

# CHAPTER06

อาร์เรย์ (Array)

# บทนำ

- ❑ ตัวแปรส่วนใหญ่ที่กำหนดขึ้นมาใช้งานนั้นจะถูกเก็บไว้ในหน่วยความจำที่ไม่ต่อเนื่องกัน และเป็นตัวแปรที่แทนข้อมูลต่างชุดหรือต่างกลุ่มกัน
- ❑ หากต้องการทำงานกับข้อมูลในชุดเดียวกันซึ่งมีหลายค่า การกำหนดตัวแปรจำนวนมากเพื่อแทนค่าข้อมูลเหล่านั้นเป็นการไม่เหมาะสม
- ❑ เราสามารถทำงานในลักษณะดังกล่าวได้ด้วย อาเรย์ (**array**)

# ข้อมูลแบบอาเรย์

- ❑ อาเรย์ (**array**) คือ ชุดของตัวแปรที่แสดงอยู่ในรูปของลำดับที่ เพื่อใช้สำหรับเก็บค่าของข้อมูลที่อยู่ในกลุ่มเดียวกัน หรือชนิดเดียวกัน
- ❑ เช่น ค่าข้อมูลคะแนนสอบของนักเรียน **40** คน จัดว่าเป็นข้อมูลในกลุ่มเดียวกัน และชนิดเดียวกัน กล่าวคืออยู่ในกลุ่มของคะแนน ซึ่งมีชนิดเป็นตัวเลขเหมือนกัน เป็นต้น
- ❑ ตัวแปรอาเรย์จะแตกต่างไปจากตัวแปรทั่วไป กล่าวคือ ตัวแปรทั่วไปจะถูกเก็บอยู่ในหน่วยความจำแต่ละตำแหน่งไม่ต่อเนื่องกัน

## ข้อมูลแบบอาเรย์ (ต่อ)

- ❑ ตัวแปรอาเรย์ จะถูกเก็บอยู่ในหน่วยความจำในตำแหน่งที่ต่อเนื่องกัน ซึ่งมีขนาดเล็กหรือขนาดใหญ่ขึ้นอยู่กับจำนวนมิติ (**Dimesion**) และจำนวนสมาชิกที่อยู่ในอาเรย์

	score[0]	score[1]	score[2]	score[3]	score[4]	score[5]
score	20	12	60	71	40	32

ภาพจำลองการเก็บข้อมูลไว้ในอาเรย์

## ข้อมูลแบบอาเรย์ (ต่อ)

- ❑ จากภาพจำลองการเก็บข้อมูลในชุดเดียวกัน คือ ชุดข้อมูลคะแนนสอบ
- ❑ ประกอบไปด้วยสมาชิกข้อมูลทั้งหมด 6 จำนวน (20, 12, 60, 71, 40, 32)
- ❑ การเก็บข้อมูลไว้ในอาเรย์จะคล้ายการเก็บข้อมูลไว้ในตารางแต่ละช่องเรียงลำดับกันต่อเนื่องกัน แต่ละช่องของตาราง คือ ตำแหน่งที่ใช้เก็บข้อมูลในอาเรย์
- ❑ โดยมีตัวชี้ “ดัชนีตำแหน่ง (index)” กำกับไว้เริ่มต้นจาก ดัชนี 0 เป็นต้นไป

## ข้อมูลแบบอาเรย์ (ต่อ)

- ❑ ถ้าต้องการนำข้อมูลคะแนนสอบของนักเรียนคนที่ **2** มาใช้ ต้องอ้างอิงตัวชี้ตำแหน่งของตัวแปร **score** ที่มีดัชนีเป็นเลข **1** คือ **score[1]**
- ❑ ถ้าขนาดของอาเรย์มีค่าเท่ากับ **n** ตัวชี้ตำแหน่งสุดท้ายของอาเรย์นี้จะมีค่าเป็น **n-1** นั่นคือ **score[n-1]** นั่นเอง

# การประกาศตัวแปรอาเรย์

- การประกาศตัวแปรอาเรย์เป็นขั้นตอนที่สำคัญที่ควรดำเนินการก่อนการใช้งานตัวแปรอาเรย์นั้น เพื่อให้ตัวแปลภาษารู้จักชื่อตัวแปรอาเรย์ ชนิดข้อมูลของตัวแปรอาเรย์ และขนาดของอาเรย์ เพื่อการจองพื้นที่ในหน่วยความจำของเครื่องได้อย่างถูกต้อง การประกาศตัวแปรอาเรย์สามารถเขียนได้ 2 รูปแบบ ดังนี้

# การประกาศตัวแปรอาเรย์ (ต่อ)

- รูปแบบที่ 1 ประกาศตัวแปรแล้วจึงสร้างอาเรย์

```
dataType [] arrayName;  
arrayName = new dataType[n];
```

หรือ

```
dataType arrayName[];  
arrayName = new dataType[n];
```

- โดยที่ **n** คือ ตัวเลขที่ใช้กำหนดขนาดของอาเรย์, **dataType** คือ ชนิดข้อมูลของอาเรย์ และ **arrayName** คือ ตัวแปรอาเรย์



## การประกาศตัวแปรอาเรย์ (ต่อ)

- รูปแบบที่ 2 ประกาศตัวแปรไปพร้อมๆ กับการสร้างอาเรย์

```
dataType [] arrayName = new dataType[n];
```

หรือ

```
dataType arrayName [] = new dataType[n];
```

- ตัวอย่างการประกาศตัวแปรอาเรย์ เพื่อเก็บคะแนนสอบของนักเรียน 10 คน

```
int score[] = new int[10];
```

# การกำหนดค่าให้แก่ตัวแปรอาเรย์

- การกำหนดค่าให้แก่อาเรย์ ก็คือ การใส่ค่าข้อมูลลงไปในตัวแปรอาเรย์ ซึ่งในภาษา **java** สามารถเขียนประโยคคำสั่งดังกล่าวได้หลายรูปแบบ ดังนี้
- รูปแบบที่ 1 ประกาศและกำหนดข้อมูลทุกค่าลงในอาเรย์โดยใช้เพียงคำสั่งเดียว ตามรูปแบบ ดังนี้

```
dataType [] arrayName = {..., ..., ..., ...};
```

## การกำหนดค่าให้แก่ตัวแปรอาเรย์ (ต่อ)

- เป็นการเขียนประโยคคำสั่งเพียงประโยคเดียวเพื่อประกาศ พร้อมกับการกำหนดค่าข้อมูลให้กับตัวแปรอาเรย์ เช่นเดียวกับการกำหนดค่าเริ่มต้นให้กับตัวแปรทั่วไป เช่น

```
5 String days[] = {"Sunday", "Monday", "Tuesday", "Wednesday",  
6                 "Thursday", "Friday", "Saturday"};  
7 int score[] = {56, 72, 60, 85, 77};
```

## การกำหนดค่าให้แก่ตัวแปรอาเรย์ (ต่อ)

- รูปแบบที่ 2 กำหนดค่าเริ่มต้นของข้อมูลไว้ในอาเรย์โดยระบุตำแหน่งของอาเรย์ ตามรูปแบบ ดังนี้

```
arrayName[index] = value;
```

- โดย **index** คือ ตัวชี้ตำแหน่งของอาเรย์ และ
- **value** คือ ค่าของข้อมูล

## การกำหนดค่าให้แก่ตัวแปรอาเรย์ (ต่อ)

- เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรอาเรย์โดยระบุตัวชี้ตำแหน่งของอาเรย์ แต่จะต้องประกาศตัวแปรอาเรย์นั้นก่อนการกำหนดค่า เช่น

```
5      String days[] = new String[7];
6      days[0] = "Sunday";
7      days[1] = "Monday";
8      days[2] = "Tuesday";
9      days[3] = "Wednesday";
10     days[4] = "Thursday";
11     days[5] = "Friday";
12     days[6] = "Saturday";
```

## การกำหนดค่าให้แก่ตัวแปรอาเรย์ (ต่อ)

- รูปแบบที่ 3 ใช้รูป **for** ควบคุมการใส่ค่าข้อมูลเข้าไปในอาเรย์ เช่น ใส่คำสั่ง **for** เพื่อใส่ค่าข้อมูล 1-10 ลงไปในตัวแปรอาเรย์ “num” ซึ่งมีชนิดข้อมูลเป็น **integer** สามารถเขียนคำสั่งได้ ดังนี้

```
4      int num[] = new int[10];  
5      for(int i=0;i<10;i++){  
6          num[i] = i+1;  
7          System.out.println("Data in num[" + i + "] = " + num[i]);  
8      }
```

# การค้นหาข้อมูลในอาเรย์

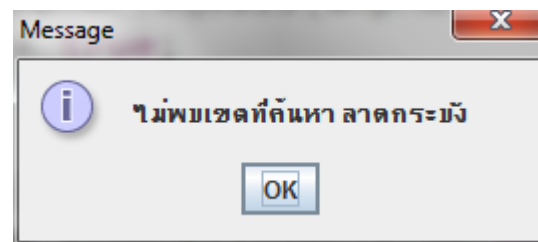
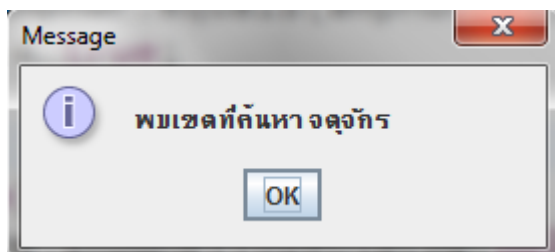
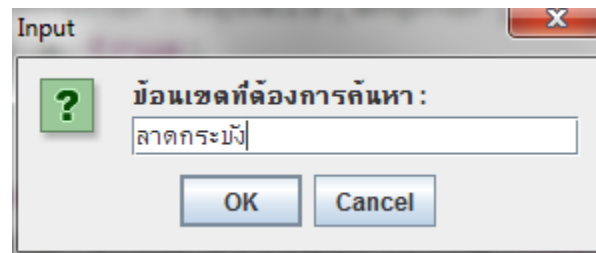
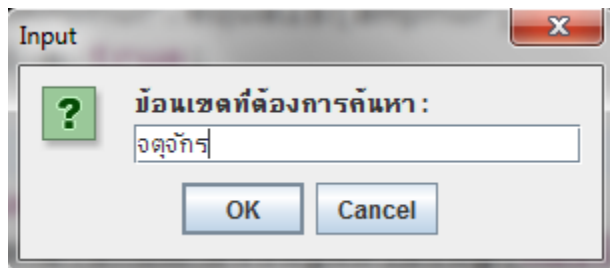
- เนื่องจากการเก็บข้อมูลแบบอาเรย์ เป็นการเก็บข้อมูลชุดเดียวกัน ไว้ในตัวแปรตัวเดียวกันในตำแหน่งเรียงลำดับต่อเนื่องกัน ดังนั้น การค้นหาข้อมูลที่ต้องการในอาเรย์ออกมาใช้งาน จึงต้องใช้วิธีการเปรียบเทียบข้อมูลที่ต้องการกับข้อมูลในอาเรย์ทีละตำแหน่ง หากข้อมูลตำแหน่งใดตรงตามที่ต้องการก็ให้นำค่าข้อมูลในตำแหน่งนั้นออกมาใช้
- พิจารณาตัวอย่างโปรแกรมค้นหาข้อมูลในอาเรย์อย่างง่าย ดังนี้

# การค้นหาข้อมูลในอาเรย์ (ต่อ)

```
SearchDataInArray.java
1 package prepare;
2 import javax.swing.JOptionPane;
3 public class SearchDataInArray {
4     public static void main(String[] args) {
5         boolean found = false;
6         String searchAmphur;
7         String amphur[] = {"จตุจักร", "อ่อนนุช", "ธนบุรี", "จอมทอง", "ลาดพร้าว"};
8         searchAmphur = JOptionPane.showInputDialog("ป้อนเขตที่ต้องการค้นหา : ");
9         for(int i=0;i<amphur.length;i++){
10             if(searchAmphur.equals(amphur[i])){
11                 found = true;
12             }
13         }
14         if(found==true)
15             JOptionPane.showMessageDialog(null, "พบเขตที่ค้นหา " + searchAmphur);
16         else
17             JOptionPane.showMessageDialog(null, "ไม่พบเขตที่ค้นหา " + searchAmphur);
18     }
19 }
```



# การค้นหาข้อมูลในอาเรย์ (ต่อ)



# การค้นหาข้อมูลแบบ binary search

- เป็นการค้นหาข้อมูลที่มีข้อมูลจำกัดว่าข้อมูลในอาเรย์ จะต้องถูกเรียงลำดับไว้เรียบร้อยแล้ว (น้อยไปมาก หรือมากไปน้อย)
- โดยมีแนวคิดในการค้นแบบ “แบ่งครึ่ง” ข้อมูลในอาเรย์
- เช่นการค้นหาชื่อคนในสมุดโทรศัพท์ ปกติจะเริ่มต้นจากครึ่งเล่มของสมุด แล้วจึงดูว่าชื่อที่ต้องการนั้นอยู่ทางครึ่งเล่มด้านหน้า หรือครึ่งเล่มด้านหลัง ถ้าสมมติว่าอยู่ครึ่งเล่มด้านหน้า เราก็จะไปเปิดที่ครึ่งเล่มด้านหน้า ( $1/4$  ของเล่ม) แล้วค้นหาเช่นนี้ต่อไปทีละครึ่ง จนกว่าจะพบชื่อที่ต้องการค้นหา

# หลักการค้นหาข้อมูลแบบ **binary search**

- ❑ 1. เรียงลำดับข้อมูลในอาเรย์จากน้อยไปมาก หรือจากมากไปน้อยก็ได้
- ❑ 2. แบ่งครึ่งข้อมูลในอาเรย์ออกเป็น 2 กลุ่ม เพื่อหา **index** กึ่งกลาง (ขนาดอาเรย์ - 1 แล้วหาร 2)
- ❑ 3. นำข้อมูลที่ต้องการค้นหามาเปรียบเทียบกับค่าของข้อมูลที่อยู่ **index** กึ่งกลาง สำหรับข้อมูลที่ถูกเรียงลำดับจากน้อยไปหามากแล้ว ถ้าค่าข้อมูลที่ต้องการค้นหามีค่าน้อยกว่าค่ากึ่งกลาง แสดงว่าข้อมูลที่ต้องการค้นหาจะอยู่ก่อนค่ากึ่งกลาง แต่ถ้าข้อมูลที่ต้องการค้นหามีค่ามากกว่าค่ากึ่งกลาง แสดงว่าข้อมูลที่ต้องการค้นหาอยู่หลังค่ากึ่งกลาง

# หลักการค้นหาข้อมูลแบบ **binary search**

- 4. นำข้อมูลที่อยู่กลุ่มแรกมาแบ่งครึ่งเป็น 2 กลุ่มย่อย แล้วนำค่าข้อมูลที่ต้องการค้นหามาเปรียบเทียบกับค่ากึ่งกลาง ทำซ้ำเช่นนี้จนกว่าจะไม่สามารถแบ่งกลุ่มย่อยออกไปได้อีก และหมดข้อมูลที่จะเปรียบเทียบกับค่ากึ่งกลางของกลุ่มตรงกับข้อมูลที่ต้องการค้นหา

# การค้นหาข้อมูลแบบ binary search

□ ตัวอย่าง สมมติว่าตัวเลขในอาเรย์ มีดังนี้

	10	20	30	40	50	60	70	80	90
index	0	1	2	3	4	5	6	7	8

□ ต้องการค้นหาตัวเลข 20

□ 1. แบ่งครึ่งอาเรย์ ขนาดอาเรย์คือ 9-1 แล้วหาร  $2 = 4$  (หากมีเศษให้ปัดขึ้น)

# การค้นหาข้อมูลแบบ **binary search**

- ❑ 2. นำค่าใน **index** ที่ 4 (คือ 50) เปรียบเทียบกับค่าที่ต้องการหา คือ 20 ซึ่ง 20 มีค่าน้อยกว่า 50 ดังนั้นผลลัพธ์ที่ได้คือ ได้ครึ่งด้านซ้าย ตั้งแต่ **index[0] – index[3]** คือ 4 ช่อง
- ❑ 3. นำจำนวนช่องทั้งหมดของครึ่งด้านซ้ายหาร 2 ( $4-1$  หาร  $2 = 1.5$  เศษ ให้ปัดขึ้น เป็น 2)
- ❑ 4. นำค่าในช่องที่ 2 (**index 1**) มาเปรียบเทียบกับค่าที่ต้องการหา (คือ 20) ได้เท่ากัน ดังนั้น คำตอบที่ได้คือ “ค้นพบข้อมูล 20 ในตำแหน่ง **index** ที่ 1 (ช่องที่ 2)

# การค้นหาข้อมูลแบบ **binary search**

- ตัวอย่าง เขียนโปรแกรมเพื่อค้นหาข้อมูลที่ต้องการ จากข้อมูลที่มีอยู่ 9 จำนวน ได้แก่ 10, 20, 30, 40, 50, 60, 70, 80, 90 โดยใช้วิธีการ **binary search**

# การค้นหาข้อมูลแบบ binary search

□ กำหนดตัวแปร

**mid**                    แทน ตัวชี้ตำแหน่ง

**low**                    แทน ตัวชี้ตำแหน่งขอบล่างนับจากตำแหน่งกลาง

**high**                    แทน ตัวชี้ตำแหน่งขอบบนนับจากตำแหน่งกลาง

**index**                    แทน แทนตัวชี้ตำแหน่งของข้อมูลในอาร์เรย์

**searchFor**                    แทน ข้อมูลที่ต้องการค้นหา

**num**                    แทน ข้อมูลทั้งหมด (เป็นตัวชี้ไปยังอาร์เรย์)



# การค้นหาข้อมูลแบบ **binary search**

## □ อัลกอริทึม

1. กำหนดตัวแปร **mid, low, high, index, searchFor** และอาร์เรย์ **num**
2. กำหนดค่าเริ่มต้นให้แก่ตัวแปร **mid = 0, low = 0, high = 0, index = -1** และใส่ค่าข้อมูลในอาร์เรย์ **num**
3. รับค่าข้อมูล **num** ที่ต้องการค้นหา

# การค้นหาข้อมูลแบบ **binary search**

□ อัลกอริทึม

4. ทดสอบค่า  $low \leq high$  และ  $index = -1$  หรือไม่

4.1 ถ้าจริง หาค่า  $index$  กลาง คือ  $mid = (low + high) / 2$  แล้วไปทำข้อ 4.3

4.2 ถ้าเท็จ ให้ไปทำข้อ 5

# การค้นหาข้อมูลแบบ **binary search**

## □ อัลกอริทึม

**4.3** ทดสอบค่า **num** ที่รับมากับค่า **num** ใน **index** กลางของอาเรย์

**4.3.1** ถ้า **searchFor < num[mid]** จริง ให้ **high = mid - 1** ขยับขอบบนของกลุ่มข้อมูลลงมานับจากตัวชี้ตำแหน่งกลาง (ยังไม่พบข้อมูล) แล้วไปทำข้อ **4**

**4.3.2** ถ้า **searchFor > num[mid]** จริง ให้ **low = mid + 1** ขยับขอบล่างของกลุ่มข้อมูลขึ้นไปนับจากตัวชี้ตำแหน่งกลาง (ยังไม่พบข้อมูล) แล้วไปทำข้อ **4**

**4.3.3** นอกเหนือจากนั้น (คือ **searchFor = num[mid]** พบข้อมูลที่ค้นหา) จริง ให้ **index = mid** แล้วไปทำข้อ **4** (ถึงตอนนี้ **index** จะมีค่าไม่เท่ากับ **-1**)

# การค้นหาข้อมูลแบบ **binary search**

□ อัลกอริทึม

5. ทดสอบค่า **index** ไม่เท่ากับ -1 จริงหรือไม่

5.1 ถ้าจริง ให้สั่งพิมพ์ “พบข้อมูลในตำแหน่ง...” แล้วไปทำข้อ 6

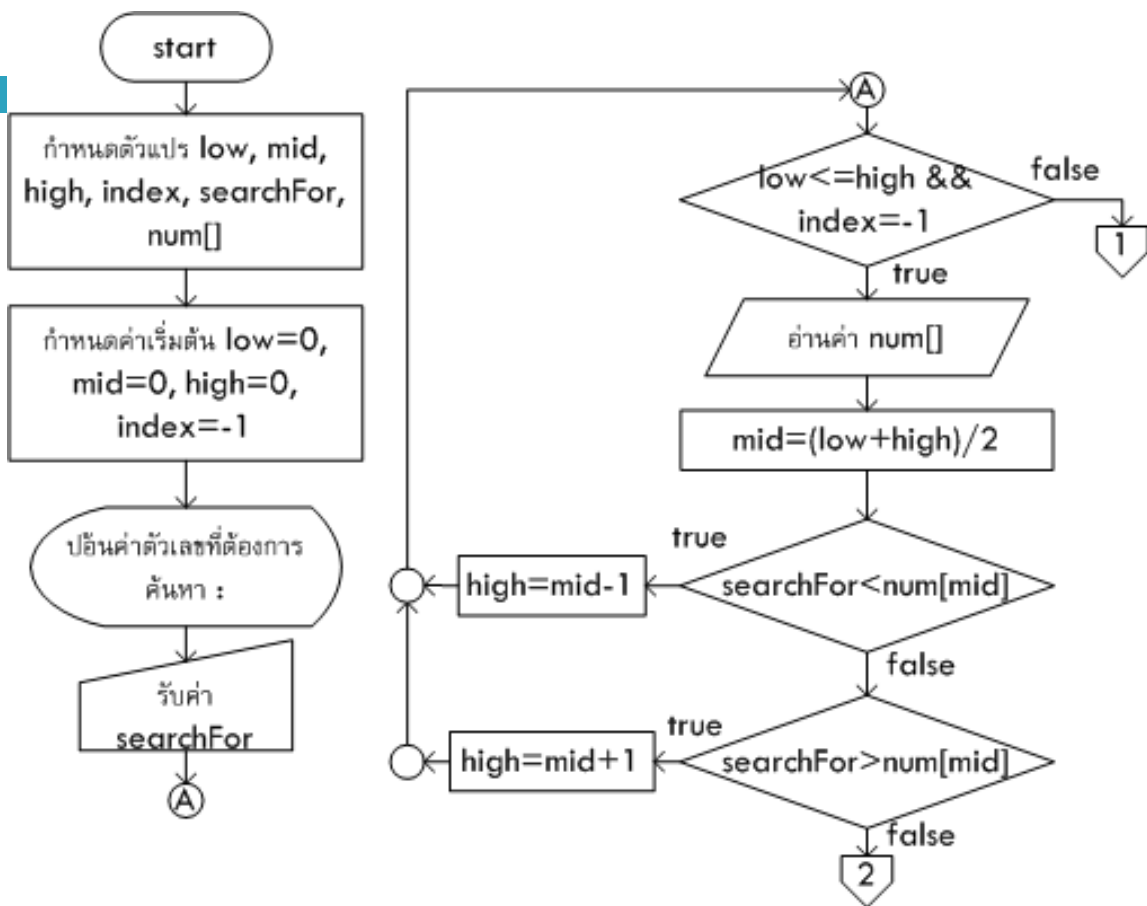
5.2 ถ้าเท็จ ให้สั่งพิมพ์ “ไม่พบข้อมูล” แล้วไปทำข้อ 6

6. สั่งพิมพ์ค่าในอาเรย์ทั้งหมด

7. จบการทำงาน

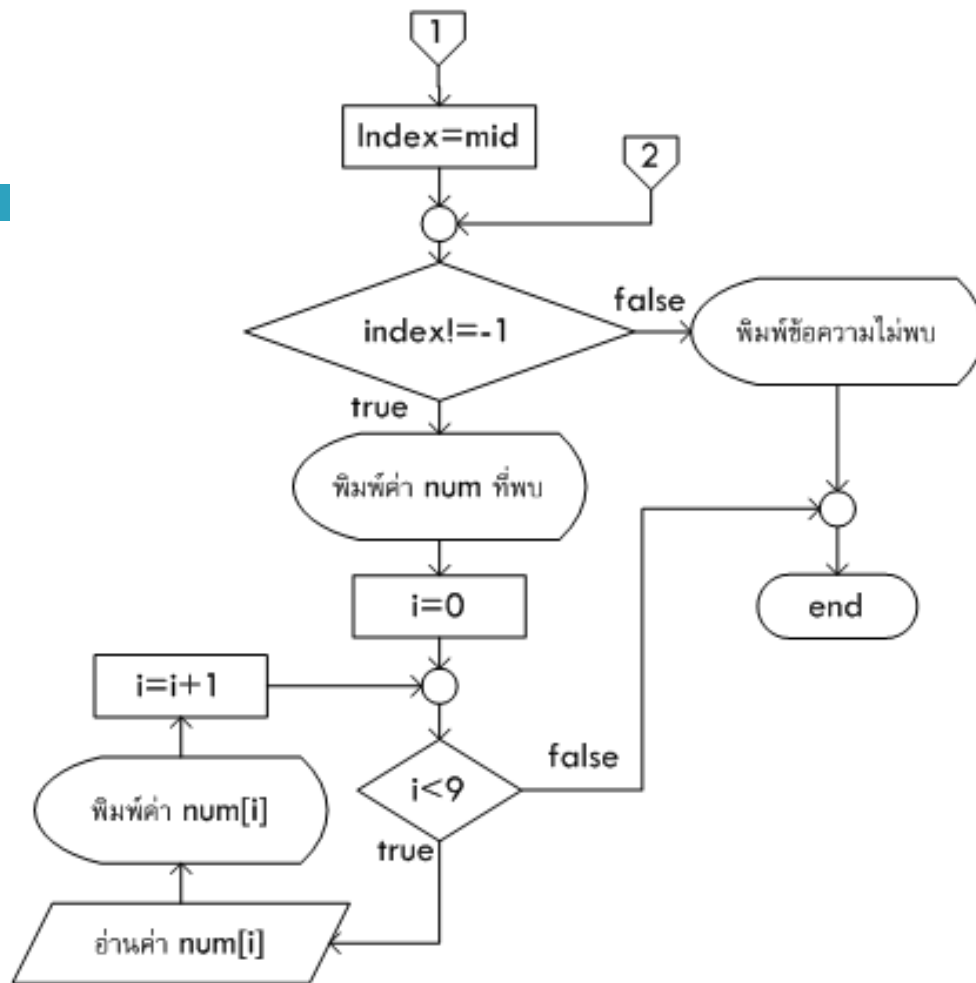
# binary search

□ ผังงาน



# binary search

□ ผังงาน



# Code

```
BinarySearch.java
1 package prepare;
2 import java.util.Scanner;
3 public class BinarySearch {
4     public static void main(String[] args) {
5         int num[] = {10,20,30,40,50,60,70,80,90};
6         int mid = 0, low = 0, high = num.length-1;
7         int searchFor, index = -1;
8         Scanner sc = new Scanner(System.in);
9         System.out.print("ป้อนตัวเลขที่ต้องการค้นหา : ");
10        searchFor = sc.nextInt();
11        while((low<=high)&&(index==-1)){
12            mid = (low+high)/2;
13            if(searchFor<num[mid])
14                high = mid-1;
15            else if(searchFor>num[mid])
16                low = mid+1;
17            else
18                index = mid;
19        }
```

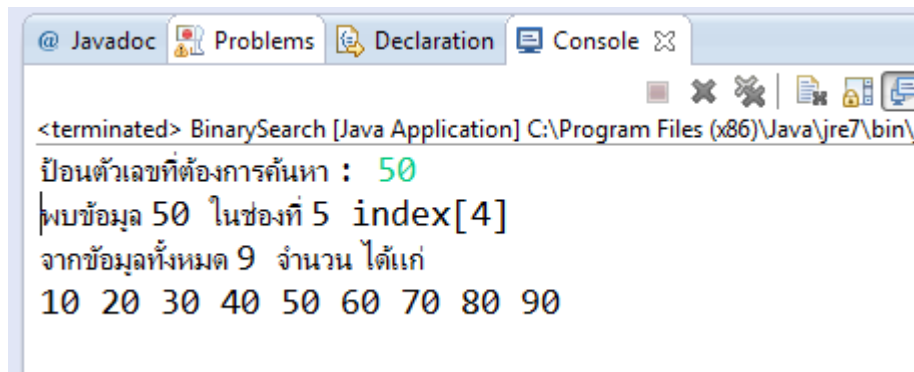
# Code

```
20         if(index!=-1)
21             System.out.println("พบข้อมูล " + searchFor +
22                 " ในช่องที่ " + (index+1) + " index[" + index + "]");
23         else
24             System.out.println("ไม่พบข้อมูลที่ต้องการ");
25         System.out.println("จากข้อมูลทั้งหมด 9 จำนวน ได้แก่");
26         for(int i=0;i<num.length;i++)
27             System.out.print(num[i] + " ");
28         sc.close();
29     }
30 }
```

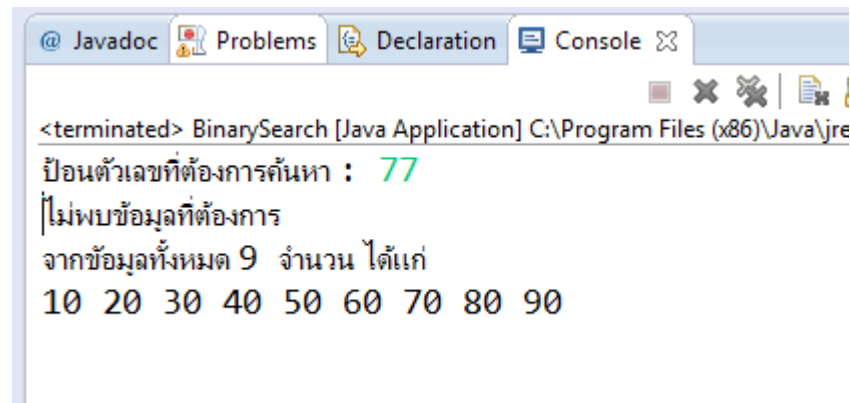


# binary search

## ❑ ผลลัพธ์



```
<terminated> BinarySearch [Java Application] C:\Program Files (x86)\Java\jre7\bin\  
ป้อนตัวเลขที่ต้องการค้นหา : 50  
พบข้อมูล 50 ในช่องที่ 5 index[4]  
จากข้อมูลทั้งหมด 9 จำนวน ได้แก่  
10 20 30 40 50 60 70 80 90
```



```
<terminated> BinarySearch [Java Application] C:\Program Files (x86)\Java\jre  
ป้อนตัวเลขที่ต้องการค้นหา : 77  
ไม่พบข้อมูลที่ต้องการ  
จากข้อมูลทั้งหมด 9 จำนวน ได้แก่  
10 20 30 40 50 60 70 80 90
```

# ลองคิดดู

- ❑ การหาค่ามากที่สุดในอาร์เรย์ (max)
- ❑ การหาดัชนีของค่าที่มากที่สุดในอาร์เรย์ (max index)
- ❑ การสลับข้อมูลสองตัวในอาร์เรย์ (swap)
- ❑ การล้างข้อมูลในอาร์เรย์ (clear)
- ❑ การเลื่อนข้อมูลในอาร์เรย์ (shift)
- ❑ การแทรกข้อมูลในอาร์เรย์ (insert)

# การเรียงลำดับข้อมูลในอาเรย์

- การเรียงลำดับข้อมูลในอาเรย์มีหลายวิธี ในที่นี้จะนำเสนอวิธีการที่ง่าย 3 วิธี การ อันได้แก่
- 1. Selection Sort
- 2. Bubble Sort
- 3. Insertion Sort

# Selection Sort

0	1	2	3	4
42	27	13	67	20
42	27	13	20	67
20	27	13	42	67
20	13	27	42	67
13	20	27	42	67

# Bubble Sort

0	1	2	3	4
42	27	13	67	20
27	13	42	20	67
13	27	20	42	67
13	20	27	42	67
13	20	27	42	67

# Insertion Sort

0	1	2	3	4
27	42	13	67	20
67	27	42	13	20
67	42	27	13	20
67	42	27	13	20
67	42	27	20	13

# อาร์เรย์ 2 มิติ

- การประกาศอาร์เรย์ 2 มิติ

```
int [][] num = new int[3][4];
```

- จากตัวอย่างนี้เป็นการประกาศตัวแปรอาร์เรย์ 2 มิติที่ชื่อ **num**
- โดยมีขนาด **3x4** (3 แถว 4 หลัก) ซึ่งมีค่าเท่ากับ **12** ข้อมูล
- ตัวแปรอาร์เรย์ **num** นี้ใช้ในการเก็บจำนวนเต็ม **int**

# อาร์เรย์ 3 มิติ

- การประกาศอาร์เรย์ 3 มิติ

```
int [][][] num = new int[3][4][3];
```

- จากตัวอย่างนี้เป็นการประกาศตัวแปรอาร์เรย์ 3 มิติที่ชื่อ **num**
- โดยมีขนาด **3x4x3** ซึ่งมีค่าเท่ากับ **36** ข้อมูล
- ตัวแปรอาร์เรย์ **num** นี้ใช้ในการเก็บจำนวนเต็ม **int**



# การประมวลผลเมทริกซ์

- อาร์เรย์ 1 มิติ บางทีเรียกว่า เวกเตอร์ (**vector**) (เมทริกซ์ที่มี 1 แถว หรือ มี 1 หลัก)
- อาร์เรย์ 2 มิติ บางทีก็เรียกว่า เมทริกซ์ (**matrix**)
- ในการประยุกต์ใช้อาร์เรย์ 2 มิติ ก็นำมาประมวลผลเมทริกซ์
- ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร เมทริกซ์ เป็นต้น

## การคูณเวกเตอร์ กับ เมทริกซ์

$$v \times A = [v_0 \ v_1] \times \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \end{bmatrix}$$

$$v \times A = [v_0 a_{0,0} + v_1 a_{1,0} \quad v_0 a_{0,1} + v_1 a_{1,1} \quad v_0 a_{0,2} + v_1 a_{1,2}] = [u_0 \ u_1 \ u_2]$$

□ สรุปได้ว่า สมาชิกของ  $u_i$  ของผลลัพธ์คำนวณได้จาก

$$u_i = \sum_{k=0}^{n-1} v_k a_{k,i}$$

□  $n$  คือจำนวนข้อมูลใน  $v$

# การคูณเมทริกซ์ กับ เมทริกซ์

- กำหนดให้  $C = A \times B$  (เมทริกซ์  $C$  เท่ากับ เมทริกซ์  $A$  คูณเมทริกซ์  $B$ )
- ถ้าจำนวนหลักของเมทริกซ์  $A$  เท่ากับ จำนวนแถวของเมทริกซ์  $B$  จะได้ว่า

$$c_{i,j} = \sum_{k=0}^{n-1} a_{i,k} b_{k,j}$$

- โดยที่  $n$  คือ จำนวนหลักของเมทริกซ์  $A$

## การคูณเมทริกซ์ กับ เมทริกซ์ (ต่อ)

□ ตัวอย่าง **A** มีขนาด **2x3** และ **B** มีขนาด **3x2** ได้ **C** มีขนาด **2x2**

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \end{bmatrix} \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \\ b_{2,0} & b_{2,1} \end{bmatrix} = \begin{bmatrix} a_{0,0}b_{0,0} + a_{0,1}b_{1,0} + a_{0,2}b_{2,0} & a_{0,0}b_{0,1} + a_{0,1}b_{1,1} + a_{0,2}b_{2,1} \\ a_{1,0}b_{0,0} + a_{1,1}b_{1,0} + a_{1,2}b_{2,0} & a_{1,0}b_{0,1} + a_{1,1}b_{1,1} + a_{1,2}b_{2,1} \end{bmatrix}$$
$$= \begin{bmatrix} c_{0,0} & c_{0,1} \\ c_{1,0} & c_{1,1} \end{bmatrix}$$

# เอกสารอ้างอิง

พนิดา พาณิชกุล. การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา **Java**.

กรุงเทพฯ : เค ที พี คอมพ์ แอนด์ คอนซัลท์, **2548**.

สมชาย ประสิทธิ์จิตรระกูล. เริ่มเรียนเขียนโปรแกรมฉบับภาษาจาวา. กรุงเทพฯ :

สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, **2552**.

End.

