

## Simple Expression Calculator

### จงเขียนโปรแกรม

- รับสตริงนิพจน์ที่ประกอบด้วย จำนวนเต็มต่าง ๆ คั่นด้วยเครื่องหมายบวกหรือลบ (ไม่มีช่องว่างใด ๆ) เช่น **1+20-300+4000** โดยจำนวนเต็มแรก (ทางซ้ายสุด) อาจเป็นจำนวนลบได้ เช่น **-1000+25**
- แสดงผลที่ได้จากการคำนวณนิพจน์ที่ได้รับ เช่น รับ **1+20-300+4000** จะแสดง **3721**

### หมายเหตุ

- ข้อมูลอินพุตที่ใช้ตรวจการทำงานของโปรแกรม จะไม่มีกรณีแปลก ๆ ที่ไม่เป็นไปตามข้อกำหนดของอินพุตแน่ ๆ เช่น
  - **1+2/3**
  - **1+2-3.0**
  - **+11+4**
  - **11-4+**
  - **one+two**
  - **(2+3)**
  - **1 + 2 - 3**

**ข้อห้าม** (ข้อห้ามข้างล่างนี้ ไม่ได้ถูกตรวจสอบในระบบเกรดเดอร์ แต่จะได้รับการตรวจหลังสอบ หากพบ จะได้ 0)

- ห้าม **import** ใด ๆ เพิ่ม
- ห้ามใช้คำสั่ง **eval** และ/หรือ **exec** ในโปรแกรม

### ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
20	20
-13	-13
1+2+3+4+5	15
-1-2-3-4-5	-15
9999-999+99-9	9090
10-9+8-7+6-5+4-3+2-1	5

## Flowchart

จงเขียนโปรแกรมที่ทำงานตามผังงานทางขวานี้

### ข้อมูลนำเข้า

จำนวนเต็ม 4 จำนวนบนบรรทัดเดียวกัน คั่นด้วยช่องว่าง

### ข้อมูลส่งออก

ค่าของ **a b c** และ **d** ในผังงาน คั่นด้วยช่องว่างหนึ่งช่อง

### ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
1 2 3 4	9 2 7 4
4 3 5 6	5 4 5 4

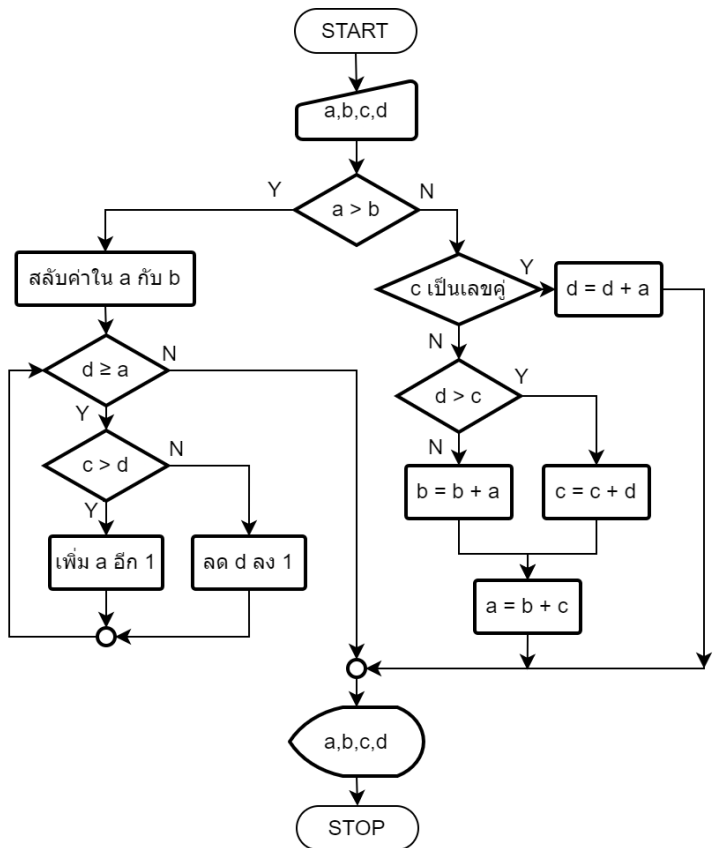
คำเตือน: ตัวอย่างข้างบนนี้ไม่ได้ครอบคลุมทุกกรณีของการทำงานในผังงาน

สำหรับคำสั่ง **input** เข้าเก็บในตัวแปร **a,b,c** และ **d** ในด้านบนผังงาน ให้ใช้คำสั่งข้างล่างนี้

```
a,b,c,d = [int(e) for e in input().split()]
```

สำหรับคำสั่งการแสดงค่าในตัวแปร **a,b,c** และ **d** ให้ใช้คำสั่งข้างล่างนี้

```
print(a,b,c,d)
```



## Multiplexer

จงเขียนฟังก์ชัน **f1**, **f2**, **f3**, **f4** และ **main** ให้ทำงานตามที่อธิบายกำกับแต่ละฟังก์ชัน ในโครงของโปรแกรมข้างล่างนี้ โดยที่ **a**, **b** และ **c** ที่เป็นพารามิเตอร์ของฟังก์ชัน **f1**, **f2**, **f3** และ **f4** มีลักษณะดังนี้แน่ ๆ ไม่ต้องตรวจอะไรในฟังก์ชัน

- เป็นจำนวนเต็ม มีค่าได้ตั้งแต่ -1000 ถึง 1000
- มีอย่างน้อยค่าหนึ่งใน **a**, **b** หรือ **c** ที่มีค่ามากกว่า 0
- มีอย่างน้อยค่าหนึ่งใน **a**, **b** หรือ **c** ที่มีค่าน้อยกว่า 0

```
def f1(a, b, c):  
    # a, b และ c มีค่าตามข้อกำหนดข้างบน  
    # คืนจำนวนเต็มที่มีค่าเท่ากับค่าน้อยที่สุดของ a, b และ c คิดเฉพาะตัวที่มีค่ามากกว่า 0  
  
def f2(a, b, c):  
    # a, b และ c มีค่าตามข้อกำหนดข้างบน  
    # คืนจำนวนเต็มที่มีค่าเท่ากับค่ามากที่สุดของ a, b และ c คิดเฉพาะตัวที่มีค่าน้อยกว่า 0  
  
def f3(a, b, c):  
    # a, b และ c มีค่าตามข้อกำหนดข้างบน  
    # คืนจำนวนเต็ม (ไม่มีเครื่องหมาย) ที่เป็นตัวเลขหลักแรก (ซ้ายสุด) ของผลบวกของ a, b และ c  
  
def f4(a, b, c):  
    # a, b และ c มีค่าตามข้อกำหนดข้างบน  
    # คืนจำนวนเต็ม (ไม่มีเครื่องหมาย) ที่เป็นตัวเลขหลักสุดท้าย (ขวาสุด) ของผลบวกของ a, b และ c  
  
def main():  
    # รับข้อมูลจากคีย์บอร์ดเป็นจำนวนเต็ม 5 จำนวน s1 s2 a b c คั่นด้วยช่องว่างในบรรทัดเดียวกัน  
    # ค่าของ a, b และ c เป็นไปตามข้อกำหนดข้างบน  
    # ถ้า s1 เป็น 0 และ s2 เป็น 0 ให้ print ค่าของ f1(a, b, c)  
    # ถ้า s1 เป็น 0 และ s2 เป็น 1 ให้ print ค่าของ f2(a, b, c)  
    # ถ้า s1 เป็น 1 และ s2 เป็น 0 ให้ print ค่าของ f3(a, b, c)  
    # ถ้า s1 เป็น 1 และ s2 เป็น 1 ให้ print ค่าของ f4(a, b, c)  
    # ถ้า s1 หรือ s2 เป็นค่าอื่น ให้แสดงคำว่า Error  
  
exec(input().strip()) # DON'T remove this line
```

## ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
print(f1(2, -18, 4))	2
print(f2(-20, -18, 4))	-18
print(f3(2, -18, 4))	1
print(f4(2, -18, 4))	2
main() 1 1 -2 -18 4	6
main() 10 1 -2 -4 6	Error