KU 01

เก็บของขวัญ

การทดสอบความสามารถในการแก้ปัญหาโดยการเขียนโปรแกรม การทดสอบวันที่ 23 พ.ย. 2562

ปราสาทแห่งหนึ่งเขียนแผนที่ภายในออกมาได้เป็นตารางขนาด R แถว C คอลัมน์ ดังตัวอย่างด้านล่างนี้ที่ R = 5, C = 7

```
A*.#...
..#..*#
.*####.
##..#..
.#.*#..
```

ส่วนที่เป็นทางเดินแสดงด้วยจุด (.) ส่วนที่เป็นกำแพงแสดงด้วยเครื่องหมายชาร์ป (#) ช่องที่เป็น A คือจุดเริ่มต้นที่คุณอยู่ ช่องที่เป็น * คือช่องที่มีของขวัญ คุณสามารถเดินไปมาในปราสาทได้ในช่องที่ไม่ใช่กำแพง ในการเดินนั้นคุณสามารถย้ายตำแหน่งได้สี่ทิศทางคือ (1) ขึ้นบน (2) ลงล่าง (3) ไปทางซ้ายและ (4) ไปทางขวา ถ้าช่องที่ต้องการเคลื่อนที่ไปไม่ใช่กำแพง (ที่แทนด้วย #) ไม่สามารถเดินทะลุออกนอกแผนที่ได้ ไม่สามารถเดินเฉียงได้ ในแผนที่ด้านบน บริเวณที่คุณเดินไปมาได้แสดงด้วยสีเหลือง สังเกตว่าคุณสามารถไปเก็บของขวัญได้ 2 ชิ้น

บางปราสาทอาจจะมีช่องวาร์ปข้ามมิติหลายช่อง ช่องเหล่านี้เมื่อเดินเข้าไปที่ช่องใดจะเลือกได้ว่าจะไปโผล่ที่ช่องวาร์ปช่องอื่น ๆ ได้ หรือจะโผล่กลับมาที่ช่องเดิมก็ได้ ช่องวาร์ปจะเขียนแทนด้วย W ในแผนที่ พิจารณาอีกตัวอย่างด้านล่างที่มีช่องวาร์ป 5 ช่อง ช่องที่คุณสามารถเดินไปได้แสดงเป็นสีเหลือง สังเกตว่าคุณสามารถเก็บของขวัญได้ 3 ชิ้น

```
A*.#...
.W#..*#
.*####.
##WW#.W
W#.*#..
```

ให้เขียนโปรแกรมรับแผนที่และคำนวณว่าคุณจะสามารถเก็บของขวัญได้มากที่สุดกี่ชิ้น มีข้อมูลทดสอบ 30% ที่ไม่มีช่องวาร์ปเลย

ข้อมูลนำเข้า

บรรทัดแรกระบุจำนวนเต็มสองจำนวน R และ C (1 <= R <= 30; 1 <= C <= 30)

อีก R บรรทัดระบุแผนที่ แต่ละบรรทัดระบุข้อความความยาว C ตัวอักษร โดยมีการใช้สัญลักษณ์ดังนี้

- # แทนช่องกำแพง
- . แทนพื้นที่ว่างธรรมดา
- A แทนจุดเริ่มต้นที่คุณอยู่ จะมีหนึ่งช่องในแผนที่ที่เป็น A
- * แทนช่องที่มีของขวัญ
- W แทนช่องที่เป็นช่องวาร์ป เมื่อเข้าไปในช่องวาร์ปสามารถไปโผล่ที่ช่องวาร์ปใดก็ได้ จะไม่ไปไหนก็ได้

ข้อมูลส่งออก

มีหนึ่งบรรทัด เป็นจำนวนเต็มของขวัญที่มากที่สุดที่คุณเก็บได้

เงื่อนไขการทำงาน โปรแกรมต้องทำงานภายใน 1 วินาที ใช้หน่วยความจำไม่เกิน 256 MB

(ตัวอย่างอยู่หน้าถัดไป)

ตัวอย่าง 1

Input	Output
5 7 A*.#... ..#..*# .*####. ##..#.. .#.*#..	2

ตัวอย่าง 2

Input	Output
5 7 A*.#... .W#..*# .*####. ##WW#.W W#.*#..	3