

แบบฝึกหัดภาคปฏิบัติการ สัปดาห์ที่สิบเอ็ด

ฟังก์ชันและตัวชี้

อ.ดร.ภิญโญ แท้ประสาทสิทธิ์ ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร
คำสั่ง

1. ให้เขียนโปรแกรมภาษาซีสำหรับปัญหาที่ให้ไป
2. ระเบียบการส่งงานเขียนโปรแกรมก็คือนักศึกษาจะส่งโค้ดเข้าไปโปรแกรมตรวจงาน

ปัญหา 1 ค่าเฉลี่ยและความแปรปรวน [average_and_variance1]

เขียนโปรแกรมเพื่อรับเลขจำนวนเต็ม 8 ตัวเก็บไว้ในอาร์เรย์ของฟังก์ชัน main จากนั้นให้ส่งอาร์เรย์ดังกล่าวไปคำนวณผลในฟังก์ชัน `double average(int* array)` เพื่อคำนวณค่าเฉลี่ยของข้อมูลในอาร์เรย์ จากนั้นให้ใช้ผลลัพธ์ที่ได้จากฟังก์ชัน `average` ไปเป็นพารามิเตอร์ของฟังก์ชัน

```
double variance(int* array, double avg);
```

เพื่อคำนวณค่าความแปรปรวนตามสูตรด้านล่าง ท้ายสุดให้พิมพ์ค่าเฉลี่ยและค่าความแปรปรวนออกมาในฟังก์ชัน main

หมายเหตุ ผลลัพธ์ที่ได้จากฟังก์ชัน `average` ถูกป้อนให้เป็นพารามิเตอร์ตัวที่สองของฟังก์ชัน `variance`

สมการสำหรับคำนวณความแปรปรวน

$$\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$
 โดยที่ N คือจำนวนข้อมูลซึ่งในที่นี้มีค่าเท่ากับ 8, x_i คือข้อมูลแต่ละตัวในอาร์เรย์ และ \bar{x} คือค่าเฉลี่ยของข้อมูลทั้ง 8 (หาได้มาจากฟังก์ชัน `average`)

ข้อมูลเข้า เป็นเลขจำนวนเต็มจำนวนแปดตัว จะเป็นเลขติดลบก็ได้

ผลลัพธ์ ค่าเฉลี่ยตามด้วยค่าความแปรปรวนของเลขจำนวนเต็มทั้งแปด เลขทั้งสองคั่นด้วยช่องว่าง ทศนิยมสองตำแหน่ง หมายเหตุ การคำนวณเลขทศนิยมในข้อนี้ให้ใช้ชนิดข้อมูลเป็นแบบ `double`

ตัวอย่าง

| ข้อมูลเข้า | ผลลัพธ์ |
|---------------------|------------|
| 1 2 3 4 5 6 7 8 | 4.50 6.00 |
| 6 7 8 9 10 11 12 13 | 9.50 6.00 |
| 7 -2 5 3 -1 0 6 -4 | 1.75 16.50 |

ปัญหา 2 ค่าเฉลี่ยและความแปรปรวน [average_and_variance2]

เขียนโปรแกรมเพื่อรับเลขจำนวนเต็ม N ตัวเก็บไว้ในอาร์เรย์ของฟังก์ชัน `main` จากนั้นให้ส่งอาร์เรย์ดังกล่าวไปคำนวณผลในฟังก์ชัน `double average(int* array, int N)` เพื่อคำนวณค่าเฉลี่ยของข้อมูลในอาร์เรย์จากนั้นให้ใช้ผลลัพธ์ที่ได้จากฟังก์ชัน `average` ไปเป็นพารามิเตอร์ของฟังก์ชัน `double varaince(int* array, double avg, int N)`; เพื่อคำนวณค่าความแปรปรวน ทำยสุดให้พิมพ์ค่าเฉลี่ยและค่าความแปรปรวนออกมาในฟังก์ชัน `main`

หมายเหตุ ความแตกต่างจากข้อที่แล้วก็คือว่า ในข้อนี้ขนาดของอาร์เรย์ถูกกำหนดด้วยตัวแปร N ไม่ได้มีขนาดตายตัว

ข้อมูลเข้า บรรทัดแรกเป็นเลขจำนวนเต็ม N โดยที่ $1 < N \leq 100$ ระบุจำนวนข้อมูลที่จะใช้คำนวณค่าทางสถิติจำนวนแปดตัว ส่วนบรรทัดที่สองจะเป็นเลขจำนวนเต็ม N ตัวใช้สำหรับคำนวณค่าทางสถิติ เลขแต่ละตัวคั่นด้วยช่องว่าง

ผลลัพธ์ ค่าเฉลี่ยตามด้วยค่าความแปรปรวนของ เลขทั้งสองคั่นด้วยช่องว่าง มีทศนิยมสองตำแหน่ง

ตัวอย่าง

| ข้อมูลเข้า | ผลลัพธ์ |
|-----------------------------------|------------|
| 8 1 2 3 4 5 6 7 8 | 4.50 6.00 |
| 12 7 -2 5 3 -1 0 6 -4 1 9 5 -8 | 1.75 24.93 |

ปัญหา 3 ฟังก์ชันทดสอบว่ามีเลขที่ซ้ำกันอยู่ในอาเรย์หรือไม่ [isUnique]

จงเขียนฟังก์ชัน isUnique ซึ่งทำการตรวจสอบเลขในอาเรย์จำนวนเต็ม arData ว่าเลขทุกตัวไม่มีซ้ำกันเลยใช่หรือไม่ หากเลขทุกตัวในอาเรย์ไม่ซ้ำกันเลย อาเรย์จะคืนเลข 1 แต่หากมีเลขที่ซ้ำกันอยู่อย่างน้อยหนึ่งตัวโปรแกรมจะคืนเลข 0 ทั้งนี้ arData มีตัวเลขอยู่ทั้งหมด N ตัว สำหรับฟังก์ชัน isUnique มีแม่แบบดังนี้

```
int isUnique(int* arData, int N);
```

[มีตัวอย่างข้อมูลเข้าและผลลัพธ์อยู่ในหน้าถัดไป]

คำแนะนำ 1 ข้อนี้ควรมีการลูบสองชั้น ชั้นแรก (ลูบชั้นนอก) ทำการเลือกตัวที่เราต้องการตรวจสอบว่าซ้ำหรือไม่ ส่วนชั้นที่สอง (ลูบชั้นใน) ทำหน้าที่ตรวจดูว่าตัวเลขที่เราสนใจนั้นซ้ำกับตัวอื่น ๆ ในอาเรย์หรือไม่

คำแนะนำ 2 ความยากสำหรับมือใหม่ก็คือตรงลูบด้านใน คือเราต้องทำการตรวจสอบโดยไม่เผลอไปเปรียบเทียบตัวเลขที่เราสนใจกับตัวของมันเอง (แล้วพลาดไปสรุปว่าเลขซ้ำกัน) วิธีป้องกันปัญหานี้มีสองแบบ

แบบแรกคือการป้องกันผ่านการกำหนดตำแหน่งเริ่มต้นและสิ้นสุดลูบแต่ละชั้น เพื่อรับประกันว่าการเปรียบเทียบจะไม่เกิดขึ้นกับอาเรย์ช่องเดียวกันเป็นอันขาด และจะไม่มีเปรียบเทียบที่ซ้ำซ้อนด้วย (วิธีนี้ดี เพราะรวดเร็ว และเป็นที่ยอมรับในหมู่ผู้มีประสบการณ์) สามารถดูตัวอย่างโค้ดได้จากด้านท้ายปัญหาข้อ 5

แบบที่สองคือการวนอาเรย์ช่องที่ k กับทุกช่องที่ไม่ใช่ k โดยป้องกันการเปรียบเทียบซ้ำกับ k ด้วยการ ใช้ if แบบง่าย ๆ ข้อดีของวิธีนี้คือมือใหม่รู้สึกเข้าใจง่ายว่าการเปรียบเทียบเกิดขึ้นโดยไม่มีเลขคู่ใดที่ถูกละเอียด แต่ข้อเสียก็คือมีการเปรียบเทียบเลขซ้ำซ้อนทำให้เสียเวลาในการคำนวณไปมากโดยไม่มีประโยชน์เชิงคำนวณเกิดขึ้นมา

เรื่องอื่น ๆ ถ้าคิดไม่ออกและจำเป็นต้องใช้ตัวช่วย ลองดูโครงสร้างของฟังก์ชันหลังข้อ 5 อย่างไรก็ตาม ผู้เรียนก็ไม่ควรหวังพึ่งตัวช่วย จนกว่าจะได้ลองคิดด้วยตนเองอย่างจริงจังแล้ว

ตัวอย่าง

| ข้อมูลเข้า | ผลลัพธ์ |
|-------------------------------|---------|
| 8 1 2 3 4 5 6 7 8 | 1 |
| 12 7 2 5 3 7 0 6 4 1 9 5 8 | 0 |
| 12 6 4 1 9 5 8 7 2 5 3 7 0 | 0 |

หมายเหตุ จำนวนข้อมูลเข้าจะมีไม่เกิน 10,000 ค่า

ปัญหา 4 อนุพันธ์ในอาเรย์ [array_derivative]

การหาอนุพันธ์ในอาเรย์เป็นวิธีที่พบบ่อยในการประมวลผลภาพและสัญญาณไฟฟ้า โดยคำนวณได้จากการนำอาเรย์ช่องที่ i ลบด้วยช่องที่ $i - 1$ เช่น หากอาเรย์มีสามช่อง เราจะเอาช่องที่ 2 ลบด้วยช่องที่ 1 และช่องที่ 1 ลบด้วยช่องที่ 0 ส่วนช่องที่ 0 นั้น เราจะกำหนดให้ค่าเป็นศูนย์ (ค่าพิเศษเนื่องจากไม่มีช่องด้านซ้ายมาทำการลบ)

ยกตัวอย่าง หากอาเรย์นี้มีข้อมูลเป็น 3, 5, และ 4 ดังภาพข้างล่างนี้

| | | |
|---|---|---|
| 3 | 5 | 4 |
|---|---|---|

ผลการหาอนุพันธ์จะเก็บลงในอาเรย์ผลลัพธ์ (AOutput) ทำให้ข้อมูลภายใน Aoutput มีค่าเป็น 0, 2, และ -1 ตามลำดับ ซึ่งช่องข้อมูลหมายเลขหนึ่งได้ผลลัพธ์เป็นสอง มาจาก $5 - 3$ และช่องข้อมูลหมายเลขสองมาจาก $4 - 5 = -1$

จงเขียนโปรแกรมเพื่อรับเลขจำนวนเต็ม N ตัวเก็บไว้ในอาเรย์ โดยที่ $N \leq 100$ จากนั้นให้ส่งอาเรย์ดังกล่าวไปคำนวณผลในฟังก์ชัน

```
void derivative(int* AInput, int* AOutput, const int N);
```

ผลลัพธ์ที่ได้จะถูกเก็บไว้ในอาเรย์ AOutput และจะถูกนำกลับมาแสดงผลในฟังก์ชัน main

ข้อมูลเข้า

- บรรทัดแรกระบุจำนวนเต็ม N ที่ระบุจำนวนข้อมูลในอาเรย์
- บรรทัดที่สองระบุจำนวนเต็ม N ตัวที่เป็นข้อมูลในอาเรย์

ผลลัพธ์ เลขจำนวนเต็มในอาเรย์ผลลัพธ์ที่เก็บค่าอนุพันธ์ของอาเรย์ข้อมูลเข้า เลขแต่ละตัวในอาเรย์ถูกคั่นด้วยช่องว่างหนึ่งช่อง

ตัวอย่าง

| ข้อมูลเข้า | ผลลัพธ์ |
|-----------------------|--------------------|
| 6 1 7 3 5 2 4 | 0 6 -4 2 -3 2 |
| 8 7 8 3 2 -1 0 5 9 | 0 1 -5 -1 -3 1 5 4 |

ปัญหา 5 ฟังก์ชันนับการซ้ำของเลขที่มีค่าสูงสุดในอาร์เรย์ [count_max]

จงเขียนฟังก์ชัน countMax สำหรับหาว่าเลขที่มีค่ามากที่สุดในอาร์เรย์มีค่าซ้ำกันทั้งหมดกี่ค่า (ถ้ามีตัวเดียวถือว่าซ้ำกัน 1 ค่า) โดยฟังก์ชันนี้รับอาร์เรย์เลขจำนวนเต็ม arData ซึ่งมีความยาว N หลังจากนับจำนวนเลขค่ามากที่สุดที่ซ้ำกันแล้ว ฟังก์ชันจะคืนจำนวนครั้งที่ซ้ำกันของเลขค่ามากที่สุด ทั้งนี้ฟังก์ชันที่ต้องการมีแม่แบบดังนี้

```
int countMax(int* arData, int N);
```

ตัวอย่าง

| ข้อมูลเข้า | ผลลัพธ์ |
|-------------------------------|---------|
| 10 1 1 1 7 7 3 5 5 2 4 | 2 |
| 8 7 8 3 2 -1 0 5 9 | 1 |
| 12 1 1 1 7 7 7 1 1 1 7 7 7 | 6 |

อธิบายตัวอย่างที่ 1 เลข 7 เป็นเลขที่มีค่ามากที่สุด และในข้อมูลชุดนี้มีเลข 7 อยู่สองครั้ง ผลลัพธ์จึงเป็นเลข 2

*** เราไม่ได้ต้องการนับเลขที่ค่าซ้ำกันมากที่สุด แต่เราต้องการนับการซ้ำของเลขที่มีค่ามากที่สุด

คำแนะนำ ควรมีลูปชั้นเดียวสองชุด ชุดแรกมีหน้าที่หาเลขที่มีค่าสูงสุดในอาร์เรย์ ส่วนชุดที่สองมีหน้าที่นำค่าสูงสุดที่หาได้ ไปตรวจดูว่ามีเลขที่ซ้ำกับมันกี่ครั้ง จำนวนครั้งที่นับได้คือผลลัพธ์ของโปรแกรมนี้

หมายเหตุ จำนวนข้อมูลเข้าจะมีไม่เกิน 10,000 ค่า

[มีตัวช่วยอยู่ในหน้าถัดไป]

ตัวช่วยข้อ 3: isUnique

```
int isUnique(_____ arData, int N) {
    int p, q;
    for(p = 0; p < _ _ _ _ _ _; ++p) {
        for(q = _ _ _ _ _ _; q < N; ++q) {
            if(arData[p] _ _ _ _ _ arData[q]) {
                return _____;
            }
        }
    }
    return _____;
}
```

ตัวช่วยข้อ 5: countMax

```

int countMax(_____ arData, int N) {
    int _____ = INT_MIN;
    int i;
    for(i = 0; i < N; ++i) {
        if(arData[i] > _____) {
            M = _____;
        }
    }
    int _____ = 0;
    for(i = 0; i < N; ++i) {
        if(arData[i] _ _ _ _ _ _ _ _ _ _ ) {
            ++count;
        }
    }
    return _____;
}

```