

## CHAPTER 07

## อาเรย์หนึ่งมิติ (One Dimensional Arrays)

### 1. ความรู้เบื้องต้นเกี่ยวกับอาเรย์ (Introduction to Arrays)

#### 1. นิยามของอาเรย์

- 1) อาเรย์ (Arrays) คือ โครงสร้างข้อมูลประเภทหนึ่งที่ใช้เก็บรายการของข้อมูลประเภทเดียวกัน โดยเก็บข้อมูลทั้งหมดเรียงต่อกันไปในหน่วยความจำ เสมือนนำข้อมูลเก็บไว้ใน "ช่อง" ที่วางเรียงกัน เช่น

6	1	7	9
---	---	---	---

'C'
'U'

ไม่สนใจว่าช่องของอาเรย์จะวางเรียงกันในแนวตั้งหรือแนวนอน ซึ่งจะเรียกว่าอาเรย์ทั้งหมด โดยแต่ละช่องจะเก็บข้อมูลได้เพียง 1 ค่า

- 2) ตำแหน่งของอาเรย์ อาเรย์แต่ละชุดจะมีชื่อของอาเรย์และหมายเลขช่องที่ระบุตำแหน่ง (Index) ของข้อมูลในอาเรย์แต่ละตัว ซึ่งจะเริ่มต้นที่ตำแหน่งที่ 0 (Zero Index) เสมอ (เหมือนกับสตริง)

0	1	2	3
6	1	7	9

ตำแหน่ง (Index) เริ่มที่ 0 แต่ลำดับ (Order) เริ่มที่ 1

- 3) ขนาดหรือความยาวของอาเรย์ คือ จำนวนสมาชิกหรือจำนวนข้อมูลที่เก็บอยู่ในอาเรย์ชุดนั้น

0	1	2	3
6	1	7	9

อาเรย์ยาว 4 (มีสมาชิก 4 ตัว)

- 4) มิติของอาเรย์ สามารถมีได้ไม่จำกัด เช่นอาเรย์หนึ่งมิติ สองมิติ สามมิติ หรือ n มิติ แต่โดยทั่วไปแล้วเรานิยมใช้อาเรย์หนึ่งมิติและอาเรย์สองมิติ ซึ่งในบทนี้จะนำเสนออาเรย์หนึ่งมิติ

6	1
---	---

หนึ่งมิติ

2	1
7	0

สองมิติ

9	4
5	3

สามมิติ

### 2. การประกาศและสร้างอาเรย์หนึ่งมิติ

#### 1. การประกาศและสร้างอาเรย์หนึ่งมิติโดยใช้ Initialized List

การประกาศและสร้างอาเรย์หนึ่งมิติโดยใช้ Initialized List เป็นการสร้างอาเรย์แบบง่าย ซึ่งจะประกาศใช้ได้ก็ต่อเมื่อทราบค่าทุกค่าที่จะเก็บลงไปในอาเรย์แล้ว โดยมีรูปแบบคำสั่งดังนี้

<ประเภทข้อมูล> <ชื่ออาเรย์>[] = { สมาชิก , ..., สมาชิก };

หลังชื่ออาเรย์ต้องมีสัญลักษณ์ [ ]

สมาชิกแต่ละตัวคั่นด้วย Comma

เช่น `int a[] = {34, 56, 52, 12, 90, 0, 75, 23, 45, 8};` ซึ่งจะได้เป็นโครงสร้างดังนี้

0	1	2	3	4	5	6	7	8	9
34	56	52	12	90	0	75	23	45	8

มีสมาชิก 10 ตัว แต่ตำแหน่งสูงสุดคือ 9

ค่าสมาชิกแต่ละตัวของอาร์เรย์สามารถอ้างถึงหรือเรียกชื่อได้โดยใช้คำสั่ง <ชื่ออาร์เรย์>[<ตำแหน่ง>] เช่น

0	1	2	3	4	5	6	7	8	9
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

**โจทย์ข้อที่ 1 [ระดับง่าย]** จงหาผลลัพธ์จากส่วนของโปรแกรมต่อไปนี้ โดยกำหนดให้โปรแกรมสามารถทำงานจนจบได้แม้มี Error พร้อมทั้งแสดงผลลัพธ์สุดท้ายของสมาชิกในอาร์เรย์แต่ละช่อง (8 คะแนน)

```

1  int i = 3;
2  int a[] = { 0, 2, 4, 6, 8, 1, 3, 5, 7, 9 };
3  System.out.println(a[0] + a[6]);
4  System.out.println(a[1] * --a[4]);
5  System.out.println(a[9]++);
6  System.out.println(a[10] + a[4]);
7  System.out.println(a[1]++ + a[0]);
8  System.out.println(a[5 + 3]);
9  System.out.println(--a[2*i] + a[1]++);
10 System.out.println(a[1] + a[4] + a[9]);

```

ค่าของสมาชิกแต่ละช่องคือ

0	1	2	3	4	5	6	7	8	9

## 2. การประกาศและสร้างอาร์เรย์หนึ่งมิติโดยการ new

การประกาศและการสร้างอาร์เรย์โดยการ new นี้จะเป็นวิธีแบบทั่วไปที่ใช้ในการเขียนโปรแกรมภาษาจาวา ซึ่งมีขั้นตอนดังต่อไปนี้

### 1) การประกาศตัวแปรอาร์เรย์หนึ่งมิติ

<ประเภทข้อมูล> <ชื่ออาร์เรย์>[];

หลังชื่ออาร์เรย์ต้องมีเครื่องหมาย [ ]

เช่น int a[];

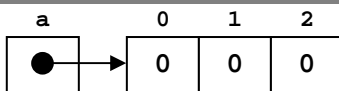


### 2) การสร้างอาร์เรย์หนึ่งมิติ

<ชื่ออาร์เรย์> = new <ประเภทข้อมูล>[<ขนาด>];

หลังชื่ออาร์เรย์ไม่ต้องมีเครื่องหมาย [ ]

เช่น a = new int[3];



ระบุขนาดหรือจำนวนช่องของอาร์เรย์ (ไม่ใช่ตำแหน่ง)

### 3) อาร์เรย์หนึ่งมิติที่ได้หลังจากการประกาศและการสร้าง

a[0]	a[1]	a[2]
------	------	------

มีค่าเริ่มต้นของทุกช่องเป็น 0 (ประเภท int)

### 4) การประกาศและการสร้างอาร์เรย์หนึ่งมิติโดยรวมขั้นตอนที่ 1 และ 2 เข้าด้วยกัน (มี 2 วิธีให้เลือกใช้)

<ประเภทข้อมูล> <ชื่ออาร์เรย์>[] = new <ประเภทข้อมูล>[<ขนาด>];

<ประเภทข้อมูล> []<ชื่ออาร์เรย์> = new <ประเภทข้อมูล>[<ขนาด>];

เครื่องหมาย [ ] จะอยู่หน้าหรือหลังชื่ออาร์เรย์ก็ได้

เช่น **int** a[] = new **int**[3];

## 5) ข้อสังเกตในการประกาศและสร้างอาร์เรย์หนึ่งมิติ

- ขนาดของอาร์เรย์ถูกระบุตอนสร้างอาร์เรย์ ถ้าต้องการให้อาร์เรย์มีที่ช่องก็ให้ระบุจำนวนช่องลงไป ซึ่งส่วนนี้ไม่ใช่ตำแหน่งของอาร์เรย์ (อย่าสับสน) → ลำดับของอาร์เรย์เริ่มนับที่ 1 แต่ตำแหน่งของอาร์เรย์เริ่มนับที่ 0
- ขนาดของอาร์เรย์เป็นค่าของนิพจน์ได้ เช่น `int data[] = new int[3 * n + 1];` เป็นต้น
- อาร์เรย์ที่สร้างขึ้น ไม่สามารถเพิ่มหรือลดขนาดได้ (สร้างแล้วต้องใช้อย่างเดิมตลอดไป)
- ระบบจะตั้งค่าเริ่มต้นของข้อมูลให้กับอาร์เรย์ทุกช่องโดยอัตโนมัติเมื่อเริ่มสร้างอาร์เรย์ตามชนิดของอาร์เรย์ เช่น `double a[] = new double[5];` จะได้อาร์เรย์ 5 ช่องโดยมีค่าเริ่มต้นของทุกช่องเป็น 0.0

## 3. การหาขนาดและความยาวของอาร์เรย์หนึ่งมิติ

การหาขนาดของอาร์เรย์จะใช้คำสั่ง `.length` โดยระบุเข้าไปหลังชื่ออาร์เรย์โดยไม่ต้องใส่วงเล็บ ดังรูปแบบต่อไปนี้

`<ชื่ออาร์เรย์>.length`

อย่าจำสับสนกับเมทอดที่ใช้หาความยาวของสตริงซึ่งใช้ `.length()` เช่นเดียวกัน ดังตัวอย่าง

```
1 int a[] = new int[5];
2 System.out.println(a.length);
3 String s = "JAVA!";
4 System.out.println(s.length());
5 int b[] = new int[12];
6 System.out.println(b.length());
```

หลัง length ไม่ใส่วงเล็บ

หลัง length ใส่วงเล็บ

Error

โจทย์ข้อที่ 2 [ระดับง่าย] จงเขียนคำสั่งภาษาจาวาเพื่อประกาศและสร้างอาร์เรย์ต่อไปนี้ (12 คะแนน)

- 1) สร้างอาร์เรย์ชื่อ `num` เก็บข้อมูลตัวเลขจำนวนเต็มที่ยาว 15 หลักขึ้นไป จำนวน 200 ค่า (2 คะแนน)
- 2) สร้างอาร์เรย์ชื่อ `dice` เก็บหมายเลขหน้าของลูกเต๋า (2 คะแนน)
- 3) สร้างอาร์เรย์ชื่อ `avgGrade` เก็บผลการเรียนเฉลี่ยของนิสิตจำนวน 451 คน (2 คะแนน)
- 4) สร้างอาร์เรย์ชื่อ `grade` เก็บผลการเรียนรายวิชา 2110101 ของนิสิตจำนวน 369 คน (2 คะแนน)
- 5) สร้างอาร์เรย์ชื่อ `x` เก็บค่าความจริง ซึ่งมีตำแหน่งสูงสุดของช่องสมาชิกคือตำแหน่งที่ 10 (2 คะแนน)
- 6) สร้างอาร์เรย์ชื่อ `merge` เก็บข้อมูลที่ได้จากการนำอาร์เรย์ในข้อที่ 1 มาต่อกับอาร์เรย์ในข้อที่ 2 (2 คะแนน)

#### 4. การอ้างอิงค่า การกำหนดค่า และการแสดงผลของอาเรย์หนึ่งมิติ

##### 1) การอ้างอิงค่าจากอาเรย์ มีรูปแบบคำสั่งดังนี้

```
<ประเภทข้อมูล> <ชื่อตัวแปร> = <ชื่ออาเรย์>[<ตำแหน่ง>];
```

เช่น `int n = num[0];` (ให้ตัวแปร `n` ประเภทจำนวนเต็ม เก็บค่าจากอาเรย์ `num` ตำแหน่งที่ 0)

##### 2) การกำหนดค่าลงไปในอาเรย์ มีรูปแบบคำสั่งดังนี้

```
<ชื่ออาเรย์>[<ตำแหน่ง>] = <ค่าข้อมูล>;
```

เช่น `num[0] = 13;` (ให้อาเรย์ `num` ตำแหน่งที่ 0 มีค่าเท่ากับ 13)

##### 3) การแสดงผลของอาเรย์ จะใช้คำสั่ง `for` ในการเข้าไปยังสมาชิกแต่ละตัวในอาเรย์ตั้งแต่ตัวแรก (ตำแหน่งที่ 0) จนถึงตัวสุดท้าย (ตำแหน่งที่ `length - 1`) ดังตัวอย่างต่อไปนี้

```
1 int a[] = new int[5];
2 for(int i = 0; i < a.length; i++) {
3     a[i] = 100;
4 }
```

ต้องเขียน **a.length** ไม่ควรเขียนเป็นตัว  
เลข และต้องเป็นเครื่องหมาย < ไม่ใช่ <=

i ต้องเริ่มที่ 0 ไม่ใช่ 1 (วนตำแหน่งของอาเรย์ ไม่ใช่วนลำดับ)

โจทย์ข้อที่ 3 [ระดับง่าย] จงเขียนคำสั่งภาษาจาวาเพื่ออ้างอิงและกำหนดค่าจากอาเรย์ต่อไปนี้ โดยสมมติให้อาเรย์ในทุกข้อมีการประกาศและสร้างเรียบร้อยแล้วพร้อมทั้งมีค่าข้อมูลครบทุกช่อง (12 คะแนน)

1) ประกาศตัวแปร `x` เพื่อเก็บข้อมูลประเภทจำนวนเต็มที่ได้จากอาเรย์ `num` ตัวที่ 50 (2 คะแนน)

2) ประกาศตัวแปร `y` เพื่อเก็บข้อมูลประเภทจำนวนเต็มที่ได้จากอาเรย์ `num` ตำแหน่งที่ 50 (2 คะแนน)

3) ประกาศตัวแปร `c` เพื่อเก็บข้อมูลประเภทอักขระที่ได้จากอาเรย์ `code` ตำแหน่งสุดท้าย (2 คะแนน)

4) ให้ค่าสมาชิกตำแหน่งที่ 12 ของอาเรย์ `bank` เก็บข้อมูลประเภทตรรกะจากตัวแปร `var1` (2 คะแนน)  
`boolean var1 = 5 > 6;`

5) ให้ค่าสมาชิกตัวที่ 1 ของอาเรย์ `bank` เก็บข้อมูลประเภทจำนวนจริงจากตัวแปร `var2` (2 คะแนน)  
`double var2 = kb.nextDouble();`

6) ให้ค่าสมาชิกตัวรองสุดท้ายของอาเรย์ `bank` เก็บข้อมูลประเภทอักขระจากตัวแปร `var3` (2 คะแนน)  
`char var3 = "java".charAt(3);`

โจทย์ข้อที่ 4 [ระดับง่าย-ปานกลาง] จงเขียนส่วนของโปรแกรมภาษาจาวาตามคำสั่งต่อไปนี้ (55 คะแนน)

- 1) สร้างอาร์เรย์ชื่อ  $n$  แบบ Initialed List เพื่อเก็บเลขคี่ที่อยู่ในช่วง 1 ถึง 20 พร้อมทั้งแสดงค่าของตัวเลขทุกตัวในอาร์เรย์  $n$  ขึ้นบนจอภาพทีละบรรทัด (5 คะแนน)

- 2) สร้างอาร์เรย์ชื่อ  $m$  ขนาด 1000 ช่อง พร้อมทั้งกำหนดค่าให้กับสมาชิกทุกตัวในอาร์เรย์  $m$  ให้มีค่าเป็นตัวเลขตั้งแต่ 1-1000 เรียงลำดับจากสมาชิกตัวแรกจนถึงสมาชิกตัวสุดท้ายของอาร์เรย์  $m$  (5 คะแนน)

- 3) สร้างอาร์เรย์ชื่อ  $a$  เพื่อเก็บตัวเลขจำนวนจริงใดๆ ที่รับมาจากแป้นพิมพ์จำนวน 100 ตัว (5 คะแนน)
- ```
Scanner kb = new Scanner(System.in);
```

- 4) นับจำนวนเลขคี่และคู่ที่เก็บอยู่ในอาร์เรย์  $x$  ว่ามีอย่างละกี่จำนวนแล้วแสดงผลขึ้นบนจอภาพ (10 คะแนน)
- ```
int x[] = { 7, 9, -1, 4, 12, 9, 3, 2, -7, 2, 1, 9, -15, 24, 11, 13 };
```

- 5) รับตัวเลขจำนวนเต็มจากแป้นพิมพ์เก็บไว้ในตัวแปร `x` แล้วค้นหาว่า ค่าสมาชิกตำแหน่งแรกในอาร์เรย์ `num` ที่มีค่าเท่า `x` คือตำแหน่งใด พร้อมทั้งแสดงผลลัพธ์ขึ้นบนจอภาพ (10 คะแนน)

```
int num[] = { 3, 2, 1, 10, 2, 8, 3, 2, 1, 1, 8, 5, 10, 11, 7, 6, 10 };
Scanner kb = new Scanner(System.in);
```

ถ้าค้นหาไม่เจอให้แสดงผลลัพธ์เป็นค่า -1

โจทย์ข้อที่ 5 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อทำการผสาน (Merge) ข้อมูลในอาร์เรย์ `a` และอาร์เรย์ `b` ซึ่งเป็นจำนวนเต็มเข้าเป็นอาร์เรย์เดียวกันในอาร์เรย์ชื่อ `ab` พร้อมทั้งแสดงค่าสมาชิกของอาร์เรย์ `ab` (10 คะแนน)

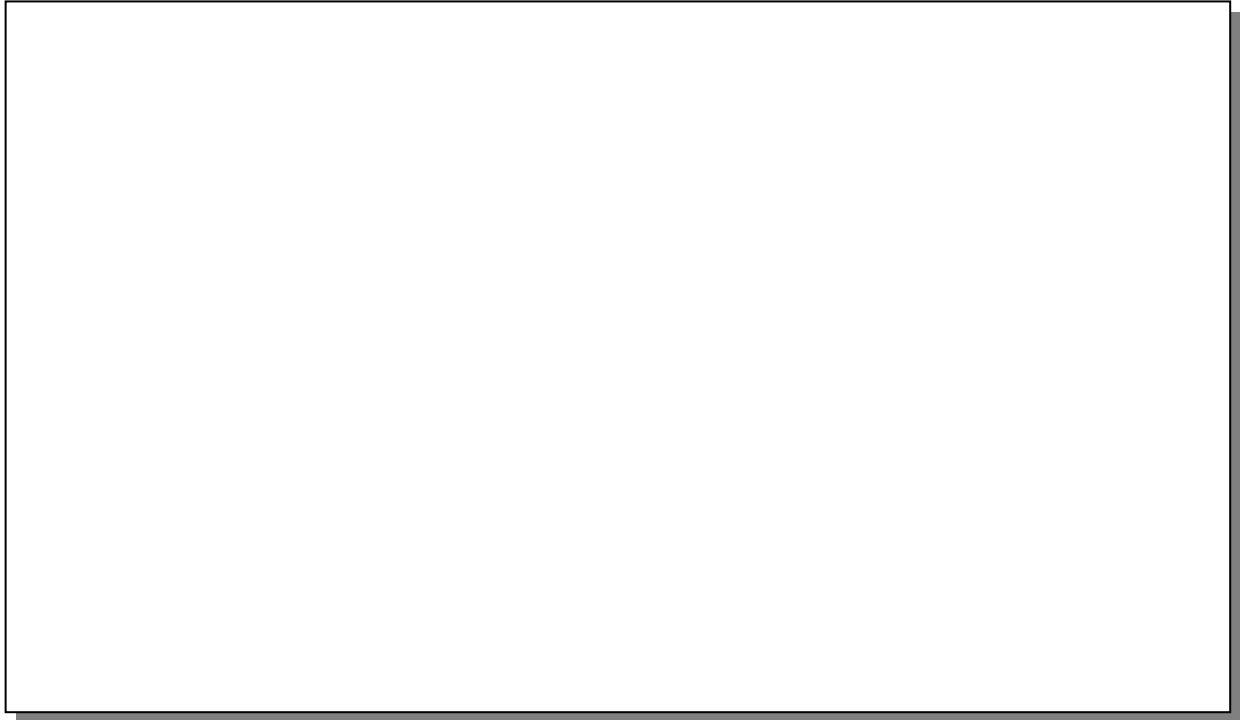
```
import java.util.Scanner;
public class MergeArrays {
    public static void main(String[] args) {
        int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
        int b[] = { 1, 4, 6, 8, 10, 12 };
```

```
    } //End of main
} //End of class
```

โจทย์ข้อที่ 6 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อรับตัวเลขจำนวนเต็มจากแป้นพิมพ์เก็บไว้ในตัวแปร `x` แล้วทำการแบ่ง (Split) ค่าสมาชิกในอาร์เรย์ `num` ออกเป็นอาร์เรย์ใหม่ โดยถ้าค่าสมาชิกมีค่ามากกว่าหรือเท่ากับ `x` ให้เก็บไว้ในอาร์เรย์ `upper` ไม่เช่นนั้นให้เก็บไว้ในอาร์เรย์ `lower` (10 คะแนน)

```
import java.util.Scanner;
public class SplitArray {
    public static void main(String[] args) {
```

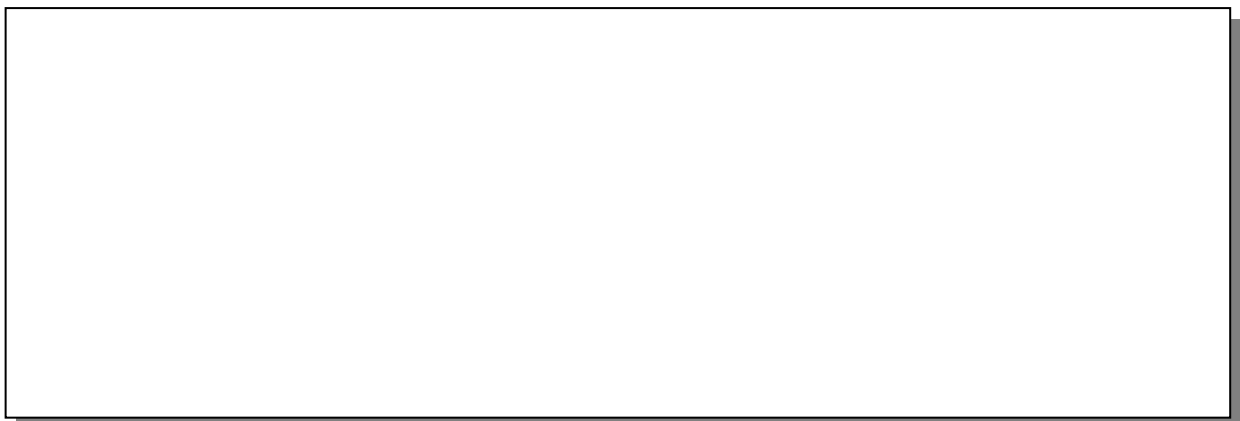
```
int num[] = {95, 1, 6, 34, 5, 9, 123, -2, 57, 82, 12, 79, 45, 34, 1};
```



```
} //End of main
} //End of class
```

**โจทย์ข้อที่ 7 [ระดับยาก]** จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อทำการกลับค่าสมาชิกในอาเรย์ (Reverse Array) ที่ชื่อ num จากหลังมาหน้า แล้วเก็บค่าไว้ในอาเรย์ตัวเดิม โดยในการเขียนโปรแกรมห้ามประกาศอาเรย์ตัวใหม่เพิ่มโดยเด็ดขาด (10 คะแนน)

```
import java.util.Scanner;
public class ReverseArray {
    public static void main(String[] args) {
        int num[] = {95, 1, 6, 34, 5, 9, 123, -2, 57, 82, 12, 79, 45, 34, 1};
```



```
} //End of main
} //End of class
```

**โจทย์ข้อที่ 8 [ระดับยาก]** จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อตรวจสอบและแสดงข้อความว่า อาเรย์ a และอาเรย์ b เท่ากันหรือไม่ โดยการเท่ากันของอาเรย์คือ ความยาวของอาเรย์ต้องเท่ากัน และ สมาชิกทุกตัวตำแหน่งต่อตำแหน่งต้องเท่ากัน (10 คะแนน)

```
import java.util.Scanner;
public class EqualityOfArrays {
    public static void main(String[] args) {
        int a[] = { 1, 2, 3, 4, 5, 6, 7 }, b[] = { 1, 2, 3, 5, 5, 7, 7 };
```

```
    } //End of main
} //End of class
```

โจทย์ข้อที่ 9 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อทำการรับค่าตัวเลขจำนวนเต็มจากแป้นพิมพ์เก็บไว้ในตัวแปร key เพื่อค้นหาว่าค่าของตัวแปร key ปรากฏเป็นลำดับรองสุดท้ายที่ตำแหน่งใดในอาร์เรย์ num โดยถ้าค้นเจอให้แสดงตำแหน่งนั้น แต่ถ้าไม่เจอให้แสดงค่า -1 (10 คะแนน)

```
import java.util.Scanner;
public class LastSecondSearch {
    public static void main(String[] args) {
        int num[] = { 3, 2, 1, 10, 2, 8, 3, 2, 1, 1, 8, 5, 10, 11, 7, 6, 10 };
```

```
    } //End of main
} //End of class
```



### 3. การประยุกต์ใช้อาร์เรย์ (Applications of Array)

#### 1. การเรียงลำดับแบบเลือก (Selection Sort) (ดูการสาธิตตัวอย่างการจัดเรียงในโจทย์ข้อที่ 10)

```

1  for (int i = data.length - 1; i >= 1; i--) {
2      int maxIndex = 0;
3      for (int j = 0; j <= i; j++)
4          if (data[j] > data[maxIndex]) maxIndex = j;
5      int temp = data[i];
6      data[i] = data[maxIndex];
7      data[maxIndex] = temp;
8  }

```

จำ

เรียงจากน้อยไปหามาก

เลือกค่าที่สนใจในแต่ละรอบ ซึ่งอาจจะเป็นค่ามากที่สุดหรือค่าน้อยที่สุดในรอบนั้น เพื่อจัดเรียง

- คำสั่ง `for i` ชั้นนอก เป็นการกำหนดช่องหรือเลือกช่องตำแหน่งที่ `i` ที่ต้องการจะเรียงลำดับ
- คำสั่ง `for j` ชั้นใน เป็นการหาค่าสมาชิกที่สูงที่สุดตั้งแต่ช่องแรกจนถึงช่องตำแหน่งที่ `i` ที่เลือกไว้ เพื่อจะได้สลับค่าสูงสุดไปไว้ในช่องตำแหน่งที่ `i` นั้น

#### 2. การเรียงลำดับแบบฟอง (Bubble Sort) (ดูการสาธิตตัวอย่างการจัดเรียงในโจทย์ข้อที่ 10)

```

1  for (int i = data.length - 1; i >= 1; i--) {
2      for (int j = 0; j < i; j++) {
3          if (data[j] > data[j + 1]) {
4              int temp = data[j];
5              data[j] = data[j + 1];
6              data[j + 1] = temp;
7          }
8      }
9  }

```

จำ

เรียงจากน้อยไปหามาก

จับคู่และขยับค่าที่สนใจในแต่ละรอบ อาจจะเป็นค่ามากหรือค่าน้อยของคู่ นั้น เพื่อจัดเรียง

- คำสั่ง `for i` ชั้นนอก เป็นการกำหนดช่องตำแหน่งที่ `i` ที่ต้องการจะเรียงลำดับ
- คำสั่ง `for j` ชั้นใน เป็นการเปรียบเทียบค่าเป็นคู่ๆ เริ่มตั้งแต่ช่องแรกจนถึงช่องตำแหน่งที่ `i` ที่กำหนดไว้ เพื่อจะขยับค่าหรือดันค่าสมาชิกที่สูงสุดไปไว้ในช่องตำแหน่งที่ `i` นั้น (เหมือนฟองสบู่ที่ลอยขึ้นไปเรื่อยๆ)

**โจทย์ข้อที่ 10 [ระดับง่าย]** จงแสดงรายละเอียดในแต่ละรอบของการจัดเรียงแบบเลือก (Selection Sort) และรายละเอียดของรอบที่ 1 และรอบที่ 2 ของการจัดเรียงแบบฟอง (Bubble Sort) จากชุดข้อมูลในอาร์เรย์ `x` ดังนี้ให้ถูกต้อง โดยให้จัดเรียงข้อมูลจากน้อยไปมาก (20 คะแนน)

```
int x[] = {30, 45, 23, 2, 19, 14, 5, 99, 9};
```

30	45	23	2	19	14	5	99	9
----	----	----	---	----	----	---	----	---

#### การจัดเรียงแบบเลือก

รอบที่ 1

--	--	--	--	--	--	--	--	--

รอบที่ 2

--	--	--	--	--	--	--	--	--

รอบที่ 3

--	--	--	--	--	--	--	--	--

รอบที่ 4

--	--	--	--	--	--	--	--	--

รอบที่ 5

--	--	--	--	--	--	--	--	--

รอบที่ 6

--	--	--	--	--	--	--	--	--

รอบที่ 7

--	--	--	--	--	--	--	--	--

รอบที่ 8

--	--	--	--	--	--	--	--	--

**การจัดเรียงแบบฟอง**

รอบที่ 1

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

รอบที่ 2

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--


### 3. การสืบค้นแบบทวิภาค (Binary Search)

เป็นการค้นข้อมูลโดยการแบ่งครึ่งข้อมูลในอาร์เรย์ แล้วพิจารณาเพื่อตัดข้อมูลออกไปครึ่งหนึ่ง (ครึ่งซ้ายหรือครึ่งขวาของอาร์เรย์) โดยข้อมูลต้องจัดเรียงก่อนเสมอ ซึ่งมีอัลกอริทึมในการสืบค้นดังนี้ (กำหนดให้อาร์เรย์ชื่อ data)

```

1  int left = 0, right = data.length - 1, mid = 0;
2  while (left <= right) {
3      mid = (left + right) / 2;
4      if (key == data[mid]) {
5          break;
6      } else if (key < data[mid]) {
7          right = mid - 1;
8      } else {
9          left = mid + 1;
10     }
11 }

```

left คือ ตำแหน่งทางซ้ายของชุดข้อมูล  
right คือ ตำแหน่งทางขวาของชุดข้อมูล  
mid คือ ตำแหน่งตรงกลางของชุดข้อมูล

คำตอบจากการสืบค้นคือค่าของ mid

**โจทย์ข้อที่ 11 [ระดับง่าย]** จงเขียนลำดับการทำงานและนับจำนวนรอบจากการสืบค้นข้อมูลแบบทวิภาคของข้อมูลที่เก็บอยู่ในอาร์เรย์ x ต่อไปนี้ เพื่อค้นหาค่าว่า 14 อยู่ ณ ตำแหน่งใดในอาร์เรย์ชุดนี้ และให้เปรียบเทียบคำตอบที่ได้จากการสืบค้นนี้ กับการสืบค้นแบบลำดับ (10 คะแนน)

```
int x[] = {30, 45, 23, 2, 19, 14, 5, 99, 9};
```

30	45	23	2	19	14	5	99	9
----	----	----	---	----	----	---	----	---

**โจทย์ข้อที่ 12 [ระดับยาก]** จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่ออ่านแฟ้มข้อมูลวันเกิดที่ชื่อ `birthdate.txt` ซึ่งเก็บข้อมูลในรูปแบบ `dd-mm-yyyy` เช่น `13-06-1987` เป็นต้น โดยมีข้อมูลวันเกิดของคนทั้งหมด 200 คน จงทำการจัดเรียงเดือนที่มีคนเกิดมากที่สุดจนถึงเดือนที่มีคนเกิดน้อยที่สุด แล้วแสดงผลลัพธ์ที่เป็นชื่อเดือนทั้ง 12 เดือนที่ถูกจัดเรียงแล้วขึ้นบนจอภาพตามลำดับ (10 คะแนน)

**birthdate.txt**

```
13-02-2000
01-10-1989
31-01-1999
21-12-1995
17-07-1994
...
```

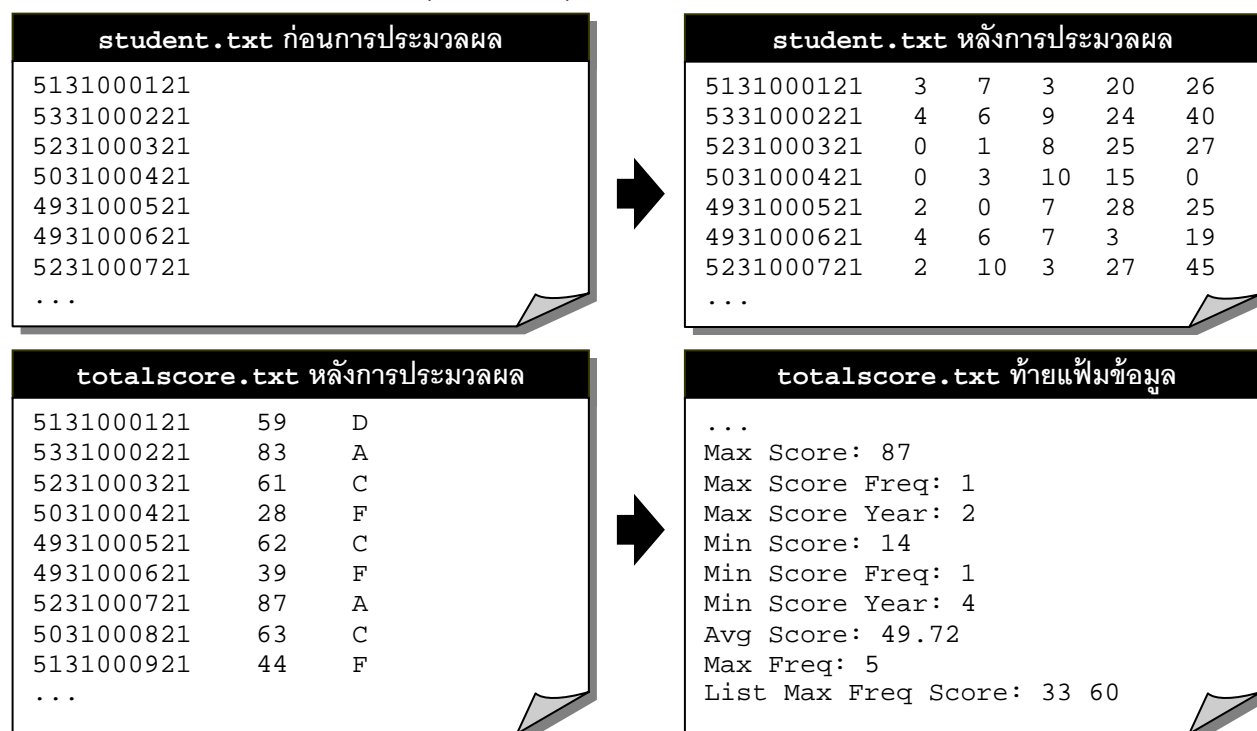
**ตัวอย่างการแสดงผลบนจอภาพ**

```
September [25 people]
March [22 people]
January [21 people]
May [19 people]
June [19 people]
...
```

```
import java.util.Scanner;
import java.io.*;
public class SortMonthByBirthDate {
    public static void main(String[] args) throws IOException {
```

```
    } //End of main
} //End of class
```

โจทย์ข้อที่ 13 [ระดับเทพ] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อใช้สำหรับประมวลผลคะแนนรายวิชา “How To Be Noob” โดยโปรแกรมจะเริ่มอ่านเลขประจำตัวนิสิตจากแฟ้มข้อมูล `student.txt` แล้วทำการสร้างคะแนนของนิสิตแต่ละคนขึ้นมาโดยการสุ่ม ซึ่งประกอบไปด้วย 5 ส่วนได้แก่ คะแนนสอบย่อย (5%) คะแนนการบ้าน (10%) คะแนนปฏิบัติการ (10%) คะแนนสอบกลางภาค (30%) และคะแนนสอบปลายภาค (45%) แล้วนำกลับไปบันทึกไว้ในแฟ้มข้อมูล `student.txt` อีกครั้ง จากนั้นให้อ่านค่าคะแนนทั้ง 5 ส่วนของนิสิตแต่ละคนจากแฟ้มข้อมูลเดิมเพื่อนำมาคำนวณหาค่าคะแนนรวม (100%) พร้อมทั้งตัดเกรดรายวิชานั้น แล้วนำผลลัพธ์ที่ได้ไปบันทึกไว้ในแฟ้มข้อมูล `totalscore.txt` พร้อมทั้งบันทึกข้อมูลสรุปต่าง ๆ ไว้ท้ายแฟ้มข้อมูลนั้นด้วย โดยให้เขียนโปรแกรมตามขั้นตอนต่าง ๆ ที่โจทย์กำหนดไว้ดังรายละเอียดต่อไปนี้ (85 คะแนน)



```
import java.io.*;
import java.util.Scanner;
public class ScoreCalculator {
    public static void main(String[] args) throws IOException {
        //สร้างตัวอ่านชื่อ in1 เพื่อเปิดอ่านแฟ้มข้อมูล student.txt สำหรับนับจำนวนนิสิตทั้งหมดที่
        //ปรากฏในแฟ้มนั้น แล้วเก็บผลลัพธ์ไว้ในตัวแปร count (5 คะแนน)
```

//สร้างตัวอ่านชื่อ in2 เพื่อเปิดอ่านแฟ้มข้อมูล student.txt อีกครั้ง และสร้างอาร์เรย์ชื่อ id แล้วทำการอ่านค่าเลขประจำตัวนิสิตแต่ละคนเก็บไว้ในอาร์เรย์ id ทีละช่อง (5 คะแนน)

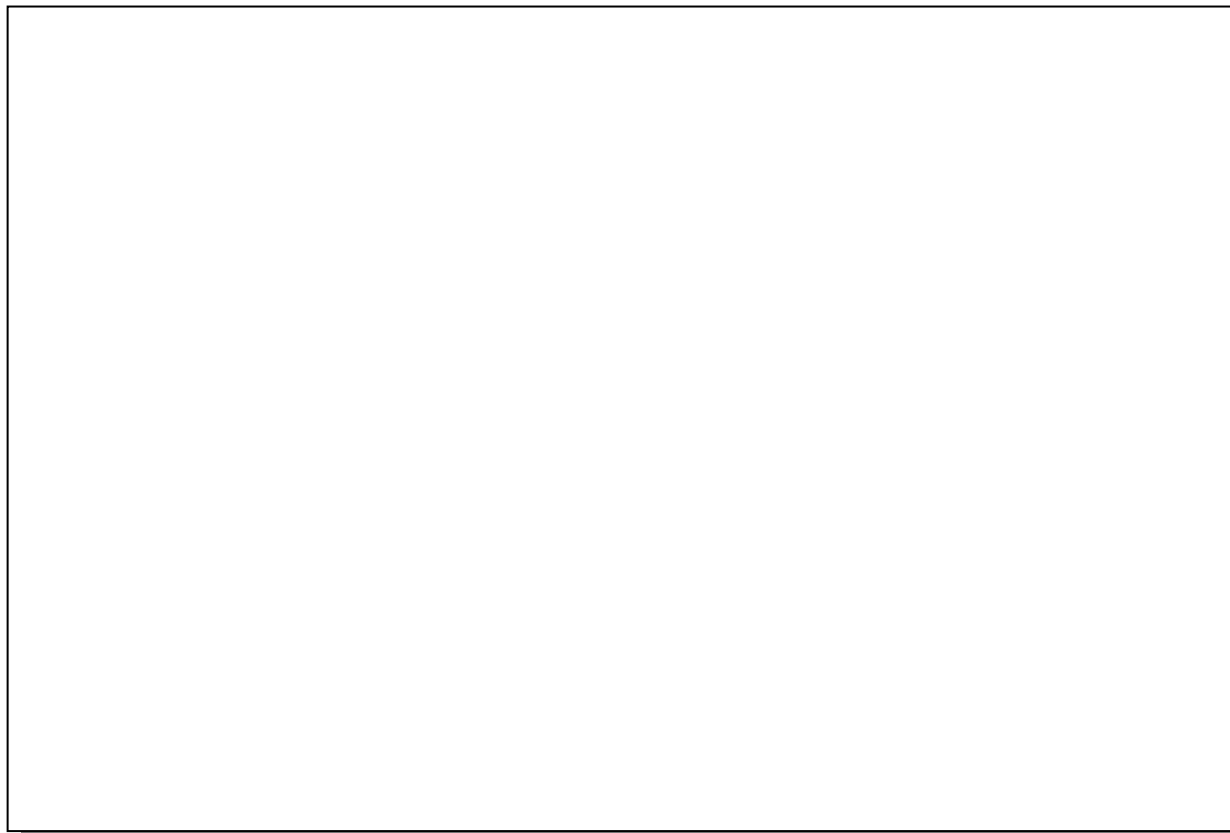
//สร้างตัวเขียนชื่อ out1 เพื่อเขียนแฟ้มข้อมูล student.txt โดยในแต่ละบรรทัดจะประกอบไปด้วยเลขประจำตัวนิสิต ตามด้วยข้อมูลคะแนนทั้ง 5 ส่วนของของนิสิตที่ได้จากการสุ่มตัวเลขจำนวนเต็มตามขอบเขตดังต่อไปนี้ คะแนนสอบย่อย (สุ่มตัวเลข 0-5) คะแนนการบ้าน (สุ่มตัวเลข 0-10) คะแนนปฏิบัติการ (สุ่มตัวเลข 0-10) คะแนนสอบกลางภาค (สุ่มตัวเลข 0-30) และคะแนนสอบปลายภาค (สุ่มตัวเลข 0-45) (10 คะแนน)

//สร้างตัวอ่านชื่อ in3 เพื่อเปิดอ่านแฟ้มข้อมูล student.txt อีกครั้ง และสร้างอาร์เรย์ชื่อ sc แล้วทำการคำนวณคะแนนรวมของนิสิตแต่ละคนเก็บไว้ในอาร์เรย์ sc ทีละช่อง (10 คะแนน)


//สร้างตัวเขียนชื่อ out2 เพื่อเขียนแฟ้มข้อมูล totalscore.txt โดยในแต่ละบรรทัดจะประกอบไปด้วยเลขประจำตัวนิสิต คะแนนรวม และเกรดของนิสิตแต่ละคน โดยเกรดมี 5 ระดับได้แก่ A (ช่วงคะแนน 80-100) B (ช่วงคะแนน 70-79) C (ช่วงคะแนน 60-69) D (ช่วงคะแนน 50-59) และ F (ช่วงคะแนน 0-49) (10 คะแนน)

//หาคะแนนรวมของนิสิตที่ได้คะแนนสูงสุด (Max Score) และต่ำสุด (Min Score)  
//หาจำนวนนิสิตที่ได้คะแนนสูงสุด (Max Score Freq) และต่ำสุด (Min Score Freq) มีอย่างละกี่คน  
//หาว่านิสิตผู้ที่ได้คะแนนสูงสุดและต่ำสุดเป็นนิสิตชั้นปีใด (ถ้ามีหลายชั้นปีให้แสดงชั้นปีใดก็ได้)  
//หาค่าเฉลี่ยของคะแนนรวมของนิสิตทั้งหมด (Average Score)  
//หาความถี่สูงสุด (Max Freq) ของคะแนน  
//หาว่ามีคะแนนใดบ้างที่มีนิสิตได้กันมากที่สุด (List Max Freq Score) (15 คะแนน)

//แสดงผลลัพธ์ที่ได้จากการประมวลผลทั้งหมดจากขั้นตอนก่อนหน้านี้ ต่อท้ายแฟ้มข้อมูล  
totalscore.txt (10 คะแนน)



//คำสั่งในการปิดแฟ้มข้อมูล (5 คะแนน)



```
} //End of main  
} //End of class
```