LAB3

Search for a module (function) to solve a certain problem (choose the language you like: C or JAJA). You indicate the path to the code page that you refer to (if any)

Instructions:
-You need to know at least one programming language to perform reasonably well on certain exam questions.
-Students are not allowed to use IDEs such as NetBeans, IntelliJ IDEA, or similar software to do the test.

1) Find issues in the code (i.e., coding practice, compile errors, potential logical issues, etc (Use template 1)

2) Assuming you are assigned to conduct the component test for the code mentioned in question 1 (or another piece of code), please design and create the minimum component test cases (Unit Test case) needed to achieve 100% statement coverage and 100% decision coverage. (Use template 2)

3) You are assigned to do the functional (black-box) test for a function (hoose a function and show it: image or link)

- Function Description

+ Actors:

+ Purpose

+ Business rules

- Function Details:

+ Normal case

+ Abnormal case

(1) Analyse test conditions using the Equivalence Partitioning and Boundary Value Analysis test design techniques and fill in the template 3_1.

(2) Design 10 test cases with the test data to cover as much minimal user confidence and maximum fault-finding fill in the template 3_2.

(3) Complete the 10 test cases that you have designed in (2) with decision tables using template 3_3.

==========The end ==========

Example for questin 1:

```
Pulic class sumEx{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 1 number: ");
        num = sc.nextInt();
        for(int i = 1; i <= num; ++i)
        {
            sum += i;
        }
        System.out.println("sum is: "+sum);
}
}
```

Example for question 2:

	Priority	Name	Initial Conditions	Steps	Expected Results
1	1	Login attempt: correct login and password	There is a registered user in the system to perform the test on.	1. Correct login and password is entered 2. The "Login" button is clicked	The user successfully logged in to the system
2	2	Login attempt: correct login, incorrect password	There is a registered user in the system to perform the test on.	1. Correct login and incorrect password is entered 2. The "Login" button is clicked	The user did not log in. A message is displayed: "Incorrect login or password"
3	2	Login attempt: incorrect login, incorrect password	-	1. Incorrect login and incorrect password is entered 2. The "Login" button is clicked	The user did not log in. A message is displayed: "Incorrect login or password"

LAB4.

Lab4 has 2 options:

1) Option 1

Take an example using your project or an existing project, you perform an Integration Test or System Test case on that project.

List at least one of the orders in which subsystems might be formed and tested

if you use top-down integration,

if you use bottom-up integration.

If you have even more time, write code for this lab4. Writing Integration Tests. We'll look at how to send requests with path variables and parameters.

===

Example an example of an integration test case for a retail website:
Test Case Title: Checkout Flow Integration Test
Objective: To test the integration of the shopping cart, payment gateway, and order management systems.
Prerequisites:

- ✓ A customer account must be created
- ✓ The customer must have at least one product in their shopping cart

Steps:

1. Log in to the retail website as a customer
2. Navigate to the shopping cart and verify that the correct products are listed
3. Select a payment method and enter payment information
4. Click the "Place Order" button
5. Verify that the payment is processed successfully
6. Verify that the order is listed in the order management system
7. Verify that the order details, such as the products, shipping address, and payment method, are correct

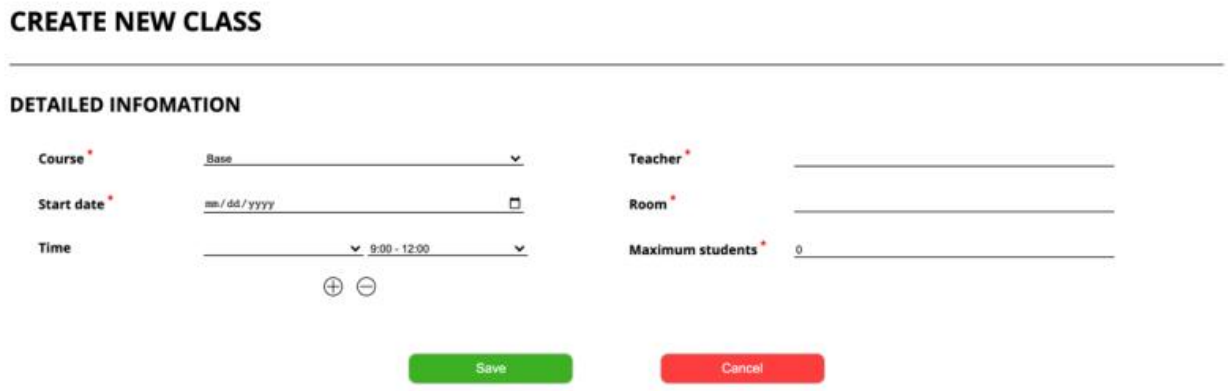
Expected Results:

- ✓ The payment should be processed successfully
- ✓ The order should be listed in the order management system
- ✓ The order details should be correct

This integration test case verifies that the different components of the retail website, such as the shopping cart, payment gateway, and order management systems, are working together to process an order.

2) Option 2

Assume you are assigned to do the Integration Test between the front-end and back-end of the Create Class function.



Function description:

- Actor/Role: Staff (Manager, Coordinator)
- Purpose: Create a new class in the center
- Function Interface:

  - Six text fields to insert information about class
  - Two buttons (Save, Cancel)

Function Detail:

- The actor selects the course and then fills in information about the class.
- The system will verify all data that the actor fills in. If one of the data is not correct, the system will
- Show notification and change color to red, which field is not correct.
- If the actor clicks back will return to the list class page, and all data in create page will not save

Assume that you are assigned to do the System Test of the "Create class schedule" feature. The business process will include the following steps

Step 1: The Admin creates a new English class.
Step 2: The Manager approves the new class schedule.
Step 3: The teacher receives a notification email about the new class.
Step 4: The teacher makes a confirmation about the assignment.
Step 5: The system shows the new class information to students.

It would be best if you designed the test cases for normal flow and alternative flow for the above feature.

1. Analyze the test condition using the Equivalence Partitioning and Boundary Value Analysis test design techniques
2. Design 10 integration test cases with the test data to cover as many Tags as possible
3. Create the high-level test case with the test data for your test case designed above.

===

Information explanation:

Depending on an input data type, we can define two types of test cases: high- and low-level.

Low-level test case (detailed)

A test case with defined input values and expected results.

Example based on an app that performs addition:

Lp.	Priority	Name	Input Data	Initial Conditions	Steps	Expected Results
1	1	Positive number addition	1,2	-	1. Number 1 is typed in the first field.	The result "3" appears on screen.
					2. Number 2 is typed in the second field.	
					The "Calculate" button is clicked.	

High-level test case (logical)

A test case without defined input values or expected results. It uses logic operators, but actual values are not defined yet. It's also known as an "abstract test case".

Example based on an app that performs addition:

Lp.	Priority	Name	Input Data	Initial Conditions	Steps	Expected Results
1	1	Positive number addition	1,2	-	1. Number A is typed in the first field.	The Sum of A and B appears on screen
					2. Number B is typed in the second field.	
					The "Calculate" button is clicked.	

High-level conditions give greater freedom, and their execution depends largely on the tester. Two different people can execute the same case and enter different data, so specific results will also differ. It could even be that one person gets an error while the other doesn't.

Low-level cases are used in automated tests, where actions are 100% repeatable. For manual tests, it's better to use high-level cases and tap into a tester's potential.

==========The End ==========