Hashing Table

ให้ Implement โครงสร้างข้อมูลแบบ Hashing ที่มีคุณสมบัติดังนี้

- Hash Function คือ H(x) = x mod N เมื่อ N คือขนาด Hash Table ปัจจุบัน
- ใช้ Chaining เมื่อเจอ Collision (ให้ Chain เป็น Linked List ที่เก็บข้อมูลตามลำดับที่เข้า)
- ถ้าจำนวนข้อมูลมีความจุเกิน 50% หลังการเพิ่ม ให้เพิ่มขนาด Hash Table เป็นสองเท่าแล้ว Rehash ใหม่ ลำดับการ Rehash เป็นไปตามลำดับข้อมูลใน Hash Table
- พิมพ์ข้อมูลตามที่อยู่ใน Hash Table ถ้าข้อมูลว่างเปล่าให้แสดงคำว่า empty

ข้อมูลเข้า

- บรรทัดแรกมีจำนวนเต็มบวกสองจำนวนคือ M ขนาด Hash Table เริ่มต้น และ N จำนวนสมาชิกที่จะเพิ่มเข้าไปใน Hash Table
- บรรทัดที่สามเป็นจำนวนเต็ม N จำนวน หรือสมาชิกที่จะเพิ่มเข้าไปใน Hash Table

ข้อมูลออก

- จำนวนบรรทัดเท่ากับขนาด Hash Table หลังการเพิ่มข้อมูลทั้งหมด
    - บรรทัดที่ i แสดงทุกค่าที่เก็บใน Hash Table ที่ตำแหน่ง i คั่นด้วยช่องว่าง
    - ถ้าข้อมูลว่างให้แสดงคำว่า empty

Examples

Input	Output
2 4 8 18 2 3	8 empty 18 2 3 empty empty empty empty

(คำอธิบายอยู่หน้าถัดไป)

คำอธิบายตัวอย่าง

เริ่มต้นที่ Hash Table ขนาด 2

เพิ่มข้อมูลแรกคือเลข 8 เข้าไปใน Hash Table จะใส่ในตำแหน่งที่ 0 เพราะ 8 mod 2 = 0

8	

เพิ่มข้อมูลเลข 18 เข้าไปใน Hash Table จะใส่ในตำแหน่งที่ 0 เพราะ 18 mod 2 = 0

8 18	

แต่เนื่องจากจำนวนข้อมูลมีจำนวนมากกว่าครึ่งหนึ่งของขนาด Hash Table เราจึงทำการเพิ่มขนาด Hash Table เป็น 4 และ Rehash 8 กับ 18 ใหม่ (8 % 4 = 0, 18 % 4 = 2)

8		18	

เพิ่มข้อมูลเลข 2 เข้าไปใน Hash Table จะใส่ในตำแหน่งที่ 2 เพราะ 2 mod 4 = 2

8		18 2	

แต่เนื่องจากจำนวนข้อมูลมีจำนวนมากกว่าครึ่งหนึ่งของขนาด Hash Table เราจึงทำการเพิ่มขนาด Hash Table เป็น 8 และ Rehash 8, 18 และ 2 ใหม่ (8 % 8 = 0, 18 % 8 = 2, 2 % 8 = 2)

8		18 2					

เพิ่มข้อมูลเลข 3 เข้าไปใน Hash Table จะใส่ในตำแหน่งที่ 3 เพราะ 3 mod 8 = 3

8		18 2	3				