

Expression

1 second, 64 megabytes

ในการแทนนิพจน์ (expression) ใด ๆ ด้วยฟังก์ชัน นิพจน์หลักจะถูกแบ่งเป็นนิพจน์ย่อยๆ ด้วยตัวดำเนินการ (operator) ต่าง ๆ ดังนี้ การบวก “+”, วงเล็บ “()”, การคูณ “*” และการยกกำลัง “^” โดยสามารถเขียนแทนด้วยฟังก์ชันได้ ดังนี้ $op(i, e)$ โดยที่ e หมายถึงนิพจน์ทางคณิตศาสตร์ใด ๆ ซึ่งสามารถถูกแบ่งเป็นนิพจน์ย่อย ๆ ได้โดยใช้ตัวดำเนินการที่มีลำดับความสำคัญในการทำงาน (priority) ต่ำสุดในนิพจน์นั้น และ i คือลำดับของนิพจน์ย่อยนั้นๆ ตัวอย่างเช่น นิพจน์ $a*b+b*c+c*d$ สามารถแบ่งเป็นสามนิพจน์ย่อย โดยมีนิพจน์ย่อยที่ 1 คือ $a*b$, นิพจน์ย่อยที่ 2 คือ $b*c$ และนิพจน์ย่อยที่ 3 คือ $c*d$ เนื่องจากตัวดำเนินการ “+” มีความสำคัญต่ำสุดในการทำงานในนิพจน์นี้

กำหนดให้ลำดับความสำคัญในการทำงานของตัวดำเนินการจากมากสุดไปน้อยสุดมีดังนี้ “()”, “^”, “*” และ “+” ตามลำดับ

วัตถุประสงค์ของฟังก์ชันแทนนิพจน์คือ ต้องการแทนนิพจน์ย่อยด้วยฟังก์ชันเพื่อใช้ในการคำนวณ เช่น $op(2, e)$ แทนนิพจน์ย่อยลำดับที่สองของ e ที่กำหนดให้ข้างบน ($a*b+b*c+c*d$) ซึ่งจะได้ $op(2, e) = b*c$

ตัวอย่าง

กำหนดให้นิพจน์ p มีค่าดังนี้: $a^b*c+(d*c)^f*z+b$, สามารถแทนนิพจน์ย่อยใดๆ ของ p ด้วยฟังก์ชันได้ดังนี้

- $op(3, p) = b$
- $op(1, op(3, p)) = b$
- $op(2, p) = (d*c)^f*z$
- $op(1, op(2, p)) = (d*c)^f$
- $op(1, op(1, op(2, p))) = (d*c)$
- $op(1, op(1, op(1, op(2, p)))) = d*c$
- $op(2, op(1, op(1, op(2, p)))) = \text{null}$ (ไม่มีคำตอบ)
- $op(2, op(2, p)) = z$

โจทย์ จงเขียนโปรแกรมเพื่อรับข้อมูลนิพจน์ p ใด ๆ และฟังก์ชันคำถาม จากนั้นคำนวณหานิพจน์ย่อยของ p ที่สอดคล้องกับฟังก์ชันที่กำหนด

ข้อมูลนำเข้า

บรรทัดแรก รับนิพจน์หลัก p ที่ประกอบด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็ก a ถึง z และตัวดำเนินการเขียนติดกัน โดยไม่มีช่องว่าง รับประกันว่าความยาวตัวอักษรและตัวดำเนินการรวมกันไม่เกิน 64 ตัว

บรรทัดที่สอง รับเลขจำนวนเต็มบวก n ($1 \leq n \leq 10$) แสดงจำนวนฟังก์ชันคำถาม n ฟังก์ชัน

บรรทัดที่ 3 ถึง $n + 2$ บรรทัดที่ $i + 2$ ให้รับฟังก์ชันคำถามที่ i โดยแต่ละบรรทัดประกอบด้วยเลขจำนวนเต็มบวกอยู่ระหว่าง 1 ถึง 9 คั่นด้วยช่องว่าง 1 ช่อง และปิดท้ายด้วย 0

ตัวอย่างข้อมูลนำเข้าในบรรทัดที่ 3 ถึง $n + 2$

ข้อมูลนำเข้า 3 0 หมายถึงฟังก์ชัน $op(3, p)$

ข้อมูลนำเข้า 2 1 1 0 หมายถึงฟังก์ชัน $op(1, op(1, op(2, p)))$

ข้อมูลนำเข้า 1 2 2 0 หมายถึงฟังก์ชัน $op(2, op(2, op(1, p)))$

ข้อมูลส่งออก

มี n บรรทัด บรรทัดที่ i ให้แสดงฟังก์ชันและนิพจน์ย่อยที่สอดคล้องกับฟังก์ชันคำถามที่ i โดยในแต่ละบรรทัดของข้อมูลส่งออกจะต้องไม่มีการเว้นวรรคใดๆ กรณีที่ไม่มีคำตอบให้แสดง “null”

ตัวอย่างข้อมูลนำเข้าและข้อมูลส่งออก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
$a*b^c+d*e^f$ 2 1 0 2 0	$op(1,p)=a*b^c$ $op(2,p)=d*e^f$
$a*b^c+d*e^f$ 3 1 1 0 1 2 0 1 2 2 0	$op(1,op(1,p))=a$ $op(2,op(1,p))=b^c$ $op(2,op(2,op(1,p)))=c$
$(x+y)+z$ 5 1 0 1 1 0 1 1 1 0 1 1 2 0 3 0	$op(1,p)=(x+y)$ $op(1,op(1,p))=x+y$ $op(1,op(1,op(1,p)))=x$ $op(2,op(1,op(1,p)))=y$ $op(3,p)=null$

การให้คะแนน

ในข้อมูลทดสอบ 10 ชุด จะมีนิพจน์ที่ใช้ตัวดำเนินการ “วงเล็บ” จำนวน 5 ชุด (50%)

แหล่งที่มา

การแข่งขันคอมพิวเตอร์โอลิมปิก สอวน. ครั้งที่ 3 มหาวิทยาลัยขอนแก่น