

1. กำหนดฟังก์ชัน `triangleType(a,b,c)`

- a. ฟังก์ชันนี้ตรวจสอบด้าน a, b, c ว่าสามารถประกอบเป็นสามเหลี่ยมได้หรือไม่
- b. คุณสมบัติการเป็นสามเหลี่ยมคือ ด้านสองด้านใด ๆ รวมกันต้องยาวกว่าด้านที่สาม
- c. ถ้าประกอบเป็นสามเหลี่ยมได้จะตรวจสอบว่าเป็นสามเหลี่ยมประเภทใด และแสดงผลที่เป็นประเภทของสามเหลี่ยมดังนี้
 - i. “Equilateral Triangle”
 - ii. “Isosceles Triangle”
 - iii. “Scalene Triangle”
- d. ถ้าประกอบเป็นสามเหลี่ยมไม่ได้จะแสดงผล “Not a Triangle”
- e. ด้าน a, b, c ต้องเป็นตัวเลข integer ที่มีค่าตั้งแต่ 1-200 เท่านั้น ถ้าไม่เป็นตามข้อกำหนดนี้จะแสดงผล "Input sides must be integers starting from 1 to 200"

- 1.1) จงเขียน Input Equivalence Classes ของฟังก์ชันนี้
- 1.2) จงเขียน Black-box test cases ที่สร้างให้ครอบคลุม (cover) ให้ครบทุก Input Equivalence Classes ที่เป็นคำตอบในข้อ 1.1
- 1.3) จงสร้าง test cases จากข้อ 1.2 ด้วย Jest เพื่อทดสอบโปรแกรมดังกล่าวเพื่อหาให้ได้ว่า test cases นั้นพบข้อผิดพลาดใดของโปรแกรมบ้าง (ห้ามนิสิตไปเปิดดู code ของโปรแกรม Triangle.js)

ส่ง filename.test.js (ตั้งชื่อ filename ด้วย รหัสนิสิตตามด้วยA)

- 1.4) เมื่อทราบข้อผิดพลาดแล้ว จงไปแก้โปรแกรม Triangle.js ให้ถูกต้อง และ สร้าง test case ด้วย Jest เพื่อให้ได้ 100% branch coverage

ส่ง Triangle.js (version ที่แก้แล้ว) (ตั้งชื่อ filename ด้วย รหัสนิสิตตามด้วยTriangle)

และส่ง filename.test.js (ตั้งชื่อ filename ด้วย รหัสนิสิตตามด้วยB)

2. Cypress

Installing

Reference starting from [this](#). (Ctrl + Click the link)

1. Open Terminal and go into the folder you want to put Cypress test file in. (cd /somePath)
2. [If you don't have node yet (you can check by using `node --version`), please download from <https://nodejs.org/en/download>]
3. `npm install cypress --save-dev` (Nothing will appear in folder, don't worry)
4. `npx cypress open`
5. Choose E2E Testing
6. Create new empty spec

Basic Commands ([Commands](#))

`.get()` / `.click()` / `.type()` / `.should()`

Three Specs for exercise

Link for the form: <https://panaryco.wixsite.com/myhotel>

1. Case 1: Submit should success.
Input all input with corrected data, click "Submit".
Then "Thanks for submitting!" should be visible.
2. Case 2: Submit should not success.
Input all input with same data except wrong email format (ex. 123), click "Submit".
Then "Thanks for submitting!" should not be visible.
3. Case 3: First name should be equal to what we input.
There is an error in this form. I can detect it by writing a simple spec.
Input first name input with very long first name (10 characters+), then check if value in the input match the value that we typed in.
*This spec should **not pass** because the first name input is limited length.

This website example is a bit too specific (because it's Wix) but the main thing is how access and verify components to test the system.