

## Description

\*-----

เหมือน AI จะรู้ตัวซะแล้ว

ไม่สิ หรือว่ารู้ตัวตั้งแต่แรกแล้ว ???

คุณก็ไม่อาจทราบได้แล้ว แต่ที่แน่ๆ คุณกำลังจะถูกขัง

อะ ถูกขังแล้ว

AI ได้ทำการ Activate สัญญาณฉุกเฉินดับเพลิง จริงๆจะกดปุ่มขโมยเพื่อเรียกตำรวจ แต่กดผิด ทำให้ระบบการปล่อยก๊าซเฉื่อยเข้าห้อง แบบเดียวกับที่ใช้ดับไฟ กำลังจะเริ่มทำงาน มุ่งหวังเพื่อสังหารคุณทั้งทางตรงและทางอ้อม

หากไม่หนี คุณจะตายแน่นอน

แต่ประตูทุกบานถูกปิด คุณได้ทำการวิ่งไปรอบๆห้องเพื่อหาทางออก แต่ก็ไม่มี

จนกระทั่งคุณเหยียบไปถูกแผ่นพื้นที่มีเสียงแปลกๆ คุณไม่มีทางเลือก จึงเปิดแผ่นพื้นขึ้นมาดู

\*-----

คุณได้พบกับแผงวงจรใต้พื้นที่แสดงเป็นไฟ LED จำนวน 16 แถว 16 หลัก และมีช่อง Input เป็นช่องสาย USB ที่รับข้อมูลทีละ 1 Byte

แผ่น LED จะแสดงไฟสีแดง แทนด้วย O และ จะมีไฟที่มีตัวเลขจะแทนด้วยเลขตัวนั้นๆเป็นเลขหลักหน่วย คุณจะต้องส่งข้อมูลตำแหน่งไฟที่แสดงตัวเลขจากน้อยไปมาก

โดยที่จะต้องส่งข้อมูลเป็นเลขฐานสองจำนวน 8 บิต โดยที่แบ่งเป็น 4:4 บิต

แทนด้วยตำแหน่ง x:y ของหลอดไฟที่มีตัวเลข

โดยที่มุมซ้ายบนคือจุด (0, 0)(00000000) ขวาล่างคือ (15, 15)(11111111)

งานของคุณคือ ให้แสดงตำแหน่งไฟที่แสดงผลตัวเลข จากน้อยไปมาก

\*-----

เมื่อคุณส่งข้อมูลได้ถูกต้อง แผ่นพื้นได้เปิดเป็นทางลับ ที่สามารถมุดออกมาจากห้องได้

โดยที่ก่อนหน้านี้ออกมา นั้น คุณได้หยิบ Harddisk ที่มีรอยของสัตว์เข้าไปข้างในออกมาจากเครื่อง AI

โดยที่คุณหวังว่า จะมีอะไรซ่อนอยู่ข้างในนั้น แต่ดูเหมือน Harddisk ลูกนี้ก็จะระบบทำลายตัวเองไว้

คุณจึงต้องดูข้อมูลข้างในให้เร็ว และให้ได้มากที่สุด

\*-----

by นายพิชาธร เอกอุ่น (<https://ejudge.it.kmitl.ac.th/account/7>)

13 September 2016, 12:47

## Specification

### ➡ Input Specification

### ➡ Output Specification

16 บรรทัด

จำนวนตัวเลขที่อยู่ใน Input บรรทัด

แต่ละบรรทัดจะเป็น String จำนวน 16 ตัวอักษร  
ประกอบไปด้วย O(โอ) และอาจจะผสมกับตัวเลขหลัก  
หน่วย

เป็นตำแหน่งของตัวเลขที่พบ จากน้อยไปมาก

ตัวเลขหลักหน่วยจะไม่ซ้ำกัน

## 1 2 3 ≡ Sample Case

### ➡ Sample Input

0000000000000000  
0000000000000000  
0000100000000000  
0000000000000000  
0000000050000000  
0030000000000000  
0000000000000000  
0000000000000000  
0000000000000000  
0000000000000000  
0000000000002000  
0000000000000000  
0000000000000000  
0000000000000000  
0000000000000000  
0000000000000000  
0000000000000000  
4000000000000000

### ➡ Sample Output

```
00001000 # <--- 0's Position
00100100 # <--- 1's Position
10011011 # <--- 2's Position
01010010 # <--- 3's Position
11110000 # <--- 4's Position
01000111 # <--- 5's Position
```

```
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000300000000000000
000000000000000000
000000000000000800
000000000000000000
000000000000000000
000000000000000000
000000000000000000
009000000000000000
```

```
10000011 # <--- 0's Position
10010011 # <--- 3's Position
10111101 # <--- 8's Position
11110010 # <--- 9's Position
```

🕒 Time Remaining

0

0

35

23

647

Day

Hour

Minutes

Seconds

## Information

**Time Limit**

1 Second

**Memory Limit**

32 MB

**Language**

py

**Deadline**

21 September 2016, 21:00

**Submission Limit****Restrict Word****Testcase**

10 case

**Rejudge Testcase**

0 case

**Full Score**

100 Point

**Bonus Score**

100

**Your Score**

N/A

## Submit File

 No file chosen