

วงจรบวกเลข (Adder)

Time limit: 1 sec

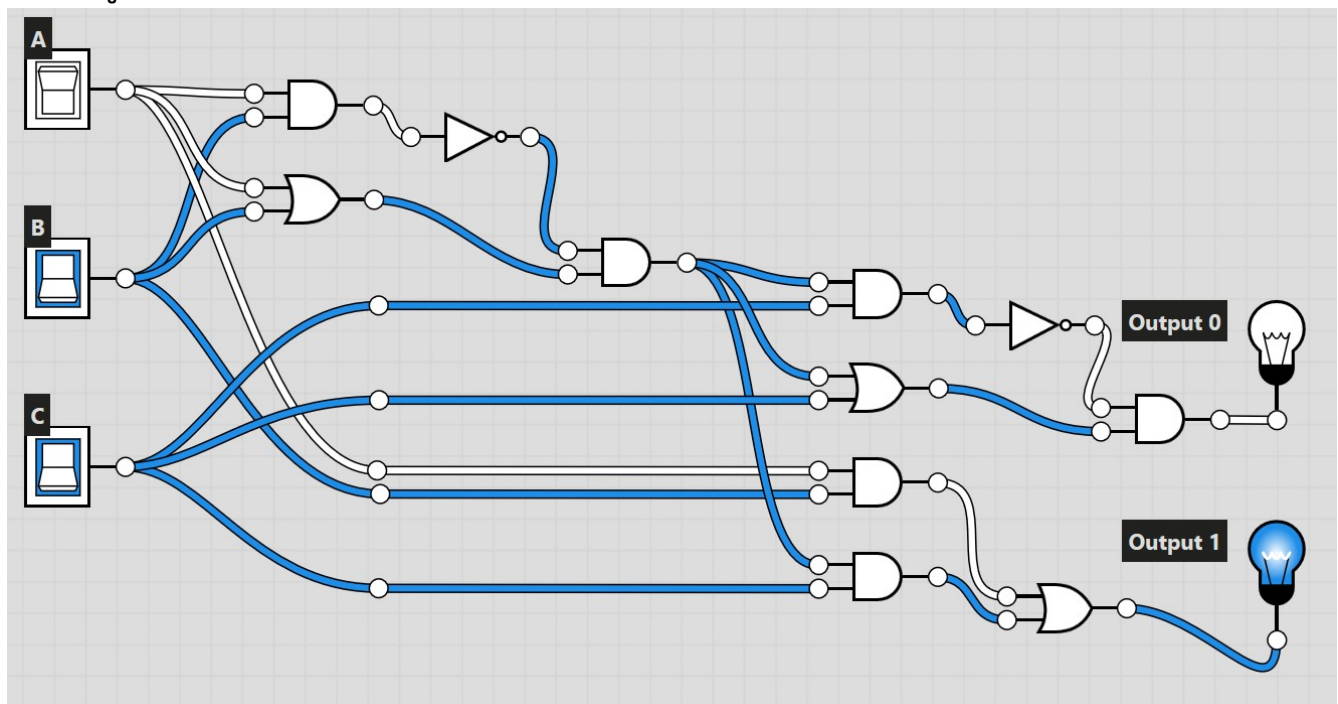
memory limit: 512MB

การบวกเลขฐานสองขนาด n บิตนั้นสามารถทำได้โดยใช้วงจรซึ่งประกอบด้วยเกตแบบ AND, OR และ NOT โดยวงจรจะประกอบด้วยเกตต่าง ๆ เหล่านั้นเชื่อมต่อกันเกตแบบ AND และ OR จะรับ input 2 บิต และผลิต output จำนวน 1 บิต ส่วนเกต NOT จะรับ input และมี output ขนาด 1 บิต

วงจรบวกเลขฐานสองรับ input เป็นตัวเลขฐานสอง ขนาด n บิต จำนวน 2 ตัวเลข คือ A และ B และรับ "เลขทด" C ขนาด 1 บิต และจะสร้าง output ขนาด $n+1$ บิต เป็นเลขฐานสองของการบวกตัวเลขทั้งสองตัวกับ "เลขทด" เข้าด้วยกัน ตัวอย่างด้านล่างนี้แสดงถึง input และ output บางตัวอย่างของวงจรบวกเลขขนาด 2 บิต

A ในรูปเลขฐาน 2	B ในรูปเลขฐาน 2	C ในรูปเลขฐาน 2	ผลลัพธ์ ในรูปเลขฐาน 2
00	00	0	000
00	00	1	001
11	11	0	110
11	11	1	111
10	01	1	100
01	01	1	011
11	00	1	100
10	11	1	110

รูปด้านล่างนี้แสดงถึงวงจรบวกเลขขนาด 1 บิต



วงจรบวกเลขนั้นจะเป็นการเชื่อมต่อเกตต่าง ๆ เข้าด้วยกัน เราสามารถพิจารณาเกตเหล่านั้นว่าเป็นกราฟแบบมีทิศทางได้ โดยที่ แต่ละปมคือเกตต่าง ๆ และการเชื่อมต่อจาก output ของเกต p ไปยัง input ของเกต q คือ เส้นเชื่อมแบบมีทิศทางในกราฟ จากปม p ไปยังปม q

วงจรบวกเลขนั้นจะต้องเป็นกราฟแบบมีทิศทางที่ไม่มีวงวน (Directed Acyclic Graph) เท่านั้น กำหนดให้ "ความลึก" ของกราฟ คือค่าความยาวของเส้นทางที่ยาวที่สุดในกราฟวงจรบวกเลข จงสร้างวงจรบวกเลขขนาด n บิต ที่มี "ความลึก" น้อยที่สุด

Input

- บรรทัดแรกประกอบด้วยจำนวนเต็มสองตัวคือ n และ m ซึ่งบอกจำนวนบิตที่ต้องการและจำนวนเกตที่ห้ามใช้เกิน

Output

- ประกอบด้วยข้อมูลหลายบรรทัดซึ่งระบุถึงเกตและการเชื่อมต่อของเกต บรรทัดแรกประกอบด้วยจำนวนเต็ม 1 ตัวคือ m ซึ่งระบุจำนวนเกตทั้งหมด เกตแต่ละเกตกำกับด้วยหมายเลข 1 ถึง m
- บรรทัดที่สองจะต้องประกอบด้วยตัวเลขจำนวน $n + 1$ ตัว ซึ่งระบุหมายเลขของเกตที่ output ของเกตนั้นจะเป็น output ของวงจรบวกเลข เรียงลำดับจาก output บิตที่ 0 ถึง output บิตที่ n ตามลำดับ (บิตที่ 0 หมายถึง least significant bit)
- บรรทัดที่ 3 ถึง $3+m-1$ จะเป็นข้อมูลของแต่ละเกตตั้งแต่เกตหมายเลข 1 ถึงหมายเลข m บรรทัดละ 1 เกตตามลำดับ สำหรับบรรทัดที่อธิบายเกตหมายเลข k นั้น จะประกอบด้วยข้อมูลสองส่วนคือ ประเภทของเกต ซึ่งเป็นคำว่า "AND" หรือ "OR" หรือ "NOT" ตามด้วยจำนวนเต็ม 1 หรือ 2 ตัวซึ่งระบุว่า input ของเกตหมายเลข k นั้นคืออะไร
 - สำหรับเกตประเภท AND และ OR นั้นจะต้องมี input จำนวน 2 input เสมอ และเกตประเภท NOT จะต้องมี input จำนวน 1 input เสมอ
 - การระบุ input ของเกตหมายเลข k นั้นมีรูปแบบดังต่อไปนี้
 - ถ้าเป็นตัวเลขจำนวนเต็มตั้งแต่ 1 ถึง m หมายถึง input คือ output จากเกตหมายเลขดังกล่าว
 - ถ้าเป็นค่า 0 หมายถึง input คือ "เลขทด" ขนาด 1 บิต
 - ถ้าเป็นค่า -1 ถึง $-n$ หมายถึง input บิตที่ 0 ถึง บิตที่ $n-1$ ของ A ตามลำดับ
 - ถ้าเป็นค่า $-(n+1)$ ถึง $-(2n)$ หมายถึง input บิตที่ 0 ถึง บิตที่ $n-1$ ของ B ตามลำดับ

ให้ดูตัวอย่างด้านล่างสำหรับ Output ที่อธิบายถึงวงจรบวกเลข 1 บิตตามภาพด้านบน

Example

Input	Output
1 100	11 8 11 AND -1 -2 NOT 1

	OR -1 -2
	AND 2 3
	AND 4 0
	NOT 5
	OR 4 0
	AND 6 7
	AND -1 -2
	AND 4 0
	OR 9 10

Subtask

แต่ละปัญหาย่อยนั้นมีเพียงชุดข้อมูลทดสอบเดียว โดยที่คำตอบจะต้องใช้เกตไม่เกินค่าที่กำหนดให้ และวงจรที่ได้จะต้องเป็นไปตามเงื่อนไขข้างต้น ถ้าหากไม่เป็นไปตามเงื่อนไขนี้ จะไม่ได้คะแนนในปัญหาย่อยนั้น

สำหรับปัญหาย่อย 3 เป็นต้นไป คะแนนที่ได้จะแปรผกผันกับ ค่าความลึกของวงจรรวมเลขที่สร้างได้ โดยสำหรับคำตอบที่ได้ค่าความลึกเป็น h นั้น คะแนนที่ได้คือ $(h_{\min} / h * \text{คะแนนของปัญหาย่อย})$ โดยที่ h_{\min} คือค่าความลึกที่น้อยที่สุดที่ทำงานถูกต้องของผู้ที่ส่งทุกคน รวมถึง คำตอบของ judge ด้วย

- ปัญหาย่อย 1 (10%): $N = 4, m = 100$
- ปัญหาย่อย 2 (15%): $N = 32, m = 50\,000$
- ปัญหาย่อย 3 (30%): $N = 53, m = 200\,000$
- ปัญหาย่อย 3 (45%): $N = 97, m = 500\,000$

Grader ตัวอย่าง

ไจกซ์xonนี้มีตัวตรวจตัวอย่างให้ คือ `adder_checker.cpp` วิธีการเรียกใช้งาน `adder_checker` คือ `adder_checker input_file circuit_file check_file` โดยที่

`input_file` คือชื่อไฟล์ข้อมูลนำเข้า (ระบุค่า n m ตามรูปแบบในหัวข้อ Input)

`circuit_file` คือชื่อไฟล์ของข้อมูลวงจรที่ต้องการตรวจสอบ (ตามรูปแบบในหัวข้อ Output)

`check_file` คือชื่อไฟล์ข้อมูลทดสอบที่จะทดสอบวงจร ไฟล์นี้มีรูปแบบดังต่อไปนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม k ซึ่งระบุจำนวนชุดข้อมูลทดสอบ
- สำหรับแต่ละข้อมูลทดสอบ จะประกอบด้วยข้อมูลสี่บรรทัด
 - บรรทัดแรก คือ สายอักขระ 0 1 ความยาว n ซึ่งระบุค่า A
 - บรรทัดสอง คือ สายอักขระ 0 1 ความยาว n ซึ่งระบุค่า B
 - บรรทัดที่สาม ระบุค่า c จำนวน 1 บิต
 - บรรทัดสี่ คือ สายอักขระ 0 1 ความยาว $n+1$ ซึ่งระบุค่า ของ $A + B + c$
 - ข้อมูล A, B, C นั้นระบุโดยเรียงจาก least significant bit ไปยัง most significant bit (กล่าวคือ สายอักขระเรียงจากบิต 0, 1, 2, ...) ตัวอย่างด้านล่างนี้แสดงตัวอย่างของ `check_file` สำหรับวงจรขนาด 4 บิต

ตัวอย่าง check_file
2
1010
1111
0
00101
1111
1101
0
01011