

Instruction สำหรับ Assignment 1

ส่วนที่ 1: เขียนแสดงวิธีการคิด test cases ให้ได้จำนวน Independent Paths ตามค่า Cyclomatic Complexity (45%)

คำถามที่ 1: จงวาด Flow Graph จากโค้ดฟังก์ชัน average (15%) โดยให้มีรายละเอียดครบถ้วนดังนี้

- โหนดใน Flow Graph ให้ Label โหนดด้วยหมายเลขบรรทัดของโค้ดตามรูป 1 หมายเลขบรรทัดของโค้ดฟังก์ชัน average
- ด้านข้างของ Node ให้ใส่ Statement อย่างย่อ เช่น Assignment , เงื่อนไข while เงื่อนไข if เป็นต้น ส่วน Edge ที่ออกจาก Predicate Node ให้เขียนระบุ Boolean Expression เพื่อให้ TA สามารถตรวจและอ่านทำความเข้าใจ Graph ได้ง่าย
- Statement ประเภท If หรือ While ที่เป็น Predicate Node หากประกอบด้วย Boolean Condition ย่อยมากกว่า 1 เงื่อนไข ให้แตก 1 Condition ย่อยเป็น Predicate Node เช่น บรรทัดที่ 8 เป็น While ที่มี 2 เงื่อนไขย่อย ให้แตกเป็นโหนด 8a สำหรับ Conditionย่อยที่ 1 ของ บรรทัดที่ 8 และโหนด 8b สำหรับ conditionย่อยที่ 2 ของบรรทัดที่ 8

คำถามที่ 2: จงคำนวณหาจำนวน Independent Paths จาก Graph โดยใช้สูตรคำนวณค่า Cyclomatic Complexity ทั้งสามสูตรต่อไปนี้ (10 %)

= #Regions หรือ (จำนวนพื้นที่)

= #Predicate nodes + 1 (จำนวน Predicate Nodes + 1)

= #Edge - #Node + 2 (จำนวนลูกศร - จำนวนโหนด + 2)

หมายเหตุ ในการตอบให้แสดงวิธีการคำนวณค่า Cyclomatic Complexity จากแต่ละสูตร ค่าที่ได้จากสามสูตรควรเท่ากัน ไม่แสดงเฉพาะค่าตัวเลขเท่านั้น

คำถามที่ 3-4: จงแสดง Basis Paths พร้อมนำไปสร้างเป็น Test cases และ กำหนดเป็น Test case Specification ที่สมบูรณ์ (20%)

- ระบุแต่ละ Independent Path โดยระบุ Node ของแต่ละ Path ตั้งแต่โหนดเริ่มต้นจนถึงโหนดสิ้นสุด ยกตัวอย่างเช่น Path1 : 1-2-10-11 เป็นต้นสมมติให้ 1 เป็นโหนดเริ่มต้น และ 11 เป็นโหนดสิ้นสุด
- ระบุ Test Cases ที่อนุมานได้จาก Independent Paths ทั้งหมด โดยให้อธิบายว่าต้องใช้ Test Case กี่กรณี , Test Case แต่ละกรณีต้อง ใช้ Input เป็นค่าใด และ Test Case แต่ละกรณีเป็นการทดสอบ Basis Path ใด เป้าหมายของแต่ละ Test case คืออะไร

ให้ระบุ Test case Specification โดยใช้รูปแบบตามตัวอย่างต่อไปนี้

Basis Path	Test case ID	Test Objective	Test Data	Expected Output	Actual Output	Test Result (Passed หรือ Failed)
ระบุ Path จาก Node เริ่มต้นไปสิ้นสุดที่ Node สิ้นสุด	TC1	คำอธิบายวัตถุประสงค์การทดสอบกรณีนี้	ข้อมูลอินพุตที่ต้องกำหนดค่าเพื่อทดสอบ	ผลลัพธ์ที่ควรจะเป็น	ผลลัพธ์จากการรัน Test Case	
1-4-5-8-9-2-10-11-13-14	TC2	ทดสอบหากค่าเฉลี่ยจากชุดตัวเลข 1 จำนวน โดยค่าในชุดตัวเลขใดเงื่อนไขคือ ค่าไม่ได้อยู่ใน range มากกว่าหรือเท่ากับ min และน้อยกว่าหรือเท่ากับ max	value[1,-999], min=10, max=20	average =-999, totalInput=1, totalValid=0		

คำถามที่ 5 : จงนำ Test cases มา implement automated test code และ รันผลแสดงผลการทดสอบและสร้าง Test Coverage Report โดยมีข้อกำหนดในการสร้าง Test Project ดังนี้

- Test Project สำหรับทดสอบ Automated Test ให้มีเฉพาะโฟลเดอร์ src, test และไฟล์ package.json เท่านั้น
- ภายในโฟลเดอร์ node_modules ไม่ต้องมีไฟล์ใดๆ เพื่อประหยัดพื้นที่ในการจัดเก็บ Assignment ของนักศึกษาบน Moodle Server
- โฟลเดอร์ src ให้มีเฉพาะไฟล์ average.js เท่านั้น
- โฟลเดอร์ test ให้เก็บไฟล์ average.test.js ที่บรรจุ Code Automated Test ที่นักศึกษาเขียนขึ้นตาม Test cases Specification ที่คิดในส่วนที่ 1

ส่วนที่ 2 : ผลการทดสอบ Test Project (55%)

- ถ้ารัน Test Case Project ไม่ได้จะไม่ได้คะแนนส่วนที่ 2 (0 คะแนน) วิธีการที่ TA จะรันทดสอบ Test Project เพื่อตรวจให้คะแนน Assignment คือ จะรันคำสั่งต่อไปนี้ npm install ตามด้วย npm test และ npm run coverage
- จากนั้นจะเช็คผลการทดสอบ Automated Test จาก Console และ Web Report หากไม่สามารถรัน Project ได้ ถือว่าคะแนนส่วนการทดสอบ Test Project เป็น 0
- ความสมบูรณ์ของโค้ด Automated test ที่เขียนจาก Test Cases Specification ที่ระบุในส่วนที่ 1 ครบถ้วนทุกกรณี มีการกำหนดคำอธิบายของ Test cases แต่ละกรณีให้เข้าใจง่าย หากรัน Test ไม่ผ่าน แสดง Message อธิบายเหตุผลไว้ชัดเจน มีการแยก Test cases และ จัดกลุ่ม Test Suite (30%)
- ความครบถ้วนของ Test Coverage ระดับ Statement, Branch, Function, Lines of Code แสดงให้เห็นว่า Test cases ครอบคลุม 100% ทุกระดับ ไม่มี Code ที่ไม่ถูก covered ด้วย test cases และ Test ครบทุก Independent Paths โดยตรวจจากรายงาน Coverage ใน Folder “coverage” และ “jest-stare” ที่สร้างจากเครื่องมือ jest-stare (25%)

ข้อกำหนดการส่งไฟล์การบ้านเข้ากล่องบน Moodle

- ให้ส่งมาเป็นไฟล์นามสกุล .zip 1 ไฟล์ต่อ 1 กลุ่ม (1-2 คน) ให้ตั้งชื่อไฟล์ตามรูปแบบต่อไปนี้ คือ “Assignment1_รหัสนักศึกษาคนที่1_รหัสนักศึกษาคนที่2.zip” โดยภายในไฟล์ .zip ประกอบไปด้วย
 1. ไฟล์ Assignment1_รหัสนักศึกษาคนที่1_รหัสนักศึกษาคนที่2.pdf ที่ตอบคำถามส่วนที่ 1
 2. Node Project ที่ zip โค้ด Test
 3. ไฟล์ Readme.txt ที่ระบุชื่อนามสกุล รหัสประจำตัวนักศึกษา ของสมาชิกในทีม

```

1  var average = function (value, min, max) {
2      ... let total = new Object(), average = 0;
3      ... total.valid = 0;
4      ... total.input = 0;
5      ... //check whether value is an array
6      ... if (Array.isArray(value)) {
7          ... let i = 0, sum = 0;
8          ... while (value[i] !== -999 && total.input < 100) {
9              ... (total.input)++;
10             ... if (value[i] >= min && value[i] <= max) {
11                 ... (total.valid)++;
12                 ... sum = sum + value[i];
13             ... }
14             ... i++;
15         ... }
16         ... if (total.valid > 0) {
17             ... average = sum / total.valid;
18         ... } else {
19             ... average = -999;
20         ... }
21     ... }
22     ... return {total: total, average: average};

```

รูป 1 หมายเลขบรรทัดของโค้ดฟังก์ชัน average