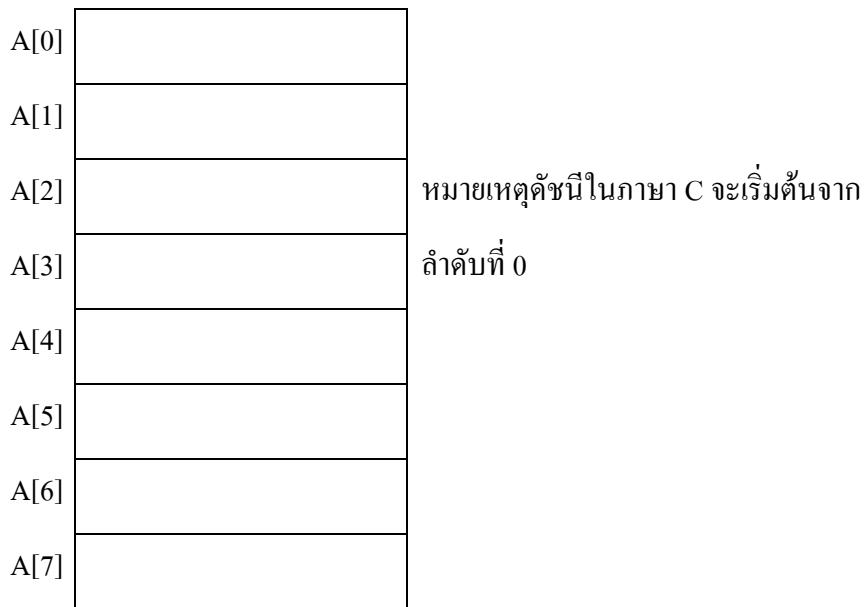


โครงสร้างข้อมูลแบบแคล้มดับ ARRAY

โครงสร้างข้อมูลแบบแคล้มดับจัดเป็นชนิดข้อมูลเชิงนามธรรมแบบเชิงเส้น (Linear Abstract Data Type) และสามารถของข้อมูลในแคล้มดับมีลำดับต่อเนื่องกันเป็นเชิงเส้น นั่นคือมีรูปความสัมพันธ์แบบหนึ่งต่อหนึ่ง การเข้าข้อมูลในโครงสร้างข้อมูลแบบแคล้มดับ (Array) นี้จะเข้าถึงโดยการใช้ดัชนี (Index) ดังตัวอย่างของโครงสร้างแบบแคล้มดับที่มีทั้งหมด 8 หน่วยข้อมูล ซึ่งชนิดข้อมูลของโครงสร้างนี้เป็นประเภทจำนวนเต็ม (integer)

รูปแบบการประกาศตัวแปรสำหรับโครงสร้างดังรูปคือ `int A[8]`

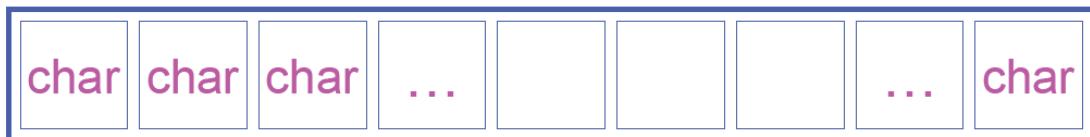


คุณสมบัติที่สำคัญของแคล้มดับ (Array)

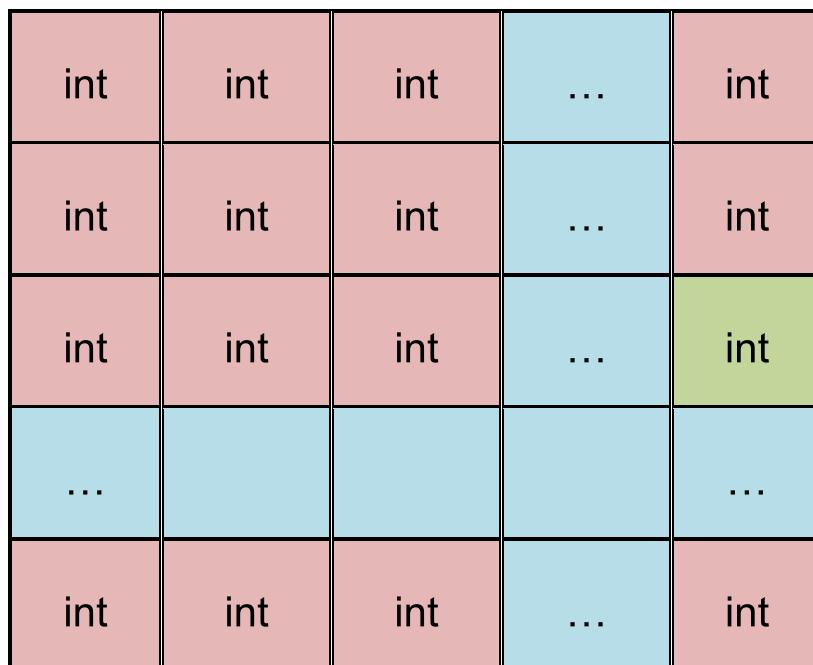
- สามารถในแคล้มดับมีคุณสมบัติเหมือนกัน เช่น เป็นชนิดข้อมูลชนิดเดียวกัน หรืออ้างถึงข้อมูลประเภทเดียวกัน
- ใช้เป็นตัวแทนกลุ่มข้อมูลที่มีความสัมพันธ์กัน เช่น คะแนนสอบของนักศึกษา หรือ จำนวนนักเรียนในแต่ละห้องของชั้นประถมศึกษาที่ 4
- เป็นโครงสร้างข้อมูลที่มีขนาดคงที่ (Static data structure)
- เป็นโครงสร้างข้อมูลที่สามารถอ้างอิง เพื่อเข้าถึงข้อมูลที่ต้องการ ได้ทันทีผ่านการอ้างอิงค่าดัชนี (Index) โดยไม่ต้องผ่านข้อมูลตัวก่อนหน้า ดังนั้นเวลาในการเข้าถึงจะใช้เวลาเพียง $O(1)$

โครงสร้างข้อมูลแบบแคลมดับแบ่งได้ 2 ชนิด

- แคลมดับมิติเดียว (One dimensional array) หรือแคลมดับเชิงเส้น



- แคลมดับหลายมิติ (Multidimensional array)



สิ่งที่เกี่ยวข้องกับโครงสร้างข้อมูลแคลมดับ

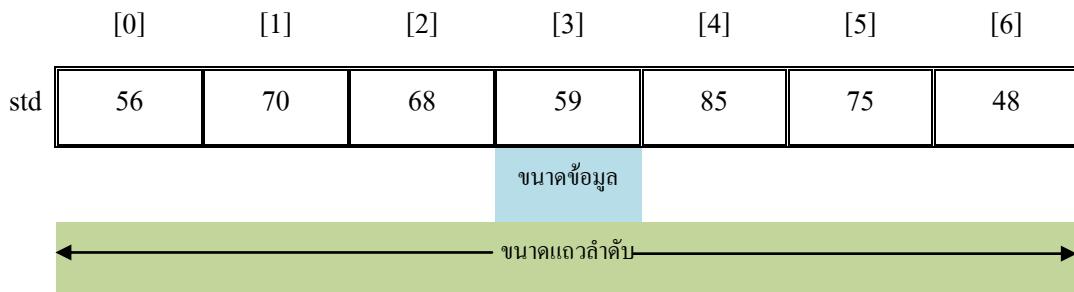
1. ชื่อของแคลมดับ/ชื่อตัวแปร (name)
2. มิติของแคลมดับ (dimension)
3. ขนาดของสมาชิกแต่ละช่อง (size) โดยขนาดนี้มักเขียนอยู่กับชนิดข้อมูลที่ประกาศใช้
4. ดัชนีตัวแรกและดัชนีตัวสุดท้าย (first index and last index) ค่านี้จะเขียนอยู่กับภาษาคอมพิวเตอร์แต่ละภาษา

ตัวอย่าง โครงสร้างข้อมูลແຄວລາດັບສໍາຫຼັບເກີນຂອງມູລຈຳນວນນັກຮຽນໃນແຕ່ລະຫັນຂອງໂຮງຮຽນ ຕັ້ງແຕ່ ປ.ເຕຣີຍມ ແລະ ປ.1 – ປ.6

int std[7] ;

ດັບນີ້ຕົວແຮກຫຼືຄ່າດໍາສຸດ

ດັບນີ້ຕົວສຸດທ້າຍຫຼືຄ່າສູງສຸດ



- ชື່ຂອງແຄວລາດັບ / ชື່ຕົວແປຣ (name) : std
- ມິຕິຂອງແຄວລາດັບ (dimensions) : 1 ມິຕິ
- ຂາດຂອງສາມາຊີກແຕ່ລະຫັນ (size) : 2 ໄບຕີ (ຈຳນວນເຕີມ)
- ຂາດແຄວລາດັບ : 7
- ດັບນີ້ຕົວແຮກແລະດັບນີ້ຕົວສຸດທ້າຍ (first index and last index) : 0 ແລະ 6 ຕາມລາດັບ

ກາຮ້າງອີງຕໍາແໜ່ງສາມາຊີກໃນແຄວລາດັບ

- ກາຮ້າງອີງຕ້ອງເຮີມດ້ວຍໜ້າຂອງແຄວລາດັບ ແລະ ຕ້າຍເລັກດັບນີ້ຕົວໜີ້ລາດັບ
- std[0] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.ເຕຣີຍມ ຊຶ່ງມີຄ່າເທົ່າກັນ 56 ດາວ
- std[1] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.1 ຊຶ່ງມີຄ່າເທົ່າກັນ 70 ດາວ
- std[2] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.2 ຊຶ່ງມີຄ່າເທົ່າກັນ 68 ດາວ
- std[3] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.3 ຊຶ່ງມີຄ່າເທົ່າກັນ 59 ດາວ
- std[4] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.4 ຊຶ່ງມີຄ່າເທົ່າກັນ 85 ດາວ
- std[5] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.5 ຊຶ່ງມີຄ່າເທົ່າກັນ 75 ດາວ
- std[6] ແທນຄ່າຈຳນວນນັກຮຽນໃນໜີ້ ປ.6 ຊຶ່ງມີຄ່າເທົ່າກັນ 48 ດາວ

ขนาดของแผลวลำดับเชิงเส้น (แผลวลำดับหนึ่งมิติ)

- จำนวนสมาชิกในแผลวลำดับ ซึ่งสามารถคำนวณโดยใช้สูตร

$$\text{ArraySize} = \text{Ub} - \text{Lb} + 1$$

- เมื่อ

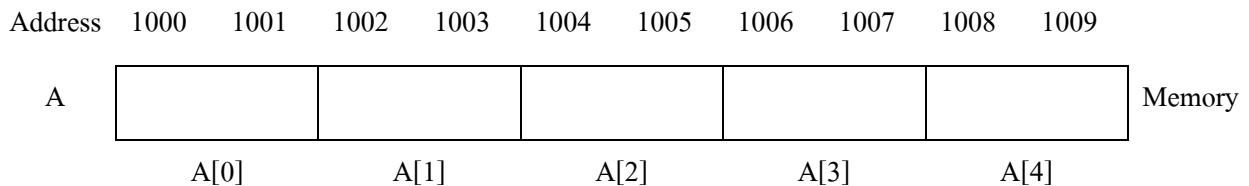
- ❖ ArraySize ก็อ ขนาดของแผลวลำดับเชิงเส้น
- ❖ Ub ก็อ ดัชนีสมาชิกตัวสูงสุด (Upper bound)
- ❖ Lb ก็อ ดัชนีสมาชิกตัวต่ำสุด (Lower bound)

- ตัวอย่าง

- ❖ C language
 - int b[10];
 - $\text{ArraySize} = 9 - 0 + 1 = 10$
- ❖ ADA language
 - Type C is array (-4..5) of integer
 - $\text{ArraySize} = 5 - (-4) + 1 = 10$
- ❖ Pascal language
 - VAR INCOME : array [2531..2550] of real
 - $\text{ArraySize} = 2550 - 2531 + 1 = 20$

โครงสร้างแ阁ล์ดับเบิลเชิงเส้นในหน่วยความจำ

โครงสร้างแบบแกล์ดับเบิลเชิงเส้นนี้จะเป็นการจัดให้เป็นลำดับต่อเนื่องกัน ณ. หมายเลขที่อยู่เริ่มต้น (base address) ที่ตำแหน่งใดตำแหน่งหนึ่งในหน่วยความจำ



- การคำนวณที่ต้องของสมาชิกแกล์ดับเบิลเชิงเส้น สามารถคำนวณโดยใช้สูตรคำนวณดังนี้

$$\text{ArrayAddress}(A[x]) = \text{BaseAddress} + W(x-Lb)$$

หรือ

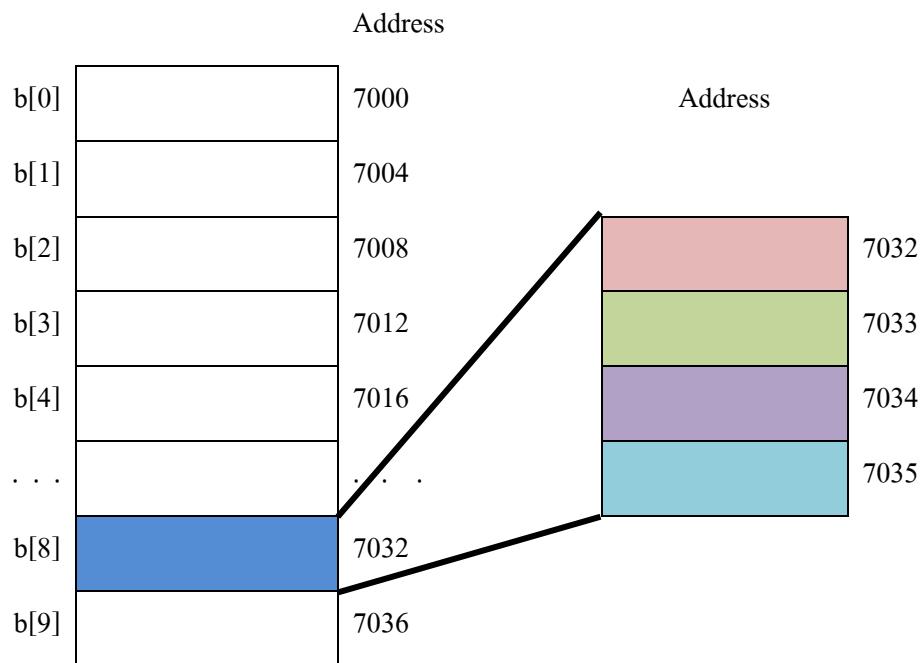
$$\text{ArrayAddress}(A[x]) = \text{BaseAddress} + W(Ub - x)$$

- เมื่อ

- ❖ $\text{ArrayAddress}(A[x])$ คือ ตำแหน่งที่อยู่ของสมาชิกลำดับ x ในแกล์ดับเบิล A
- ❖ BaseAddress คือ เลขที่อยู่เริ่มต้น
- ❖ W คือ ขนาดของข้อมูลในแต่ละช่อง
- ❖ Ub คือ ดัชนีสมาชิกตัวสูงสุด (Upper bound)
- ❖ Lb คือ ดัชนีสมาชิกตัวต่ำสุด (Lower bound)
- ❖ x คือ เลขดัชนีในแกล์ดับเบิล

■ ตัวอย่าง ใน โปรแกรมภาษา C

- ❖ float b[10];
- ❖ กำหนดให้ เลขที่เริ่มต้น คือ 7000 จงหาเลขที่อยู่ของ b[8]
- ❖ Float ใช้เนื้อที่ในการเก็บข้อมูล 4 ไบต์
- ❖ ArrayAddress (b[8]) = $7000 + 4(8 - 0) = 7032$
- ❖ ดังนั้น b[8] จะเก็บที่ address 7032 ในหน่วยความจำ



การใส่ข้อมูลของแคลดับเชิงเส้น

```
void LArrayInput (int *x , int Lb , int Ub)
{
    int i ;
    for ( i = Lb ; i <= Ub ; i++)
    {
        printf ( “ Input data : ” );
        scanf ( “%d” , &x[i]);
    }
}
```

การแสดงผลข้อมูลของแคลดับเชิงเส้น

```
void LArrayOutput (int *x , int Lb , int Ub)
{
    int i ;
    for ( i = Lb ; i <= Ub ; i++)
    {
        printf ( “%d \n” , x[i]);
    }
}
```

การเพิ่มหรือการแทรกข้อมูลลงไปในแคลดับเชิงเส้น

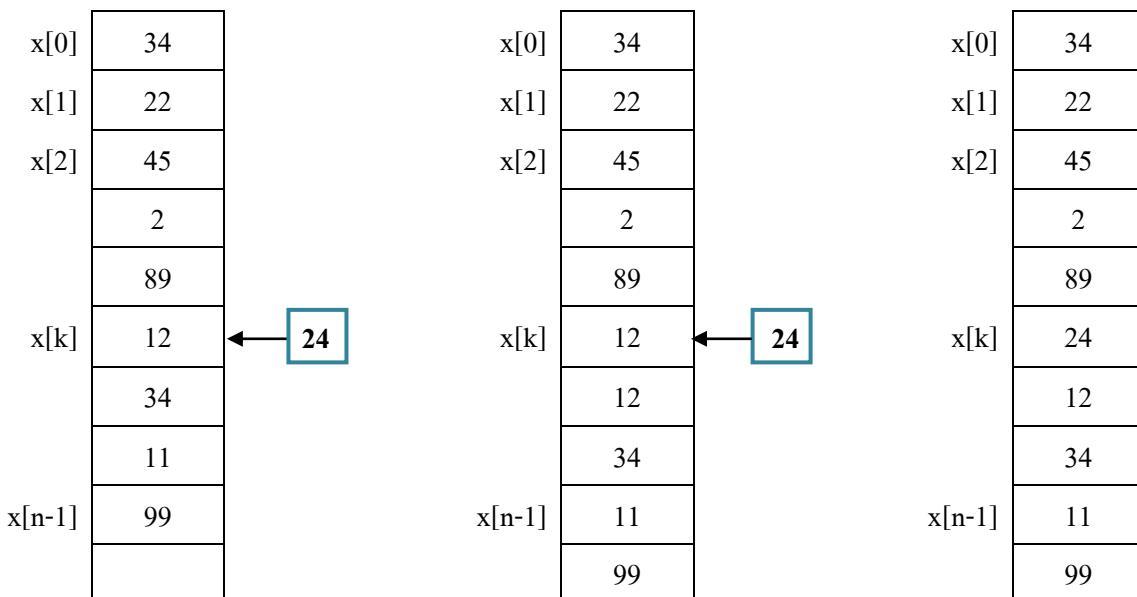
ขั้นตอนในการแทรกข้อมูลลงในแคลดับเชิงเส้น

1. หากตำแหน่งที่ต้องการแทรกข้อมูลมีข้อมูล หรือค่าของตำแหน่งที่ต้องการแทรกมีค่าน้อยกว่าหรือเท่ากับจำนวนข้อมูลทั้งหมด
 - a. ให้วนทำงานซึ่งจะนำตำแหน่งสุดท้ายของข้อมูลไปจับถึงตำแหน่งที่ต้องการแทรกข้อมูล
 - b. ทำการสำเนาข้อมูลที่ตำแหน่งก่อนหน้าไปไว้ที่ตำแหน่งถัดไป
 - c. ลดค่าตำแหน่งที่พิจารณา และวนทำงานตามเงื่อนไขข้อที่ 1
2. แทรกข้อมูลลงในตำแหน่งที่กำหนด
3. เพิ่มค่าจำนวนสมาชิกในแคลดับขึ้นอีก 1

ฟังก์ชันที่ใช้สำหรับการเพิ่มหรือการแทรกข้อมูลลงไปในแคล้มดับซิงส์น

```
void LArrayInsert (int *x , int *n , int k , int data)
{
    int i = *n ;
    while ( i > k )
    {
        x[i] = x[i-1] ;
        i-- ;
    }
    x[k] = data ;
    *n += 1 ;
}
```

ตัวอย่างการทำงาน



อธิบายเพิ่มเติม จากตัวอย่างตัวแปร n คือจำนวนข้อมูลทั้งหมด และตัวแปร k คือตำแหน่งที่จะใส่ข้อมูล สำหรับข้อมูลที่ต้องการเพิ่มหรือแทรกข้อมูลเข้าไปจะเก็บเอาไว้ในตัวแปร $data$ และสุดท้ายจะต้องมีการเพิ่มจำนวนของข้อมูลทั้งหมดด้วย

การลบข้อมูลออกจากแคล้มดับ

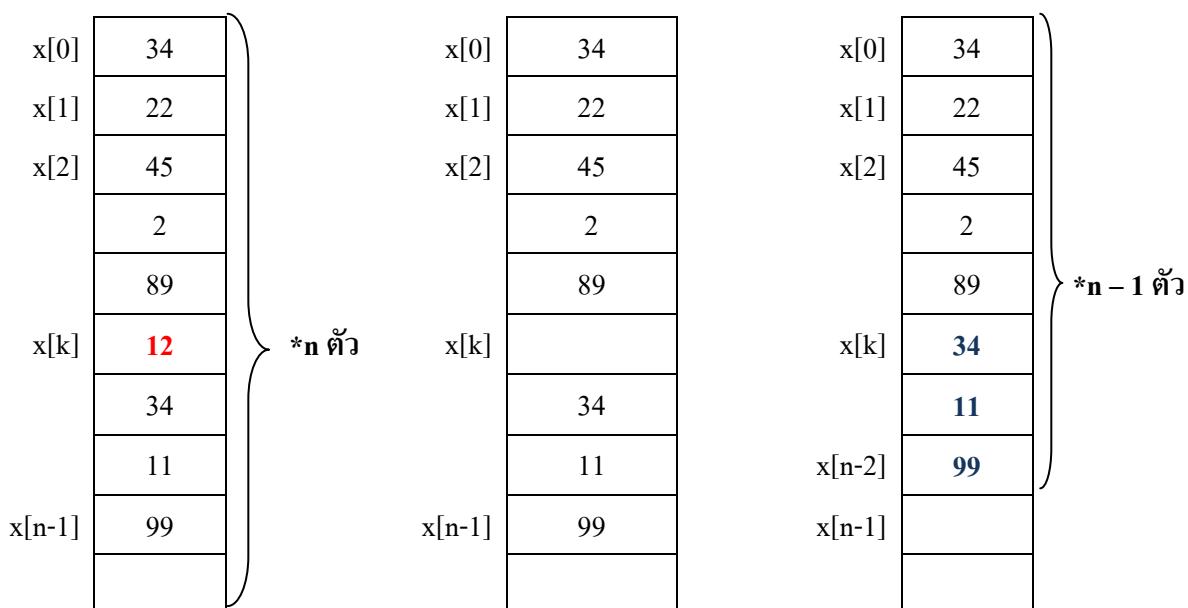
ขั้นตอนในการลบข้อมูลออกจากแคล้มดับเชิงเส้น

1. หากตำแหน่งที่ต้องการลบข้อมูลออกนั้นมีค่าน้อยกว่าจำนวนข้อมูลทั้งหมด ให้เริ่มทำงานตั้งแต่ตำแหน่งที่ต้องการลบข้อมูล
 - a. เสื่อนข้อมูลตัวถัดไปขึ้นมาแทนที่
 - b. เพิ่มค่าตำแหน่งของข้อมูล และวนทำข้อ a จนถึงตำแหน่งของข้อมูลตัวรองสุดท้าย
2. ลดค่าจำนวนข้อมูลทั้งหมดลง 1 ครั้ง

ฟังก์ชันที่ใช้สำหรับการลบข้อมูลออกจากแคล้มดับเชิงเส้น

```
void LArrayDelete (int *x , int *n , int k)
{
    int data = x[k];
    int i;
    for ( i = k ; i < *n-1 ; i++)
    {
        x[i] = x[i+1] ;
    }
    *n -= 1 ;
}
```

ตัวอย่างการทำงาน



แຄວลำดับ 2 มิติ

- เป็นโครงสร้างข้อมูลสำหรับข้อมูลที่อยู่ในรูปแบบของเมตริกซ์ ซึ่งจะประกอบด้วยหลายๆ แถว และแต่ละแถวที่จะมีหลายคอลัมน์ ซึ่งในทางคอมพิวเตอร์เรียกว่า "เมตริกซ์" (matrix)
- เราสามารถมองแຄວลำดับ 2 มิติ เป็นตาราง (table) ของข้อมูลซึ่งประกอบด้วยข้อมูลค่านั้นๆ แถว (row) และข้อมูลทางค้านคอลัมน์ (column)
- สำหรับการอ้างอิงตำแหน่งของสมาชิกในแຄວลำดับแบบ 2 มิตินั้นจะต้องประกอบด้วยข้อมูลดังต่อไปนี้
 - ชื่อของแຄວลำดับ (name)
 - ตัวเลขจำนวนหรือขนาดของมิติ แต่ละมิติ
- เวลาที่ใช้ในการอ้างอิงถึงข้อมูลในแຄວลำดับ 2 มิติก็จะมีค่าคงที่ เช่นเดียวกับแຄວลำดับเชิงเส้นนั้นคือมีค่าเท่ากับ $O(1)$

int	int	int	...	int
int	int	int	...	int
int	int	int	...	int
...				...
int	int	int	...	int

- ขนาดของแฉล์ดับ 2 มิติสามารถคำนวณได้จาก การเอาขนาดของแฉล์ดับในแต่ละมิติมาคูณกัน นั่นเอง

$$\text{ArraySize} = M_1 \times M_2$$

เมื่อ

- ArraySize กือ ขนาดของแฉล์ดับ 2 มิติ
- M₁ กือ ขนาดของแฉล์ดับในมิติที่ 1
- M₂ กือ ขนาดของแฉล์ดับในมิติที่ 2
- เราสามารถที่ประยุกต์เอาการคำนวณขนาดของแฉล์ดับแบบ 2 มิติ มาใช้ในการคำนวณแฉล์ดับแบบหลายมิติได้อีกด้วย ดังนั้นขนาดของแฉล์ดับ n มิติ สามารถคำนวณได้จาก

$$\text{ArraySize} = M_1 \times M_2 \times \dots \times M_n$$

เมื่อ

- ArraySize กือ ขนาดของแฉล์ดับ 2 มิติ
- M₁ กือ ขนาดของแฉล์ดับในมิติที่ 1
- M₂ กือ ขนาดของแฉล์ดับในมิติที่ 2
- M_n กือ ขนาดของแฉล์ดับในมิติที่ n

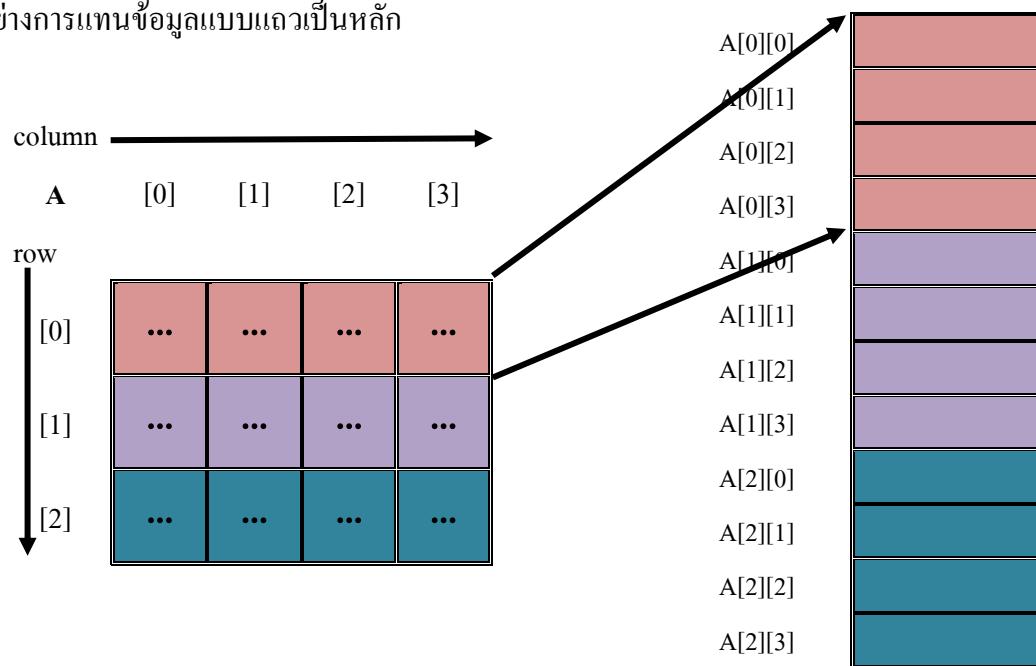
การจัดเก็บข้อมูลแฉล์ดับ 2 มิติในหน่วยความจำ

การจัดเก็บแฉล์ดับ 2 มิติ ในหน่วยความจำของเครื่องคอมพิวเตอร์ ตำแหน่งของหน่วยความจำจะต่อเนื่องกันไป เช่นเดียวกับแฉล์ดับมิติเดียว แต่การจัดเรียงในแฉล์ดับ 2 มิติแบ่งออกเป็น 2 แบบดังนี้

1. การแทนข้อมูลแบบแฉล์เป็นหลัก (Row Major Order Representation) เป็นการจัดเรียงสมาชิกในหน่วยความจำหลักเรียงกันไปทีละ列ๆ โดยเริ่มต้นตั้งแต่แฉล์แรก โดยจะจัดเรียงสมาชิกตามลำดับของคอลัมน์ตั้งแต่คอลัมน์แรกไปจนถึงสมาชิกในคอลัมน์สุดท้ายของแฉล์ที่ 1 จากนั้นก็จะเรียงสมาชิกในแฉล์ที่ 2 ตั้งแต่คอลัมน์แรกถึงคอลัมน์สุดท้ายอีกตามลำดับ และจะวนทำการจัดเก็บข้อมูลเช่นนี้ไปเรื่อยๆ จนครบทุกແล้า โปรแกรม

ภาษาคอมพิวเตอร์ที่มีการจัดเรียงด้วยวิธีนี้ มีหลายภาษาคือยกัน เช่นภาษาโคบล (COBOL language) , ภาษาพีเอลวัน (PL/1 language) และภาษาปาสคาล (Pascal language) เป็นต้น

ตัวอย่างการแทนข้อมูลแบบแطرอเป็นหลัก



การคำนวณเลขที่อยู่ของสมาชิกในแطرอลำดับ กรณีที่เป็นการเก็บข้อมูลแบบแطرอเป็นหลัก สามารถคำนวณได้ตามสูตรคำนวณ ดังนี้

$$\text{ArrayAddress } (A[r][c]) = \text{BaseAddress} + W(nc(r-lbr)+(c-lbc))$$

เมื่อ

- BaseAddress คือ เลขที่อยู่เริ่มต้นของแطرอลำดับ
- W คือ ขนาดของข้อมูลในแต่ละสมาชิกมีหน่วยเป็นไบต์
- nc คือ จำนวนคอลัมน์
- r คือ ดัชนีของแطرอ เมื่อมีค่าเริ่มต้นที่ 0
- c คือ ดัชนีของคอลัมน์ เมื่อมีค่าเริ่มต้นที่ 0
- lbr และ lbc คือค่าตำแหน่งหกของการขอบเขตล่างของโครงสร้างข้อมูล

ฟังก์ชันการใส่ข้อมูลແຕวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามແຄວเป็นหลัก

```
void input2dRMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    for(i=0; i<= nr ; i++)
        for (j = 0 ;j<= nc ;j++)
            scanf("%d", &x[i][j]);
}
```

ฟังก์ชันการแสดงผลข้อมูลແຕวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามແຄວเป็นหลัก

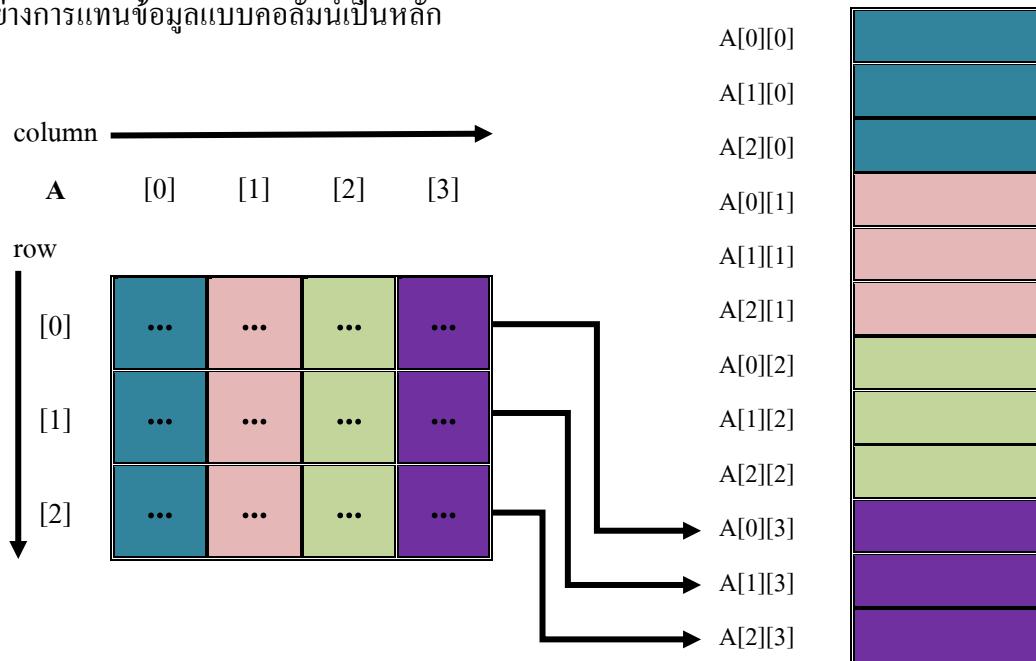
```
void output2dRMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    for(i=0; i<= nr ; i++)
        for (j = 0 ;j<= nc ;j++)
            printf("%d", &x[i][j]);
}
```

ฟังก์ชันการหาผลรวมข้อมูลແຕวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามແຄວเป็นหลัก

```
void sum2dRMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    int sum[nr];
    for(i=0; i<= nr ; i++)
    {
        sum[i] = 0;
        for (j = 0 ;j<= nc ;j++)
            sum[i] = sum[i] + x[i][j] ;
    }
}
```

2. การแทนข้อมูลแบบคอลัมน์เป็นหลัก (Column Major Order Representation) เป็นการจัดเรียงสมาชิกในหน่วยความจำหลักโดยเรียงกันไปทีละคอลัมน์ เริ่มต้นตั้งแต่คอลัมน์แรกจะจัดเรียงสมาชิกจากดาวแรก ตามด้วยสมาชิกในแถวที่ 2 และต่อไปจนถึงสมาชิกดาวสุดท้าย จากนั้นก็ทำการเรียงข้อมูลในคอลัมน์ที่ 2 ต่อไป โดยเรียงจากดาวแรกไปถึงดาวสุดท้าย และทำเช่นนี้ไปเรื่อยจนถึงคอลัมน์สุดท้ายและสุดท้าย โปรแกรมภาษาคอมพิวเตอร์ที่มีการจัดเรียงด้วยวิธีนี้ เช่นภาษาเบสิก (BASIC Language) และภาษาฟอร์TRAN (FORTRAN Language) เป็นต้น

ตัวอย่างการแทนข้อมูลแบบคอลัมน์เป็นหลัก



การคำนวณเลขที่อยู่ของสมาชิกในดาวลำดับ กรณีที่เป็นการเก็บข้อมูลแบบคอลัมน์เป็นหลัก สามารถคำนวณได้ตามสูตรคำนวณ ดังนี้

$$\text{ArrayAddress } (A[r][c]) = \text{BaseAddress} + W((r-lbr)+nr(c-lbc))$$

เมื่อ

- BaseAddress คือ เลขที่อยู่เริ่มต้นของดาวลำดับ
- W คือ ขนาดของข้อมูลในแต่ละสมาชิกมีหน่วยเป็นไบต์
- nr คือ จำนวนดาว
- r คือ ดัชนีของดาว เมื่อมีค่าเริ่มต้นที่ 0
- c คือ ดัชนีของคอลัมน์ เมื่อมีค่าเริ่มต้นที่ 0
- lbr และ lbc คือค่าตำแหน่งหัวทอค่าขอบเขตล่างของโครงสร้างข้อมูล

ฟังก์ชันการใส่ข้อมูลແຕວລາດັບ 2 ມີຕີ ສໍາຫຼັບການເກີນຂໍ້ອມູດຕາມຄອລິມນີ່ເປັນໜັກ

```
void input2dCMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    for(j=0; j<= nc ; i++)
        for (i = 0 ; i<= nr ; j++)
            scanf("%d", &x[i][j]);
}
```

ฟังก์ชันการแสดงผลຂໍ້ອມູດແຕວລາດັບ 2 ມີຕີ ສໍາຫຼັບການເກີນຂໍ້ອມູດຕາມຄອລິມນີ່ເປັນໜັກ

```
void output2dCMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    for(j=0; j<= nc ; i++)
        for (i = 0 ; i<= nr ; j++)
            printf("%d", &x[i][j]);
}
```

ฟังກ්ชනກາຮາພລຣມຂໍ້ອມູດແຕວລາດັບ 2 ມີຕີ ສໍາຫຼັບການເກີນຂໍ້ອມູດຕາມຄອລິມນີ່ເປັນໜັກ

```
void sum2dCMO(int x[][] ,int nr , int nc)
{
    int i ,j;
    int sum[nc];
    for(j=0; j<= nc ; j++)
    {
        sum[j] = 0;
        for (i = 0 ; i<= nr ; i++)
            sum[j] = sum[j] + x[i][j] ;
    }
}
```

แบบฝึกหัด

เมื่อกำหนดให้แฉล้ำลำดับ 2 มิติชุดหนึ่งชื่อ MA มีขนาด 3×4 สามารถเก็บข้อมูลขนาด 4 ไปต์ และมีเลขที่อยู่ริมต้นที่ 1000 โดยที่ดัชนีเริ่มที่ 0

1. จงคำนวณหาเลขที่อยู่ของ MA[1][2] โดยข้อมูลถูกเก็บแบบไส่ข้อมูลแฉล้ำเป็นหลัก
2. จงคำนวณหาเลขที่อยู่ของ MA[2][3] โดยข้อมูลถูกเก็บแบบไส่ข้อมูลคล้มน์เป็นหลัก
3. จงคำนวณขนาดของหน่วยความจำของแฉล้ำลำดับต่อไปนี้
 - 3.1 Class(-10 : 5) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ double
 - 3.2 BB (490 : 510) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ long integer
 - 3.3 IPAD (49 : 65) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ 5 words
4. จงคำนวณแสดงลำดับของดัชนีสำหรับแฉล้ำลำดับชนิด 2 มิติของชุดข้อมูล GG ที่มีขนาด 30×35 สามารถ
 - 4.1 สามารถลำดับที่ 100 เมื่อมีการจัดเก็บข้อมูลแบบแฉล้ำเป็นหลัก
 - 4.2 สามารถลำดับที่ 100 เมื่อมีการจัดเก็บข้อมูลแบบคล้มน์เป็นหลัก