

Midassoft programming test for backend / full stack developer.

(last update: 2023-12-06)

Instructions.

- Please submit the code only in typescript language.
- The parameters and the return value of the function must be exactly as defined in the “**Signature**” section, and please remove or comment out “console.log” before submitting
- You can assume that the input is in the correct format. No need to validate.
- Please submit the code as text file (.ts), or github link and not as an image.
- The test consists of 4 questions. Your answers will be run against numerous test cases. So, please read each question thoroughly.

Question 1/4: clock angle



Given input time in hh24:mm format, calculate the smaller angle between the hour and minute hands in an analog clock.

The return value should be between 0 and 180

Signature

```
function getClockAngle(hh_mm:string):number {  
    // return the smaller angle between the hour and minute hands of the clock, in degree  
    // 0 <= return value <= 180  
}
```

Examples

```
getClockAngle("09:00") === 90  
getClockAngle("17:30") === 15
```

Question 2/4: Remote Associates Test

There is a test that measures a person's creativity, called the "Remote Associates Test". When you take this test, you are given 3 words/phrases and have to find a common word that goes with each of them.

Example: ROOM / SALTS / BLOOD

Answer: BATH

** This is because "bathroom", "bath salts", and "bloodbath" are all well-known words/phrases.

Your task is to create some questions for this test. You'll be given the 3 well-known words/phrases. And you'll need to tease out the question part.

Signature

```
function getQuestionPart(phrases:string[]):string[] {  
    // return array of three strings that makes a question for "Remote Associates Test".  
}
```

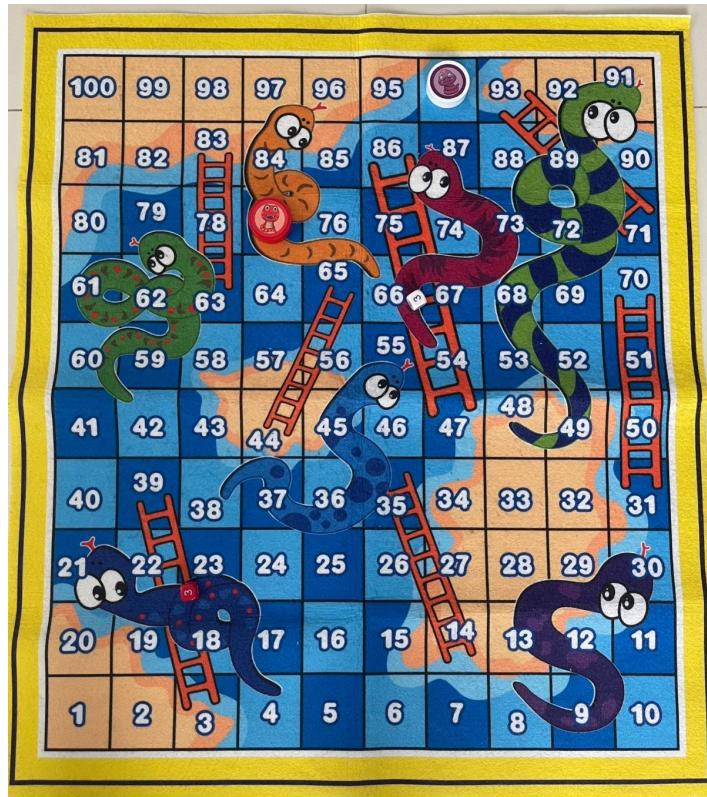
Note.

- All three input words/phrases are in all capital letters.
- You can assume that there is a common word among them.

Examples

```
getQuestionPart(["BATHROOM", "BATH SALTS", "BLOODBATH"]) === ["ROOM", "SALTS", "BLOOD"]  
getQuestionPart(["BEFRIEND", "GIRLFRIEND", "FRIENDSHIP"]) === ["BE", "GIRL", "SHIP"]
```

Question 3/4: Snakes and ladders



เกม “บันไดงู” ที่เด็กไทยรู้จักกันดี โดยนลูกเต่าหนึ่งลูก และเดินตามจำนวนบนหน้าลูกเต่า ถ้าเจอบันไดจะได้ทางลัดขึ้นต้านบน ถ้าเจองูจะตกลงมาช้างล่าง. เกมเริ่มจากต่าแรก 1, เล่นจนกว่าจะ “เต่าแรก” 100. (ถ้าเกิน 100 จะเดินโดยหลัง)

กรุณาระบุว่าในแต่ละรอบ ให้จำนวนครั้งน้อยที่สุดเป็นเท่าใด (อาจมีได้หลายคำตอบ ขอให้จำนวนครั้งน้อยที่สุดเป็นเท่าใด) โดยต้องใช้ได้กับทุกบอร์ด ไม่ใช่แค่ในภาพด้านล่างเท่านั้น

Assumption: บันไดขึ้นอย่างเดียว, งูตกอย่างเดียว, ไม่มีเลขใดที่เป็นทางเขื่อนบันไดและงูพร้อมกัน

Hint: มีข่าวลือว่า ในบางบอร์ดที่ขึ้นชื่อว่า “งู” บางทีก็ไม่ใช่สิงโตร้ายเสมอไป

Signature

```
function quickestPath(board: { ladders: [number, number][]; snakes: [number, number][]; }): number[] {
    // return array of roll results that reach 100 in the quickest possible way.
}
```

Examples

จากบอร์ดในภาพด้านบน

```
quickestPath({
    ladders: [ [3, 39], [14, 35], [31, 70], [44, 65], [47, 86], [63, 83], [71, 93] ],
    snakes: [ [21, 4], [30, 8], [55, 38], [79, 42], [87, 54], [91, 48], [96, 66] ]
}) === [ 2, 5, 6, 6, 1 ]
```

// คำตอบอื่นที่ถูกต้อง เช่น [2, 5, 6, 5, 2] เพราะโยนหัวครั้งเท่ากัน

Question 4/4: one dimension trip

มีถนนตรงเส้นยาวอยู่เส้นหนึ่ง

0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – ...

คุณต้องการจะเข้าเยี่ยมชมร้านค้าทุกร้านบนถนนเส้นนี้ และจึงเดินทางต่อไปยังเป้าหมายปลายทาง, โดยคุณสามารถจะเดินเอกสาร์ได้โดยจะเสียพลังงาน = 1 หน่วยต่อระยะทาง 1 unit, หรือคุณจะใช้รถประจำทางก็ได้ซึ่งจะจอด ณ ตำแหน่งที่เป็นสถานีเท่านั้น ซึ่งถือว่าใช้พลังงาน = 0 หน่วย.

ตำแหน่งของ จุดเริ่มต้น, ร้านค้า, สถานีรถประจำทาง, และ เป้าหมายปลายทาง ล้วนอยู่ในตำแหน่งจำนวนเต็ม

เป้าหมายคือคุณต้องการจะเดินทางไปเยี่ยมชมร้านค้าทุกร้าน (ลำดับไม่สำคัญ) และจึงเดินทางต่อไปยังเป้าหมายปลายทาง โดยใช้พลังงานรวมน้อยที่สุด

Signature

```
function minEnergy(start: number, shops: number[], stations: number[], target: number): number {  
    // return the minimum energy needed to visit all shops then go to the target.  
}
```

Examples

minEnergy(0, [4, 9], [3, 6, 8], 11) === 8

0(START) – 1 – 2 – 3(BUS) – 4(SHOP) – 5 – 6(BUS) – 7 – 8(BUS) – 9(SHOP) – 10 – 11(TARGET)

trip ที่ประหยัดพลังงานมากที่สุดย่อมจะเป็น
เดินจาก 0 => 4 (SHOP) // energy used = 4
เดินจาก 4 => 3 // energy used = 1
นั่งรถจาก 3 => 8
เดินจาก 8 => 9 (SHOP) // energy used = 1
เดินจาก 9 => 11 (TARGET) // energy used = 2
ใช้พลังงานรวม $4+1+1+2=8$