



ฝ่ายพัฒนาระบบสารสนเทศ
องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย (อบจ.)

ฝ่ายพัฒนาระบบสารสนเทศ

องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย

แบบทดสอบทักษะทางวิศวกรรมซอฟต์แวร์

เพื่อสรรหาบุคคลเข้ารับตำแหน่ง Front-end/Back-end Engineer

พฤษภาคม 2563

สงวนลิขสิทธิ์ตามกฎหมาย

คำแนะนำการทำแบบทดสอบ

1. แบบทดสอบนี้ถูกประเมินว่าสามารถทำได้ใน 24 ชั่วโมง อย่างไรก็ตาม คณะกรรมการฯ ใช้เวลาในทำโจทย์นี้ 72 ชั่วโมง เนื่องจากทราบว่าผู้สมัครแต่ละคนอาจจะมีช่วงเวลาที่เหมาะในการทำแบบทดสอบแตกต่างกัน
2. ให้ส่งคำตอบโดยการ fork repository <https://github.com/it-sgcu/sgcu63-recruitment> ไปที่ GitHub account ของตนเองที่ได้แจ้งไว้ในอีเมลตอบกลับ โดยให้คำตอบของแต่ละข้ออยู่ใน Folder นี้

Prob#<หมายเลขโจทย์> - <ชื่อโจทย์>

เมื่อทำแบบทดสอบระดับหนึ่งแล้ว ขอให้ push คำตอบไปที่ GitHub Repository ของตนเอง

- Code readability มีผลต่อการพิจารณา
 - คุณสามารถแก้ไขคำตอบโดยการ commit & push ได้ตลอดเวลาที่ให้ทำแบบทดสอบ (72 ชั่วโมง)
 - Commit message readability มีผลต่อการพิจารณาเช่นกัน
3. การตรวจคำตอบ จะตรวจโดยชุดทดสอบลับ และยึดคำตอบที่ถูก commit ครั้งสุดท้าย ก่อนเวลาสิ้นสุดการทำแบบทดสอบที่ได้รับในอีเมลของผู้สมัครแต่ละคน โปรดระลึกว่าการตรงต่อเวลาถือเป็นคุณสมบัติสำคัญอย่างหนึ่งของการเป็นวิศวกรซอฟต์แวร์ที่ดี
 4. ความคิดสร้างสรรค์ในการแก้ไขปัญหา ถือเป็นทักษะที่สำคัญอย่างหนึ่งในการพัฒนาระบบ อย่างไรก็ตาม โจทย์ทุกข้อมีจุดประสงค์ในการวัดทักษะของผู้สมัคร ดังนั้นหากคุณมีแนวคิดในการแก้ไขปัญหาที่ใช้ทักษะนอกเหนือจากที่โจทย์ต้องการวัด คุณสามารถระบุแนวคิดเพิ่มเติมได้ในส่วนของคำตอบที่เป็น source code ด้วยฟอรัมเมตนี้

// additional idea: <your idea>

อย่างไรก็ดี คำตอบหลักของคุณควรจะสามารถวัดผลได้ตามวัตถุประสงค์การทดสอบของโจทย์ข้อนั้นเสมอ

5. หากมีข้อสันนิษฐานเพิ่มเติมซึ่งโจทย์ไม่ได้ให้ข้อมูลไว้ ให้พิจารณาตามเห็นสมควร โดยให้ระบุข้อสันนิษฐานลงในส่วนของคำตอบที่เป็น source code ด้วยฟอรัมเมตนี้

// assumption: <your assumption>

อย่างไรก็ดี คำตอบของคุณควรจะสามารถวัดผลได้ตามวัตถุประสงค์การทดสอบของโจทย์ข้อนั้นเสมอ

6. สามารถใช้ภาษาใดก็ได้ในการเขียนโปรแกรมเพื่อแก้ไขปัญหา หากมีการใช้ library เพิ่มเติม กรุณาตรวจสอบว่าได้ระบุรายการ library เหล่านั้นตามข้อกำหนดของภาษาที่คุณใช้ไว้แล้ว
7. สามารถค้นหาข้อมูลเพิ่มเติมด้วยช่องทางใดก็ได้ ในระหว่างการทำแบบทดสอบนี้
8. การตัดสินใจของคณะกรรมการถือเป็นที่สุด

“สนุกกับมันดีกว่าครับ”

คณะกรรมการสรรหาบุคคลเข้ารับตำแหน่ง Front-end/Back-end Engineer

องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย

Problem 1: “The scraper”

Objectives:

1. Know concept of DOM structure.
2. Know some basics of HTTP request.
3. Know some basics of string manipulation.

ที่มหาวิทยาลัยจุฬาลงกรณ์ มีกิจกรรม “รับน้องก้าวใหม่” ซึ่งถือเป็นธรรมเนียมที่ปฏิบัติกันมาอย่างยาวนาน ซึ่งกิจกรรมนี้จะจัดขึ้นในเดือนสิงหาคมของทุกปี แต่เนื่องจากปีนี้มีแนวโน้มที่จะเกิดน้ำท่วมใหญ่ในมหาวิทยาลัย จึงจำเป็นต้องเลื่อนเข้ามาจัดในช่วงกลางเดือนกรกฎาคม

ภายในกิจกรรม “รับน้องก้าวใหม่” จะมี “บ้าน” ซึ่งเป็นซุ้มกิจกรรมให้นิสิตใหม่ได้เข้ามาร่วมสนุก โดยจะมีรูปแบบการจัดกิจกรรมที่แตกต่างกันไป ตามแนวคิดของบ้านนั้น ๆ ซึ่งในปัจจุบัน บ้านรับน้องได้มีการเปลี่ยนแปลงไปจากในอดีตอย่างมาก แนวคิดประจำบ้านของแต่ละบ้านเริ่มเปลี่ยนแปลงไปในทุก ๆ ปี

คุณซึ่งเป็น Senior Software Engineer มีหน้าที่หลักในการจัดการและควบคุมระบบสารสนเทศทุกอย่างของกิจกรรมนี้ จึงจำเป็นต้องรู้เกี่ยวกับบ้านรับน้องให้ดี ดังนั้นคุณต้องเริ่มศึกษาข้อมูลบ้านต่าง ๆ ในอดีตผ่านเว็บ

<https://rubnongkaomai.com/>



ภาพที่ 1 องค์ประกอบเว็บไซต์ rubnongkaomai.com

เมื่อคุณได้ศึกษาเว็บไซต์อย่างเข้าใจจนถึงแก่นแท้ของการออกแบบแล้ว คุณจะพบว่า “บ้าน” ภายในกิจกรรม “รับน้องก้าวใหม่” นั้นจะยังคงเดิมตลอดจนชั่วกาลปาวสาน เพียงแต่แนวคิด (สโลแกน) เปลี่ยนไปตามกาลเวลา

งานของคุณ: สร้างโปรแกรมที่เมื่อเรียกใช้แล้ว จะไปดึงข้อมูลมาจากส่วนของเว็บไซต์ <https://rubnongkaomai.com/> แล้วส่งออกข้อมูลชื่อบ้านและสโลแกนของบ้านนั้น ๆ ในเวอร์ชันเดียวกับที่ปรากฏบนเว็บไซต์ ลงในไฟล์ชื่อ “table.html” ซึ่งเมื่อเปิดด้วย Web browser แล้วจะปรากฏเป็นตารางดังภาพ

ชื่อบ้าน	สโลแกน
บ้านแอมป์	ABNormal ชื่อนี้มีแต่มิตรภาพและความอบอุ่น
บ้านอากาศเป๋	เมื่อเรารัก เรารักอย่างไม่มีเงื่อนไข
บ้านมีม	บ้านมีมณะ น้องไหวหรือ
บ้านดอก	บ้านดอก ดินแดนแห่งอารามาท
บ้านดุษฎี	ถ้าอยากรู้เรื่องราวให้กระจ่าง พวกเราก็อพร้อมแจ่มแจ้งไข เพื่อปกป้องความสุขของน้องๆ เราคือดุษฎีเกมนอน
บ้านอินเดียน่า	We will be blended together.
บ้านจัดสัน	จัดสัน อาหารเสิร์ฟ
บ้านคุณหนู	บ้านคุณหนูจะเข้าไปลู่ จะเข้าไปตกหัวใจ...
บ้านหมีน้อย	คุณพร้อมที่จะออกจากโลกของมิกเกิ้ลหรือยัง
บ้านฟักดาอากาศ	อะไรก็ไม่ว่าสำคัญเท่าเราได้ "ฟัก" ไปด้วยกัน
บ้านป่านารัก	บ้านอบอุ่นไม่เป็นรอง เสียงกลองดังสนั่น สันตนาการถึงใจ รับน้องใหม่สู่ครอบครัว

ภาพที่ 2 ตัวอย่างผลลัพธ์ของไฟล์ table.html (เฉพาะบางส่วน)

ทั้งนี้ ในไฟล์ html ของคุณ อาจมีการตกแต่งสไตล์ด้วย Cascading Style Sheets (CSS) เพิ่มเติมได้ตามต้องการ

Problem 2: “File search”

Objectives:

1. Be able to handle with JSON data type.
2. Be able to write a recursive function or know an algorithm to parse a tree in specific context.

Given a directory represented by JSON with the following format

- Folder is represented by a key. However, it must not begin with “_”
- All files in the folder are under a key named “_files” with the value of type array of strings.

For example:

```
FolderA
- file1
- file2
- SubfolderC
    - file1
- SubfolderB
    - file1
```

Should be represented as

```
" {
  "FolderA": {
    "_files": [ "file1", "file2" ] ,
    "SubfolderC": {
      "_files": [ "file1" ]
    } ,
    "SubfolderB": {
      "_files" : [ "file1" ]
    }
  }
} "
```

Your job: to implement a function

`fileSearch(fileToSearch: string, filesObj: string): string[]`

which will find the file exactly named <fileToSearch> in the JSON-formatted file directory <filesObj>. The output is an array of path of all files ordered by the depth of that file in ascending order. If the depths of 2 files are equal, the path should be ordered by alphabetical order of the path.

For example, with the above file structure, if the <fileToSearch> is “file1”, the output should be

```
[  
    "/FolderA/file1",  
    "/FolderA/SubfolderB/file1",  
    "/FolderA/SubfolderC/file1",  
]
```

Please pay attention to the order of paths.

You can implement this using any programming language. Do state the assumption you made if the info is not obviously specified.

Problem 3: “Numericalization”

Objectives:

1. Know the basics of programming (such as if-else, loop).
2. Be able to implement a complicate visualization system

การแสดงผลตัวเลขแบบ “Numericalization” คือแสดงเลขจำนวนเต็มบวกด้วยการประกอบกันของตัวเลขเป็นเลขนั้น ๆ ดังแสดงตามภาพด้านล่าง

99999	88888	77777	66666	55555	4	4	33333	22222	1	00000
9	9	8	8	7	6	5	4	4	3	2
99999	88888	7	66666	55555	44444	33333	22222	1	0	0
	9	8	8	7	6	6	5	4	3	2
99999	88888	7	66666	55555	4	33333	22222	1	00000	

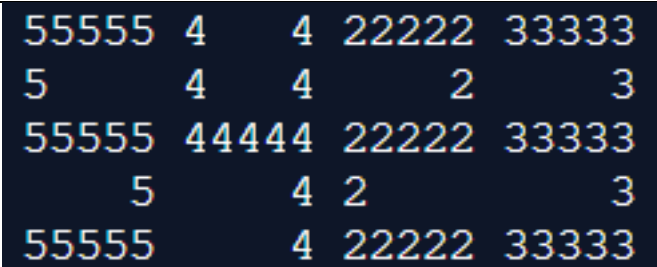
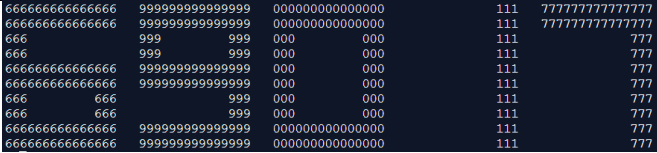
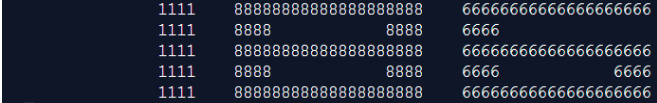
ภาพที่ 3 ตัวอย่างการแสดงผลเลขโดด ในแบบ " Numericalization "

รูปแบบของโปรแกรมจะมีการรับ input หนึ่งบรรทัด เป็นเลขจำนวนเต็มบวก 3 จำนวน คือ N, M1, M2 แต่ละตัวคั่นด้วยช่องว่างหนึ่งช่อง เลขแต่ละจำนวนมีความหมายดังนี้

- N คือ จำนวนเต็มบวกที่ต้องการแสดง
- M1 คือ กำลังขยายในแนวดิ่ง
- M2 คือ กำลังขยายในแนวนอน

งานของคุณ: สร้างโปรแกรมแสดงผลเลข N ในรูปแบบ “Numericalization” โดยเลขโดดแต่ละตัวมีส่วนสูง 5 x M1 อักขระ และความกว้าง 5 x M2 อักขระ และเลขโดดมีช่องว่างระหว่างกัน 1 x M2 อักขระ (กล่าวคือ ให้อยากการแสดงผลในภาพที่ 3 ตามแนวตั้ง M1 เท่า และขยายตามแนวนอน M2 เท่า)

ตัวอย่าง:

5423 1 1			
69017 2 3			
186 1 4			
(สังเกตระยะห่างหน้าตัวเลข 1)			