

โครงสร้างข้อมูลแบบแถวลำดับ ARRAY

โครงสร้างข้อมูลแบบแถวลำดับจัดเป็นชนิดข้อมูลเชิงนามธรรมแบบเชิงเส้น (Linear Abstract Data Type) และสมาชิกของข้อมูลในแถวลำดับมีลำดับต่อเนื่องกันเป็นเชิงเส้น นั่นคือมีความสัมพันธ์แบบหนึ่งต่อหนึ่ง การเข้าข้อมูลในโครงสร้างข้อมูลแบบแถวลำดับ (Array) นี้จะเข้าถึงโดยการใช้ดัชนี (Index) ดังตัวอย่างของโครงสร้างแบบแถวลำดับที่มีทั้งหมด 8 หน่วยข้อมูล ซึ่งชนิดข้อมูลของโครงสร้างนี้เป็นประเภทจำนวนเต็ม (integer)

รูปแบบการประกาศตัวแปรสำหรับโครงสร้างดังรูปคือ `int A[8]`

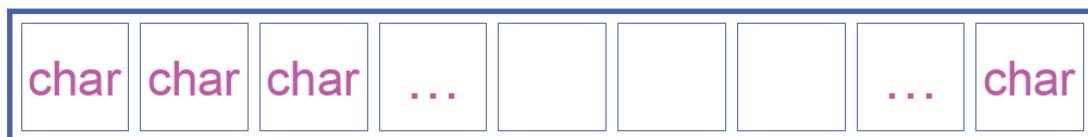
A[0]		หมายเหตุดัชนีในภาษา C จะเริ่มต้นจาก ลำดับที่ 0
A[1]		
A[2]		
A[3]		
A[4]		
A[5]		
A[6]		
A[7]		

คุณสมบัติที่สำคัญของแถวลำดับ (Array)

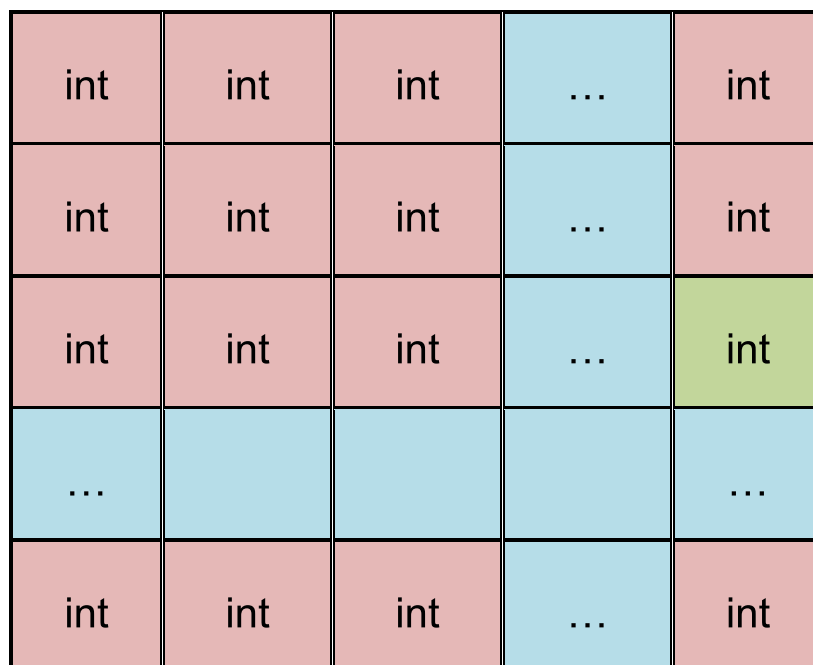
- สมาชิกในแถวลำดับมีคุณสมบัติเหมือนกัน เช่น เป็นชนิดข้อมูลชนิดเดียวกัน หรืออ้างถึงข้อมูลประเภทเดียวกัน
- ใช้เป็นตัวแทนกลุ่มข้อมูลที่มีความสัมพันธ์กัน เช่น คะแนนสอบของนักศึกษา หรือ จำนวนนักเรียนในแต่ละห้องของชั้นประถมศึกษาที่ 4
- เป็นโครงสร้างข้อมูลที่มีขนาดคงที่ (Static data structure)
- เป็นโครงสร้างข้อมูลที่สามารถอ้างอิง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

โครงสร้างข้อมูลแบบแถวลำดับแบ่งได้ 2 ชนิด

- แถวลำดับมิติเดียว (One dimensional array) หรือแถวลำดับเชิงเส้น



- แถวลำดับหลายมิติ (Multidimensional array)



สิ่งที่เกี่ยวข้องกับโครงสร้างข้อมูลแถวลำดับ

1. ชื่อของแถวลำดับ/ ชื่อตัวแปร (name)
2. มิติของแถวลำดับ (dimension)
3. ขนาดของสมาชิกแต่ละช่อง (size) โดยขนาดนี้มักขึ้นอยู่กับชนิดข้อมูลที่ประกาศใช้
4. ดัชนีตัวแรกและดัชนีตัวสุดท้าย (first index and last index) ค่านี้จะขึ้นอยู่กับภาษาคอมพิวเตอร์แต่ละภาษา

ตัวอย่าง โครงสร้างข้อมูลแถวลำดับสำหรับเก็บข้อมูลจำนวนนักเรียนในแต่ละชั้นของโรงเรียน ตั้งแต่ ป.เตรียม และ ป.1 – ป.6

```
int std[7] ;
```

ดัชนีตัวแรกหรือค่าต่ำสุด

ดัชนีตัวสุดท้ายหรือค่าสูงสุด

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
std	56	70	68	59	85	75	48

ขนาดข้อมูล

ขนาดแถวลำดับ

1. ชื่อของแถวลำดับ / ชื่อตัวแปร (name) : std
2. มิติของแถวลำดับ (dimensions) : 1 มิติ
3. ขนาดของสมาชิกแต่ละช่อง (size) : 2 ไบต์ (จำนวนเต็ม)
4. ขนาดแถวลำดับ : 7
5. ดัชนีตัวแรกและดัชนีตัวสุดท้าย (first index and last index) : 0 และ 6 ตามลำดับ

การอ้างอิงตำแหน่งสมาชิกในแถวลำดับ

- การอ้างอิงต้องเริ่มด้วยชื่อของแถวลำดับ และตามด้วยเลขดัชนีตัวชี้ลำดับ
- std[0] แทนค่าจำนวนนักเรียนในชั้น ป.เตรียม ซึ่งมีค่าเท่ากับ 56 คน
- std[1] แทนค่าจำนวนนักเรียนในชั้น ป.1 ซึ่งมีค่าเท่ากับ 70 คน
- std[2] แทนค่าจำนวนนักเรียนในชั้น ป.2 ซึ่งมีค่าเท่ากับ 68 คน
- std[3] แทนค่าจำนวนนักเรียนในชั้น ป.3 ซึ่งมีค่าเท่ากับ 59 คน
- std[4] แทนค่าจำนวนนักเรียนในชั้น ป.4 ซึ่งมีค่าเท่ากับ 85 คน
- std[5] แทนค่าจำนวนนักเรียนในชั้น ป.5 ซึ่งมีค่าเท่ากับ 75 คน
- std[6] แทนค่าจำนวนนักเรียนในชั้น ป.6 ซึ่งมีค่าเท่ากับ 48 คน

ขนาดของแถวลำดับเชิงเส้น (แถวลำดับหนึ่งมิติ)

- จำนวนสมาชิกในแถวลำดับ ซึ่งสามารถคำนวณโดยใช้สูตร

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

- เมื่อ

- ❖ ArraySize คือ ขนาดของแถวลำดับเชิงเส้น
- ❖ Ub คือ ดัชนีสมาชิกตัวสูงสุด (Upper bound)
- ❖ Lb คือ ดัชนีสมาชิกตัวต่ำสุด (Lower bound)

- ตัวอย่าง

- ❖ C language

- `int b[10];`
- $\text{ArraySize} = 9 - 0 + 1 = 10$

- ❖ ADA language

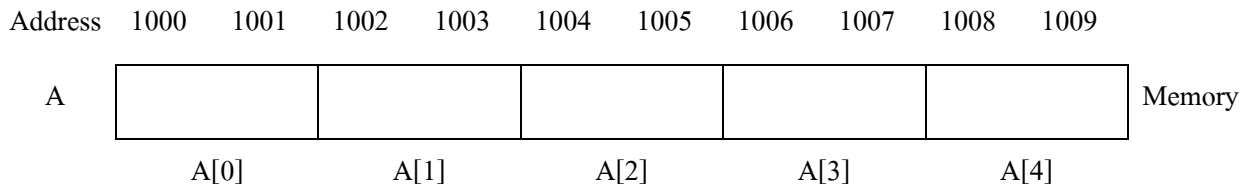
- `Type C is array (-4..5) of integer`
- $\text{ArraySize} = 5 - (-4) + 1 = 10$

- ❖ Pascal language

- `VAR INCOME : array [2531..2550] of real`
- $\text{ArraySize} = 2550 - 2531 + 1 = 20$

โครงสร้างแถวลำดับเชิงเส้นในหน่วยความจำ

โครงสร้างแบบแถวลำดับเชิงเส้นนี้จะเป็นการจัดให้เป็นลำดับต่อเนื่องกัน ณ. หมายเลขที่อยู่เริ่มต้น (base address) ที่ตำแหน่งใดตำแหน่งหนึ่งในหน่วยความจำ



- การคำนวณที่ตั้งของสมาชิกแถวลำดับเชิงเส้น สามารถคำนวณโดยใช้สูตรคำนวณดังนี้

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

หรือ

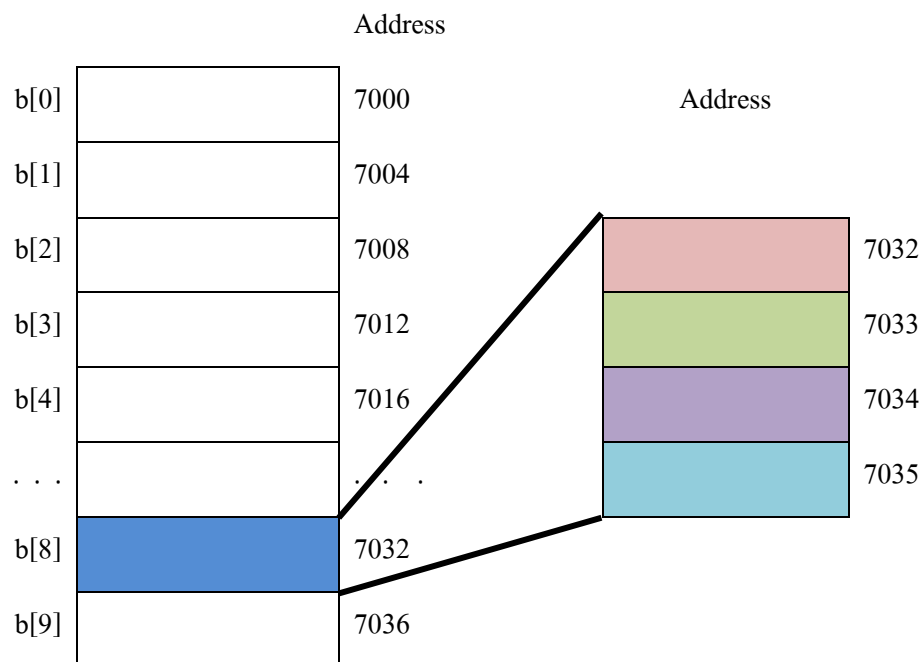
$$\text{ArrayAddress}(A[x]) = \text{BaseAddress} + W(\text{Ub} - x)$$

- เมื่อ

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

■ ตัวอย่าง ใน โปรแกรมภาษา C

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



การใส่ข้อมูลของแถวลำดับเชิงเส้น

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

การแสดงผลข้อมูลของแถวลำดับเชิงเส้น

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

การเพิ่มหรือการแทรกข้อมูลลงไปในแถวลำดับเชิงเส้น

ขั้นตอนในการแทรกข้อมูลลงในแถวลำดับเชิงเส้น

1. หากตำแหน่งที่ต้องการแทรกข้อมูลมีข้อมูล หรือค่าของตำแหน่งที่ต้องการแทรกมีค่าน้อยกว่าหรือเท่ากับจำนวนข้อมูลทั้งหมด
 - a. ให้อวนทำงานซึ่งจากตำแหน่งสุดท้ายของข้อมูลไปจนถึงตำแหน่งที่ต้องการแทรกข้อมูล
 - b. ทำการสับเปลี่ยนข้อมูลที่ตำแหน่งก่อนหน้าไปไว้ที่ตำแหน่งถัดไป
 - c. ลดค่าตำแหน่งที่พิจารณา และวนทำงานตามเงื่อนไขข้อที่ 1
2. แทรกข้อมูลลงในตำแหน่งที่กำหนด
3. เพิ่มค่าจำนวนสมาชิกในแถวลำดับขึ้นอีก 1

ฟังก์ชันที่ใช้สำหรับการเพิ่มหรือการแทรกข้อมูลลงไปในแถวลำดับเชิงเส้น

```
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

x[0]	34
x[1]	22
x[2]	45
	2
	89
x[k]	12
	34
	11
x[n-1]	99

← 24

x[0]	34
x[1]	22
x[2]	45
	2
	89
x[k]	12
	12
	34
x[n-1]	11
	99

← 24

x[0]	34
x[1]	22
x[2]	45
	2
	89
x[k]	24
	12
	34
x[n-1]	11
	99

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

การลบข้อมูลออกจากแถวลำดับ

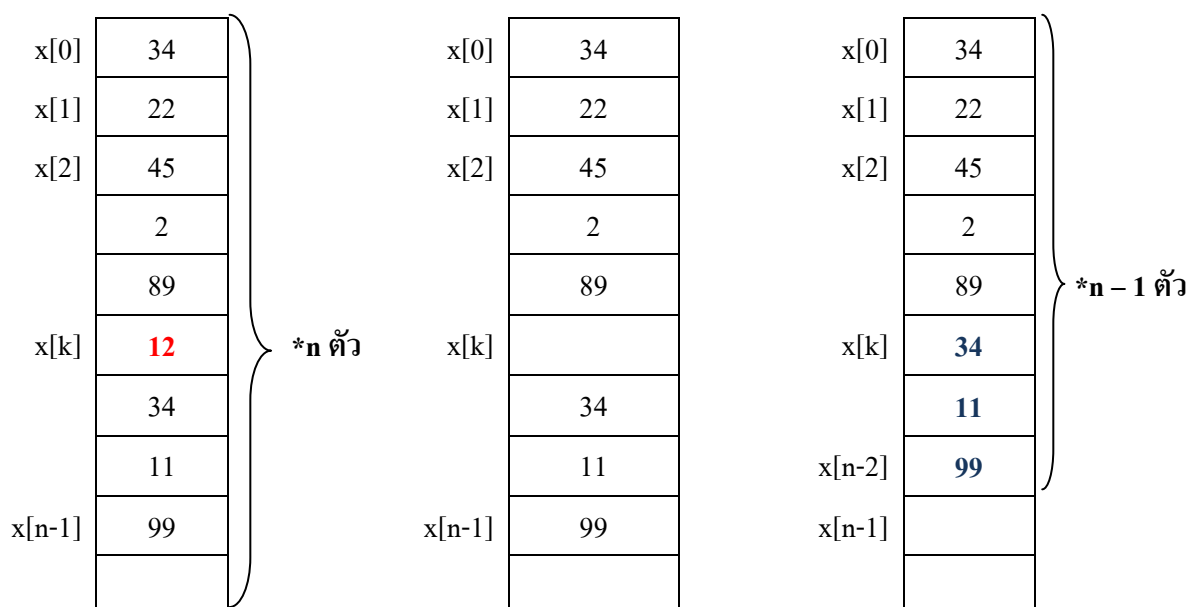
ขั้นตอนในการลบข้อมูลออกจากแถวลำดับเชิงเส้น

1. หากตำแหน่งที่ต้องการลบข้อมูลออกนั้นมีค่าน้อยกว่าจำนวนข้อมูลทั้งหมด ให้เริ่มทำงานตั้งแต่ตำแหน่งที่ต้องการลบข้อมูล
 - a. เลื่อนข้อมูลตัวถัดไปขึ้นมาแทนที่
 - b. เพิ่มค่าตำแหน่งข้อมูล และวนทำข้อ a จนถึงตำแหน่งข้อมูลตัวรองสุดท้าย
2. ลดค่าจำนวนข้อมูลทั้งหมดลง 1 ค่า

ฟังก์ชันที่ใช้สำหรับการลบข้อมูลออกจากแถวลำดับเชิงเส้น

```
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



แถวลำดับ 2 มิติ

- เป็นโครงสร้างข้อมูลสำหรับข้อมูลที่อยู่ในรูปแบบของเมตริกซ์ ซึ่งจะประกอบด้วยหลายๆ แถว และแต่ละแถวก็จะมีหลายคอลัมน์ ซึ่งในทางคณิตศาสตร์ เรียกว่า "เมตริกซ์ (matrix)"
- เราสามารถมองแถวลำดับ 2 มิติ เป็นตาราง (table) ของข้อมูลซึ่งประกอบด้วยข้อมูลด้านแถว (row) และข้อมูลทางด้านคอลัมน์ (column)
- สำหรับการอ้างอิงตำแหน่งของสมาชิกในแถวลำดับแบบ 2 มิตินั้นจะต้องประกอบด้วยข้อมูลดังต่อไปนี้
 - ชื่อของแถวลำดับ (name)
 - ตัวเลขจำนวนหรือขนาดของมิติ แต่ละมิติ
- เวลาที่ใช้ในการอ้างอิงถึงข้อมูลในแถวลำดับ 2 มิติก็จะมีค่าคงที่เช่นเดียวกับแถวลำดับเชิงเส้นนั่นคือมีค่าเท่ากับ $O(1)$

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

- ขนาดของแถวลำดับ 2 มิติสามารถคำนวณได้จาก การเอาขนาดของแถวลำดับในแต่ละมิติมาคูณกันนั่นเอง

$$\text{ArraySize} = M1 \times M2$$

เมื่อ

- ArraySize คือ ขนาดของแถวลำดับ 2 มิติ
- M1 คือ ขนาดของแถวลำดับในมิติที่ 1
- M2 คือ ขนาดของแถวลำดับในมิติที่ 2
- เราสามารถที่ประยุกต์เอาการคำนวณขนาดของแถวลำดับแบบ 2 มิติ มาใช้ในการคำนวณแถวลำดับแบบหลายมิติได้อีกด้วย ดังนั้นขนาดของแถวลำดับ n มิติ สามารถคำนวณได้จาก

$$\text{ArraySize} = M1 \times M2 \times \dots \times Mn$$

เมื่อ

- ArraySize คือ ขนาดของแถวลำดับ 2 มิติ
- M1 คือ ขนาดของแถวลำดับในมิติที่ 1
- M2 คือ ขนาดของแถวลำดับในมิติที่ 2
- Mn คือ ขนาดของแถวลำดับในมิติที่ n

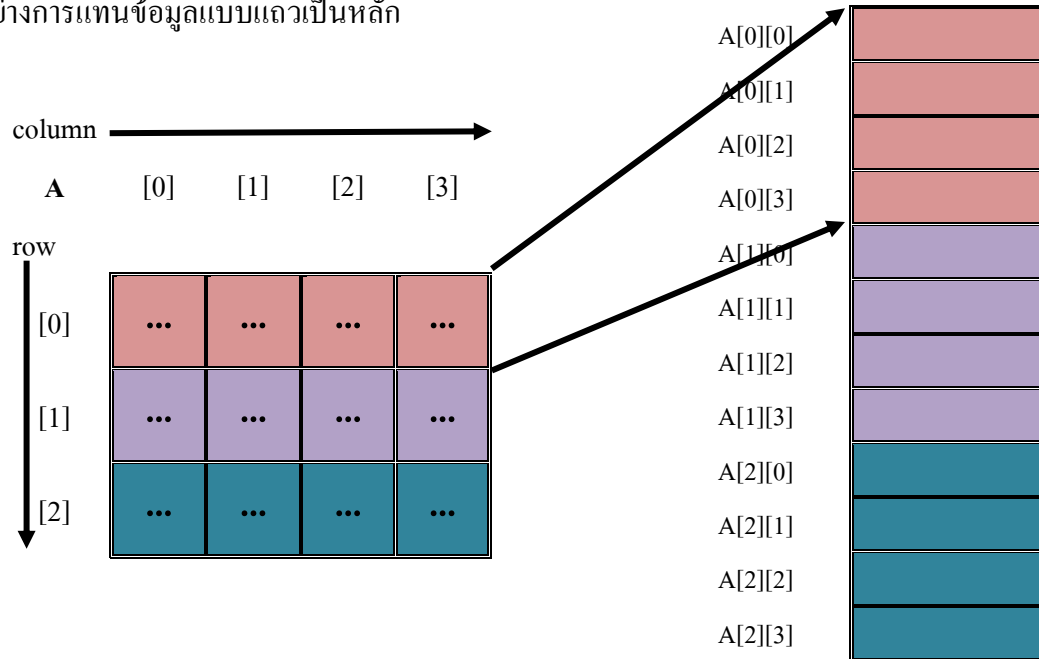
การจัดเก็บข้อมูลแถวลำดับ 2 มิติในหน่วยความจำ

การจัดเก็บแถวลำดับ 2 มิติ ในหน่วยความจำของเครื่องคอมพิวเตอร์ ตำแหน่งของหน่วยความจำจะต่อเนื่องกันไปเช่นเดียวกับแถวลำดับมิติเดียว แต่การจัดเรียงในแถวลำดับ 2 มิติแบ่งออกเป็น 2 แบบดังนี้

1. การแทนข้อมูลแบบแถวเป็นหลัก (Row Major Order Representation) เป็นการจัดเรียงสมาชิกในหน่วยความจำหลักเรียงกันไปทีละแถว โดยเริ่มต้นตั้งแต่แถวแรกโดยจะจัดเรียงสมาชิกตามลำดับคอลัมน์ตั้งแต่คอลัมน์แรกไปจนถึงสมาชิกในคอลัมน์สุดท้ายของแถวที่ 1 จากนั้นก็จะเรียงสมาชิกในแถวที่ 2 ตั้งแต่คอลัมน์แรกถึงคอลัมน์สุดท้ายอีกตามลำดับ และจะวนทำการจัดเก็บข้อมูลเช่นนี้ไปเรื่อยๆ จนครบทุกแถว โปรแกรม

ภาษาคอมพิวเตอร์ที่มีการจัดเรียงด้วยวิธีนี้ มีหลายภาษาด้วยกัน เช่น ภาษาโคบอล (COBOL language) , ภาษาพีแอลวัน (PL/1 language) และภาษาปาสคาล (Pascal language) เป็นต้น

ตัวอย่างการแทนข้อมูลแบบแถวเป็นหลัก



การคำนวณเลขที่อยู่ของสมาชิกในแถวลำดับ กรณีที่เป็นการเก็บข้อมูลแบบแถวเป็นหลัก สามารถคำนวณได้ตามสูตรคำนวณ ต่อไปนี้

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

เมื่อ

- BaseAddress คือ เลขที่อยู่เริ่มต้นของแถวลำดับ
- W คือ ขนาดของข้อมูลในแต่ละสมาชิกมีหน่วยเป็นไบต์
- nc คือ จำนวนคอลัมน์
- r คือ ดัชนีของแถว เมื่อมีค่าเริ่มต้นที่ 0
- c คือ ดัชนีของคอลัมน์ เมื่อมีค่าเริ่มต้นที่ 0
- lbr และ lbc คือค่าต่ำสุดหรือค่าขอบเขตล่างของโครงสร้างข้อมูล

ฟังก์ชันการใส่ข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามแถวเป็นหลัก

```
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]);
}
```

ฟังก์ชันการแสดงผลข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามแถวเป็นหลัก

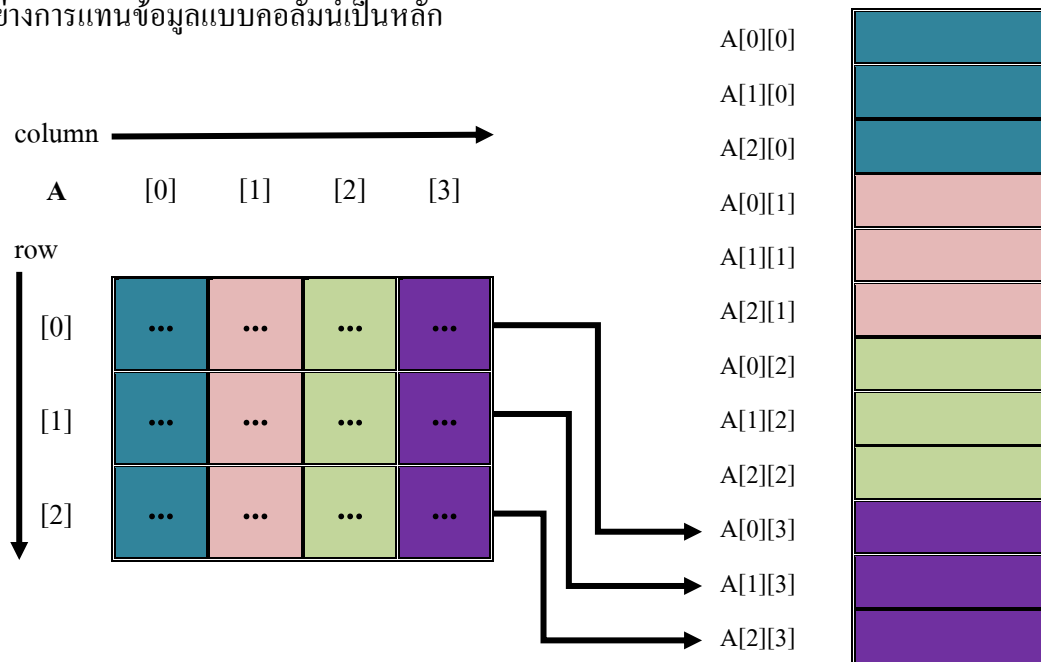
```
void output2dRMO(int x[][] ,int nr , int nc)
{
    int i , j;
    for(i=0; i<= nr ; i++)
        for (j = 0 ; j<= nc ; j++)
            printf("%d", &x[i][j]);
}
```

ฟังก์ชันการหาผลรวมข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามแถวเป็นหลัก

```
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 เริ่มต้นตั้งแต่คอลัมน์แรกจะจัดเรียงสมาชิกจากแถวแรก ตามด้วยสมาชิกในแถวที่ 2 และต่อไปจนถึงสมาชิกแถวสุดท้าย จากนั้นก็ทำการเรียงข้อมูลในคอลัมน์ที่ 2 ต่อไป โดยเรียงจากแถวแรกไปถึงแถวสุดท้าย และทำเช่นนี้ไปเรื่อยจนถึงคอลัมน์สุดท้ายแถวสุดท้าย โปรแกรมภาษาคอมพิวเตอร์ที่มีการจัดเรียงด้วยวิธีนี้ เช่น ภาษาเบสิก (BASIC Language) และภาษาฟอร์แทรน (FORTRAN Language) เป็นต้น

ตัวอย่างการแทนข้อมูลแบบคอลัมน์เป็นหลัก



การคำนวณเลขที่อยู่ของสมาชิกในแถวลำดับ กรณีที่เป็นการเก็บข้อมูลแบบคอลัมน์เป็นหลัก สามารถคำนวณได้ตามสูตรคำนวณ ต่อไปนี้

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

เมื่อ

- BaseAddress คือ เลขที่อยู่เริ่มต้นของแถวลำดับ
- W คือ ขนาดของข้อมูลในแต่ละสมาชิกมีหน่วยเป็นไบต์
- nr คือ จำนวนแถว
- r คือ ดัชนีของแถว เมื่อมีค่าเริ่มต้นที่ 0
- c คือ ดัชนีของคอลัมน์ เมื่อมีค่าเริ่มต้นที่ 0
- lbr และ lbc คือค่าต่ำสุดหรือค่าขอบเขตล่างของโครงสร้างข้อมูล

ฟังก์ชันการใส่ข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามคอลัมน์เป็นหลัก

```
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]);
}
```

ฟังก์ชันการแสดงผลข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามคอลัมน์เป็นหลัก

```
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]);
}
```

ฟังก์ชันการหาผลรวมข้อมูลแถวลำดับ 2 มิติ สำหรับการเก็บข้อมูลตามคอลัมน์เป็นหลัก

```
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

เมื่อกำหนดให้แถวลำดับ 2 มิติชุดหนึ่งชื่อ MA มีขนาด 3 x 4 สมาชิกแต่ละสมาชิกเก็บข้อมูลขนาด 4 ไบต์ และมีเลขที่อยู่เริ่มต้นที่ 1000 โดยที่ดัชนีเริ่มที่ 0

1. จงคำนวณหาเลขที่อยู่ของ MA[1][2] โดยข้อมูลถูกเก็บแบบใส่ข้อมูลแถวเป็นหลัก
2. จงคำนวณหาเลขที่อยู่ของ MA[2][3] โดยข้อมูลถูกเก็บแบบใส่ข้อมูลคอลัมน์เป็นหลัก
3. จงคำนวณขนาดของหน่วยความจำของแถวลำดับต่อไปนี้

3.1 Class(-10 : 5) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ double

3.2 BB (490 : 510) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ long integer

3.3 IPAD (49 : 65) โดยแต่ละช่องมีชนิดข้อมูลเป็นแบบ 5 words

4. จงคำนวณแสดงลำดับของดัชนีสำหรับแถวลำดับชนิด 2 มิติของชุดข้อมูล GG ที่มีขนาด 30 x 35 สมาชิก

4.1 สมาชิกลำดับที่ 100 เมื่อมีการจัดเก็บข้อมูลแบบแถวเป็นหลัก

4.2 สมาชิกลำดับที่ 100 เมื่อมีการจัดเก็บข้อมูลแบบคอลัมน์เป็นหลัก