life, camera resolution, processing power, and screen size.
5. **Establish Target Values:** Set target values for each technical requirement to guide the product development process. These targets should be realistic, attainable, and aligned with customer expectations.

c) To tackle the competing market using QFD, it is important to identify and prioritize customer requirements that differentiate the product from existing competitors. This can be done by:

1. **Conducting a Competitive Assessment:** Evaluate the features and functionalities of competitors' smartphones and compare them with the identified customer requirements. Identify areas where competitors fall short or where the product can outperform the competition.
2. **Assigning High Importance Ratings:** Give higher importance ratings to the customer requirements that differentiate the product from competitors. This ensures that the product development process focuses on delivering features that set it apart from existing offerings.
3. **Developing Unique Selling Points (USPs):** Use the prioritized customer requirements and technical specifications to create USPs that highlight the product's distinctive features and advantages over competitors. This can be communicated through marketing and promotional materials.
4. **Continuous Improvement:** Regularly monitor the market and gather customer feedback to identify evolving customer needs and preferences. Use this information to refine the product and stay ahead of the competition.

By leveraging QFD, the company can create a product that not only meets customer expectations but also offers unique features and benefits that differentiate it from competitors. This can help the company gain a competitive edge and capture a significant market

**Q5. Password field accepts minimum 6 characters and maximum 10 characters. It also accepts numbers from 0 to 9. It doesn’t accept any special characters except “@” character. Design at least 10 different and effective test cases to validate the password field using Boundary Value Analysis and Equivalence Partitioning Testing techniques. Ensure that there are no duplicate scenarios while designing test cases.**

Ans:

Test Cases for Password Field Validation:

**Boundary Value Analysis:**

1. Test Case 1: Minimum length boundary - Password with 6 characters (e.g., "abc123").
2. Test Case 2: Maximum length boundary - Password with 10 characters (e.g., "abcdefghij").
3. Test Case 3: Password length below the minimum limit - Password with 5 characters (e.g., "abc12").
4. Test Case 4: Password length above the maximum limit - Password with 11 characters (e.g., "abcdefghijk").
5. Test Case 5: Password with only numeric characters at the minimum length (e.g., "123456").
6. Test Case 6: Password with only numeric characters at the maximum length (e.g., "0123456789").
7. Test Case 7: Password with only special character "@" at the minimum length (e.g., "@@@@@@").
8. Test Case 8: Password with only special character "@" at the maximum length (e.g., "@@@@@@@@@@").
9. Test Case 9: Password with a combination of numeric characters and "@" at the minimum length (e.g., "1@2@3").
10. Test Case 10: Password with a combination of numeric characters and "@" at the maximum length (e.g., "0@1@2@3@4@5@6@7@8@9").

Equivalence Partitioning Testing:

1. Test Case 1: Valid password with alphanumeric characters within the length limit (e.g., "abc123").
2. Test Case 2: Valid password with alphanumeric characters at the minimum length (e.g., "abc12").
3. Test Case 3: Valid password with alphanumeric characters at the maximum length (e.g., "abcdefghij").
4. Test Case 4: Valid password with special character "@" within the length limit (e.g., "abc@123").
5. Test Case 5: Valid password with special character "@" at the minimum length (e.g., "@@123").
6. Test Case 6: Valid password with special character "@" at the maximum length (e.g., "abcd@efghi").
7. Test Case 7: Invalid password with alphanumeric characters below the length limit (e.g., "ab").
8. Test Case 8: Invalid password with alphanumeric characters above the length limit (e.g., "abcdefghijk").
9. Test Case 9: Invalid password with special characters other than "@" (e.g., "abc\*123").
10. Test Case 10: Invalid password with a combination of special characters and alphanumeric characters (e.g., "abc@\*123").

These test cases cover different scenarios based on boundary values and equivalence partitions, ensuring effective validation of the password field according to the specified requirements.

Top of Form

Bottom of Form