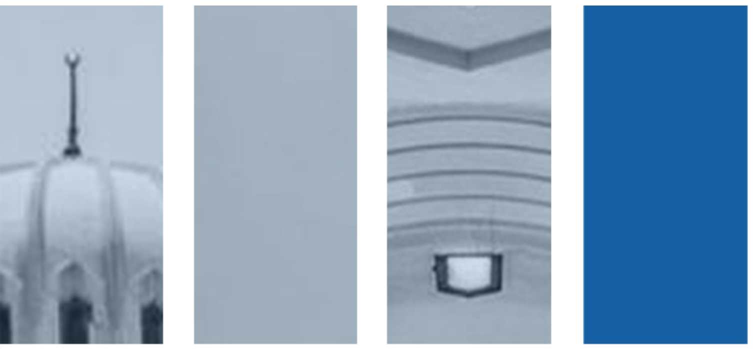


Ain Shams University

Faculty of Engineering

Computer and Systems Engineer

Project Name: ATM Machine

Course Code: CSE337s

Course Name: Software Testing

Team Members:

Ahmed Magdi Mostafa Hosni	1808714
Ahmed Magdy Fahmy Mohamed	1805862
Ahmed Abdallah Mansour Abdel Lateif	1809252
Omar Mohamed Diaaeldin Ibrahim	1802932
Ahmed Mohamed Ahmed Sayed	1804904

Assignment 2

Black box testing

➢ **Testing after fixing the code after last white box testing (One bug was still unintentionally unfixed so we left it)**

Test 1 | Per transaction

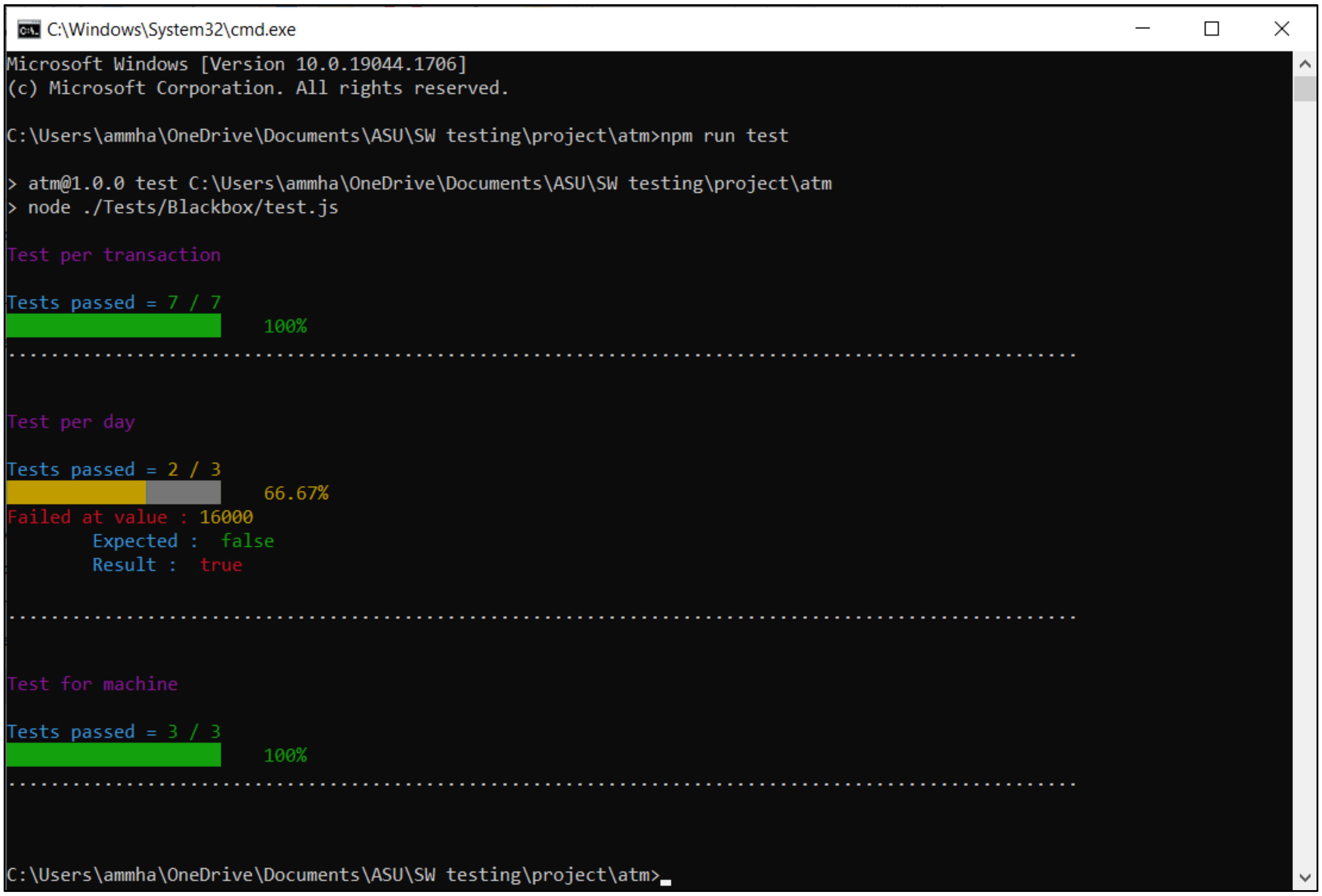
Value	**Range**	**Region name**	**Expected**	**Result**	**Status**
-50	t < 0	Region 1	False	False	Pass
45	0 < t < 50	Region 2	False	False	Pass
50	50 ≤ t ≤ 10000, t % 50	Region 3	True	True	Pass
55	50 ≤ t ≤ 10000, t !% 50	Region 3.1	False	False	Pass
5000	50 ≤ t ≤ 10000, t % 50	Region 3	True	True	Pass
10000	50 ≤ t ≤ 10000, t % 50	Region 3	True	True	Pass
12000	10000 < t	Region 4	False	False	Pass

Test 2 | Per day

Value	**Range**	**Region name**	**Expected**	**Result**	**Status**
12000	t ≤ 15000	Region 1	True	True	Pass
15000	t ≤ 15000	Region 1	True	True	Pass
16000	15000 < t	Region 2	False	True	Fail

Test 3 | For machine

Value	Range	Region name	Expected	Result	Status
4500	t ≤ 5000	Region 1	True	True	Pass
5000	t ≤ 5000	Region 1	True	True	Pass
5500	5000 < t	Region 2	False	False	Pass



```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ammha\OneDrive\Documents\ASU\SW testing\project\atm>npm run test

> atm@1.0.0 test C:\Users\ammha\OneDrive\Documents\ASU\SW testing\project\atm
> node ./Tests/Blackbox/test.js

Test per transaction

Tests passed = 7 / 7
                    100%
.....................................................................................................


Test per day

Tests passed = 2 / 3
                    66.67%
Failed at value : 16000
        Expected :  false
        Result :  true

.....................................................................................................


Test for machine

Tests passed = 3 / 3
                    100%
.....................................................................................................



C:\Users\ammha\OneDrive\Documents\ASU\SW testing\project\atm>
```

<u>Test after changing one condition from (amount % 50 != 0) to (amount % 50 == 0) so all values divisible by 50 will fail which is the opposite of expected</u>

Test 1 | Per transaction

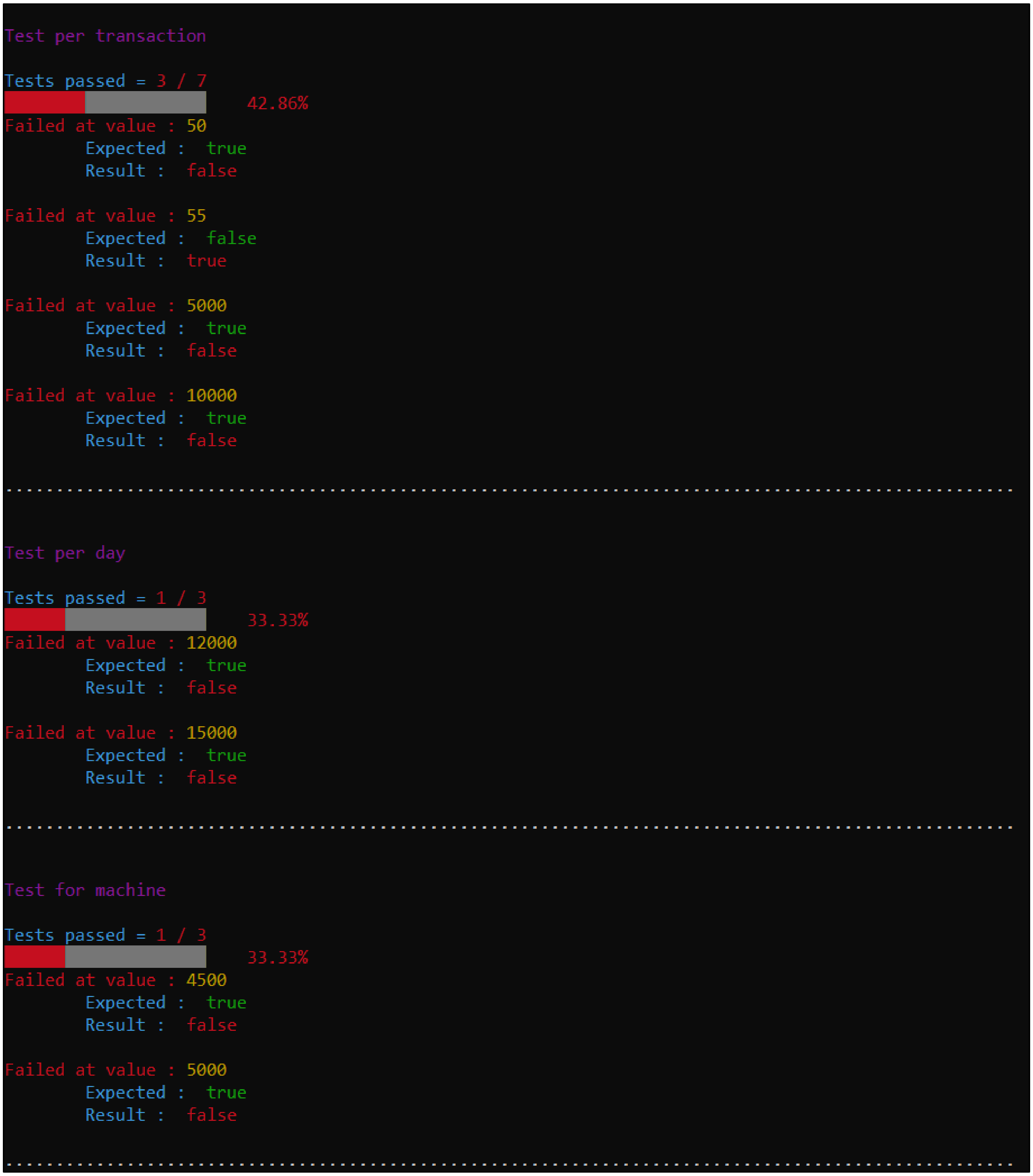
Value	Range	Region name	Expected	Result	Status
-50	t < 0	Region 1	False	False	Pass
45	0 < t < 50	Region 2	False	False	Pass
50	50 ≤ t ≤ 10000, t % 50	Region 3	True	False	Fail
55	50 ≤ t ≤ 10000, t !% 50	Region 3.1	False	True	Fail
5000	50 ≤ t ≤ 10000, t % 50	Region 3	True	False	Fail
10000	50 ≤ t ≤ 10000, t % 50	Region 3	True	False	Fail
12000	10000 < t	Region 4	False	False	Pass

Test 2 | Per day

Value	Range	Region name	Expected	Result	Status
12000	t ≤ 15000	Region 1	True	False	Fail
15000	t ≤ 15000	Region 1	True	False	Fail
16000	15000 < t	Region 2	False	False	Pass

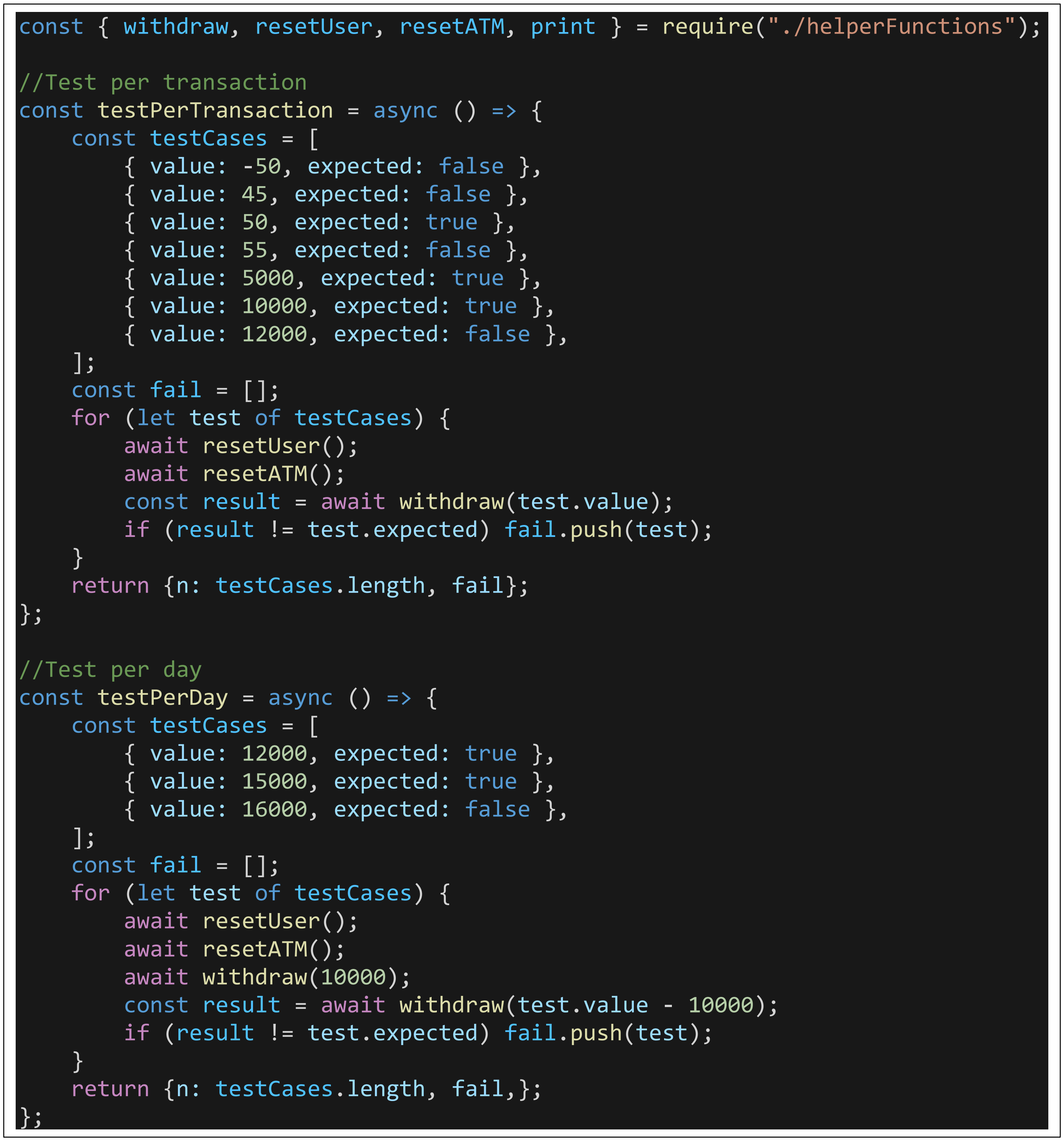
Test 3 | For machine

Value	Range	Region name	Expected	Result	Status
4500	t ≤ 5000	Region 1	True	False	Fail
5000	t ≤ 5000	Region 1	True	False	Fail
5500	5000 < t	Region 2	False	False	Pass



```
Test per transaction

Tests passed = 3 / 7
                        42.86%
Failed at value : 50
        Expected :  true
        Result :  false

Failed at value : 55
        Expected :  false
        Result :  true

Failed at value : 5000
        Expected :  true
        Result :  false

Failed at value : 10000
        Expected :  true
        Result :  false

-----------------------------------------------------------------------------------------------------


Test per day

Tests passed = 1 / 3
                        33.33%
Failed at value : 12000
        Expected :  true
        Result :  false

Failed at value : 15000
        Expected :  true
        Result :  false

-----------------------------------------------------------------------------------------------------


Test for machine

Tests passed = 1 / 3
                        33.33%
Failed at value : 4500
        Expected :  true
        Result :  false

Failed at value : 5000
        Expected :  true
        Result :  false

-----------------------------------------------------------------------------------------------------
```

➢ Testing Code



```javascript
const { withdraw, resetUser, resetATM, print } = require("./helperFunctions");

//Test per transaction
const testPerTransaction = async () => {
    const testCases = [
        { value: -50, expected: false },
        { value: 45, expected: false },
        { value: 50, expected: true },
        { value: 55, expected: false },
        { value: 5000, expected: true },
        { value: 10000, expected: true },
        { value: 12000, expected: false },
    ];
    const fail = [];
    for (let test of testCases) {
        await resetUser();
        await resetATM();
        const result = await withdraw(test.value);
        if (result != test.expected) fail.push(test);
    }
    return {n: testCases.length, fail};
};

//Test per day
const testPerDay = async () => {
    const testCases = [
        { value: 12000, expected: true },
        { value: 15000, expected: true },
        { value: 16000, expected: false },
    ];
    const fail = [];
    for (let test of testCases) {
        await resetUser();
        await resetATM();
        await withdraw(10000);
        const result = await withdraw(test.value - 10000);
        if (result != test.expected) fail.push(test);
    }
    return {n: testCases.length, fail,};
};
```

```
//Test for machine
const testForMachine = async () => {
    const testCases = [
        { value: 4500, expected: true },
        { value: 5000, expected: true },
        { value: 5500, expected: false },
    ];
    const fail = [];
    for (let test of testCases) {
        await resetUser();
        await resetATM(5000);
        const result = await withdraw(test.value);
        if (result != test.expected) fail.push(test);
    }
    return {
        n: testCases.length,
        fail,
    };
};

const runTest = async () => {
    let result;
    //Test per transaction
    result = await testPerTransaction();
    print("Test per transaction", result);

    //Test per day
    result = await testPerDay();
    print("Test per day", result);

    //Test for machine
    result = await testForMachine();
    print("Test for machine", result);
};

runTest();
```