

2110101 : การทำโปรแกรมคอมพิวเตอร์

Arrays

ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

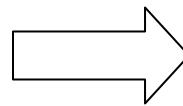
หัวข้อ

- **อาเรย์**
 - การประกาศ การสร้าง การใช้
- **การส่งอาเรย์ให้เมทอด**
- **การค้นข้อมูล**
- **การเรียงลำดับข้อมูล**
- **อาเรย์หลายมิติ**

โปรแกรมเก็บสถิติคะแนนสอบ

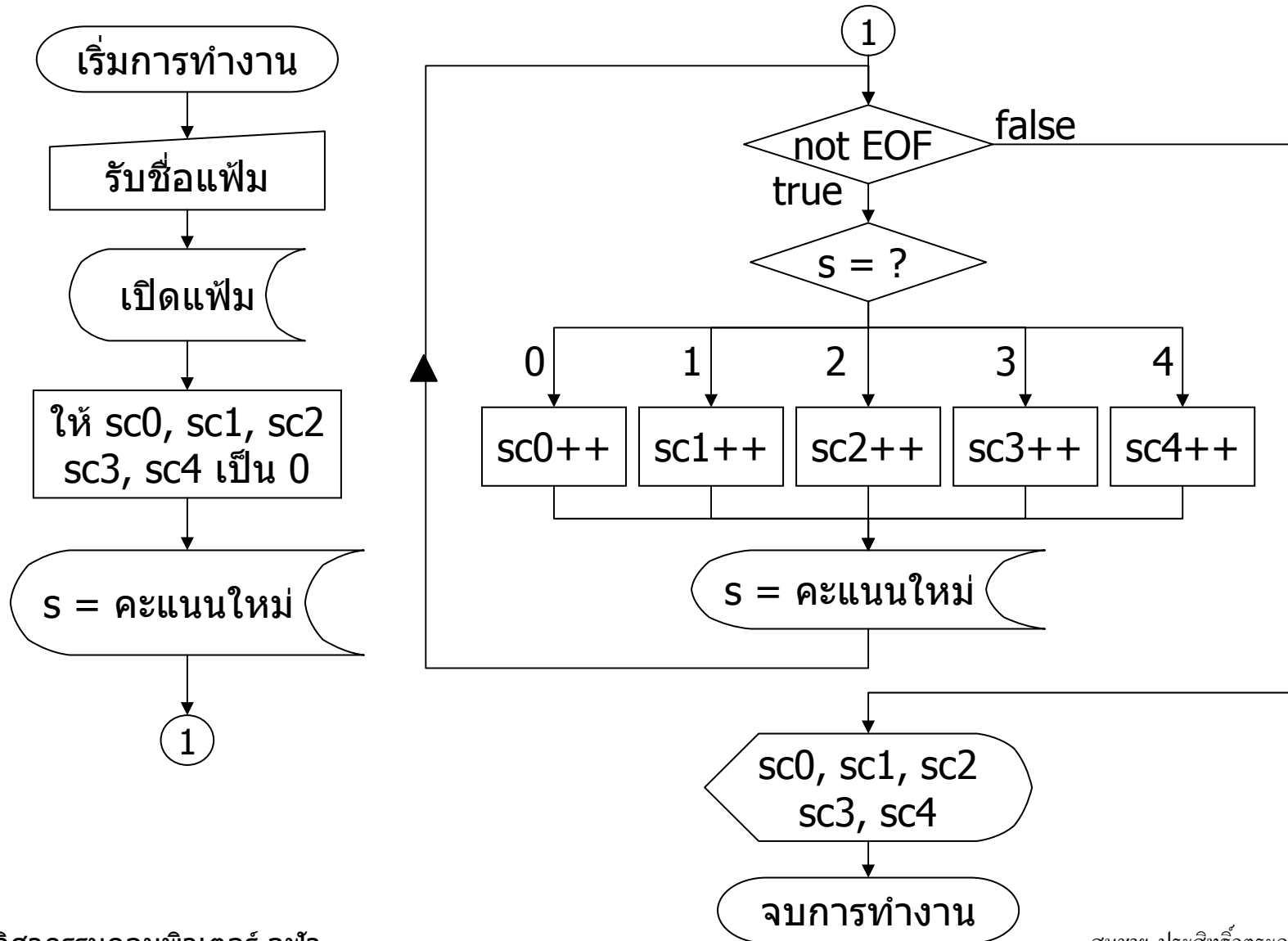
- มีแฟ้มข้อมูลเก็บรหัสนักเรียน + คะแนน
- คะแนนมีตั้งแต่ 0 ถึง 4 (จำนวนเต็ม)
- อยากทราบว่ามึนักเรียนได้ 0 ก็คน 1 ก็คน ... 4 ก็คน

4531001321	3
4531003621	3
4531004221	4
4531005921	3
4531006521	3
4531101421	4
4531102021	4
4531102221	0



คะแนน	จำนวนนักเรียน
0	1
1	0
2	0
3	4
4	3

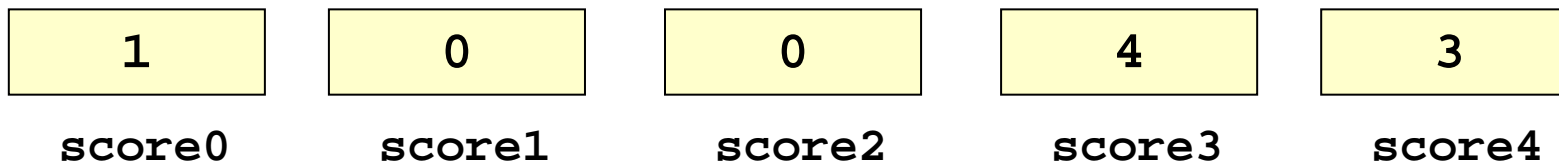
ผังงานการเก็บสถิติคะแนนสอบ



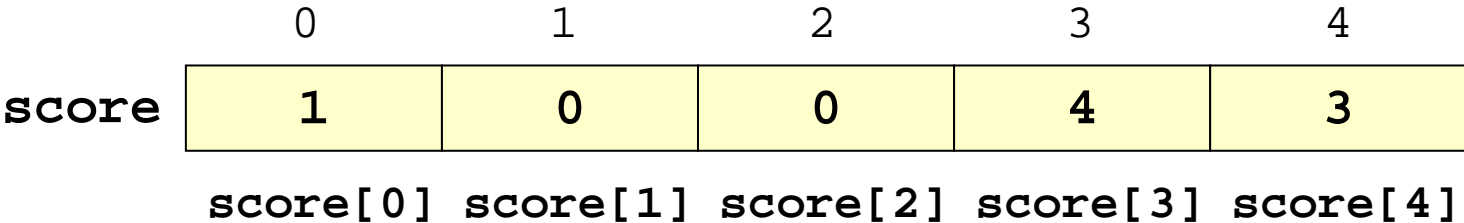
โปรแกรมเก็บสถิติคะแนนสอบ

```
import jlab.JLabIO;
import java.io.*;
public class Frequency1 {
    public static void main(String[] args) throws IOException {
        String txt, t;
        int sc, score0, score1, score2, score3, score4;
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));
        score0 = score1 = score2 = score3 = score4 = 0;
        while ((txt = in.readLine()) != null) {
            t = txt.substring(10, txt.length());
            switch ( Integer.parseInt(t.trim()) ) {
                case 0 : score0++; break;
                case 1 : score1++; break;
                case 2 : score2++; break;
                case 3 : score3++; break;
                case 4 : score4++; break;
                default : System.out.println("data error -> " + txt);
            }
        }
        in.close();
        // Five more lines to display score frequencies
        ...
    }
}
```

ใช้อาเรย์ดีกว่า



ตัวแปร 5 ตัว 5 ชื่อ



นำที่เก็บข้อมูลต่างๆ มารวมกลุ่มกัน 1 กลุ่ม 1 ชื่อ
โดยแต่ละที่เก็บมีหมายเลขกำกับ

โปรแกรมเก็บสถิติคะแนนสอบ (ใช้อาเรย์)

```
import jlab.JLabIO;
import java.io.*;
public class Frequency2 {
    public static void main(String[] args) throws IOException {
        String txt, t;
        int sc;
        int [] score = new int[5]; // declare + allocate array
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));

        for (int i=0; i<5; i++) score[i] = 0; // unnecessary

        while ((txt = in.readLine()) != null) {
            t = txt.substring(10, txt.length());
            sc = Integer.parseInt(t.trim());
            score[sc]++;
        }

        for (int i=0; i<5; i++) {
            System.out.println( i + " -> " + score[i] );
        }
    }
}
```

แถวลำดับ (Arrays)

- อาร์เรย์คือที่เก็บรายการของข้อมูลประเภทเดียวกัน
- อาร์เรย์เก็บข้อมูลทั้งหลายติดๆ กันไปในหน่วยความจำ เสมือนนำข้อมูลมาเก็บใน "ช่อง" ที่วางเรียงกัน
- แต่ละอาร์เรย์มีชื่อกำกับ
- ใช้หมายเลขช่องระบุตำแหน่งของข้อมูลในอาร์เรย์

	0	1	2	3	4	5	6	7	8	9	10	11
a	23	2	3	14	5	99	9	8	7	44	23	3

`System.out.println(a[6] + a[9]);`

การประกาศและการสร้างอาเรย์

- ก่อนใช้อาเรย์ ต้อง
 - ประกาศตัวแปรอาเรย์
 - สร้างตัวอาเรย์

```
int [] data;
```

ประกาศตัวแปรชื่อว่า data มีไว้อ้างอิงอาเรย์ของ int

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

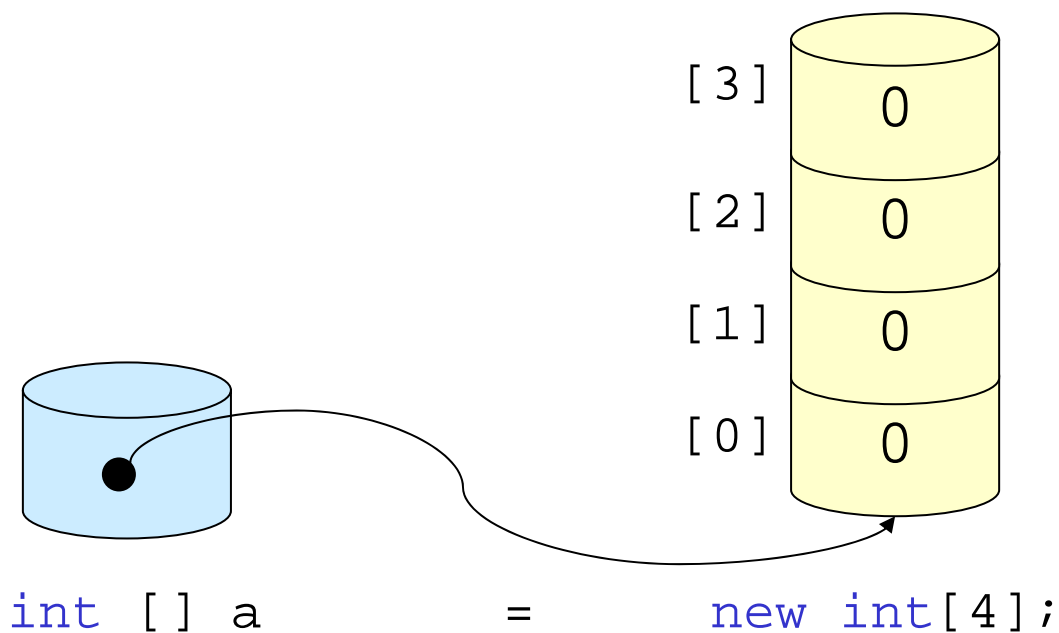
สร้างอาเรย์ของ int จำนวน 10 ช่อง แล้วให้ data เป็นตัวแปร
สำหรับอ้างอิงอาเรย์ที่ได้สร้างขึ้น

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

ประกาศตัวแปรและสร้างอาเรย์พร้อมกันเลยก็ได้

ตัวแปรอาร์เรย์กับตัวอาร์เรย์

- การประกาศตัวแปรอาร์เรย์ เป็นการสร้างที่เก็บ "ที่อยู่" (reference) ของอาร์เรย์ที่จะสร้าง

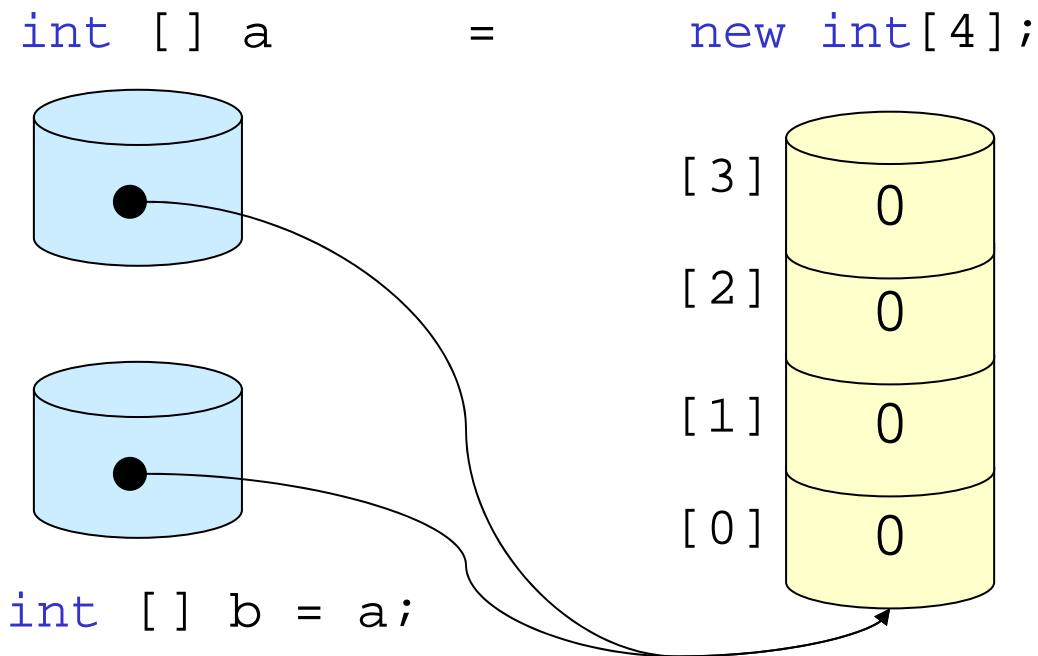


ข้อสังเกต : การประกาศ+การสร้าง

- การประกาศตัวแปรอาเรย์ ไม่ได้เป็นการสร้างตัวอาเรย์
- ขนาดของอาเรย์ถูกกำหนดตอนสร้างตัวอาเรย์ ไม่ใช่ตอนประกาศตัวแปร
- ขนาดของอาเรย์เป็นค่าของนิพจน์ก็ได้
(`data = new int[3*n + 1];`)
- อาเรย์ที่สร้างขึ้น ไม่สามารถเพิ่มหรือลดขนาดได้
- ระบบจะตั้งค่าเริ่มต้นของข้อมูลในอาเรย์โดยอัตโนมัติ
(ให้ค่าศูนย์กับอาเรย์ของจำนวน และ ให้ค่า `false` กับอาเรย์ของ `boolean`)

ข้อสังเกต : การอ้างอิงอาร์เรย์

- อาร์เรย์แถวหนึ่ง อาจมีตัวแปรหลายตัวอ้างอิงได้



a กับ b มีค่าเท่ากัน หมายความว่าอ้างอิงอาร์เรย์เดียวกัน
ดังนั้น $a[i]$ กับ $b[i]$ อ้างอิงที่เก็บข้อมูลที่เดียวกัน

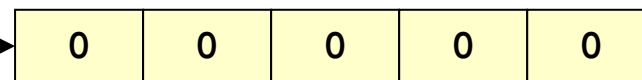
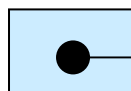
สร้างใหม่ได้ แต่ของเดิมหาย

```
int[] data;  
  
data = new int[4];  
  
data[2] = 9;  
  
data = new int[3];
```

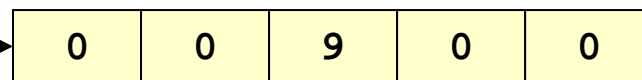
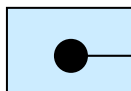
data



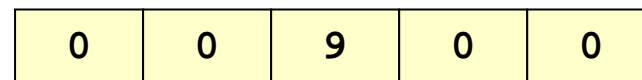
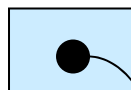
data



data



data



ประกาศ+สร้าง+ตั้งค่าเริ่มต้น

- ใช้ initializer list ระบุค่าเริ่มต้น
- ไม่ต้องกำหนดจำนวนช่องที่จะสร้าง
- จำนวนช่องที่สร้างเท่ากับจำนวนข้อมูลใน list

```
int[] data = new int[4];  
data[0] = 23;  
data[1] = 3;  
data[2] = 21;  
data[3] = 47;
```

```
int[] data = {23, 3, 21, 47};
```

```
int[] data;  
data = {23, 3, 21, 47};
```

เขียนแยกเป็นสองบรรทัดแบบนี้ไม่ได้

การใช้ข้อมูลในอาเรย์

- ใช้หมายเลขช่อง (เรียกว่า index) ของอาเรย์เป็นตัวอ้างอิงข้อมูลในอาเรย์ เช่น data[2] เป็นการอ้างอิงข้อมูลช่องที่ index 2 ในอาเรย์ data
- index ของอาเรย์เริ่มที่ 0 เสมอ
- ถ้าอาเรย์มีขนาด n ตัว index ต้องเป็นจำนวนเต็ม มีค่าได้ตั้งแต่ 0 ถึง n-1

```
public class Array1 {  
    public static void main(String [] args) {  
        int [] data = new int[100];  
        int sum = 0;  
        for(int i = 0; i < 100; i++)  
            sum += data[i];  
        System.out.println( sum );  
    }  
}
```

i = 0; i < 100

การอ้างอิงข้อมูลแบบผิดๆ

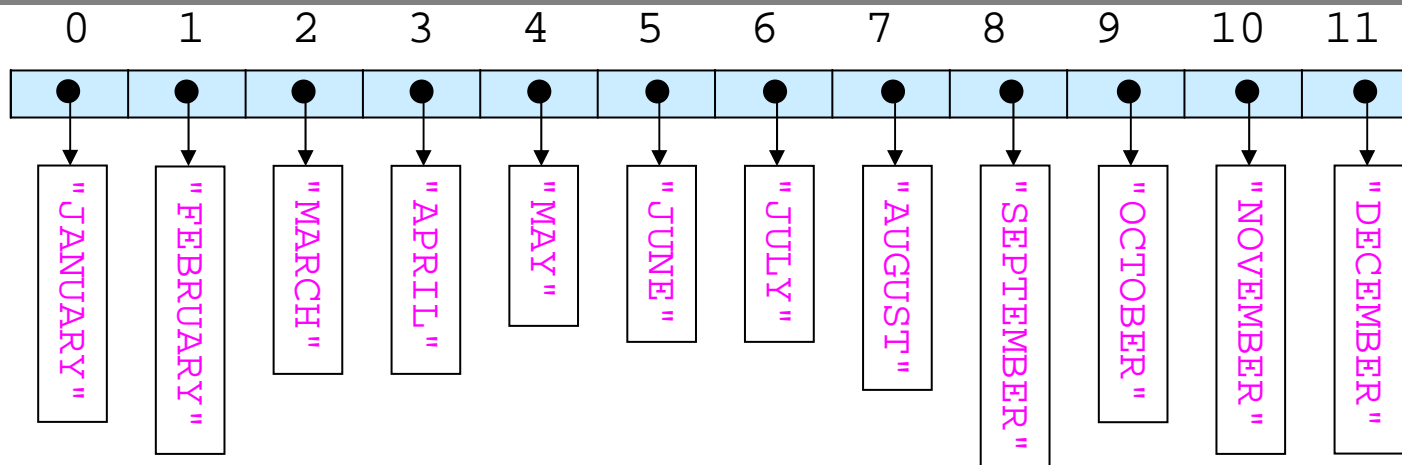
- ถ้าอาเรย์มีขนาด n ตัว
index ต้องเป็นจำนวนเต็มมีค่าได้ตั้งแต่ 0 ถึง $n-1$
- index ไม่ใช่จำนวนเต็ม - compile error
- index เป็นจำนวนเต็มที่อยู่นอกช่วง
 - เกิด **ArrayIndexOutOfBoundsException** ขณะทำงาน

```
public class Array3 {  
    public static void main(String [] args) {  
        int [] data = new int[100];  
        System.out.println( data[100] );  
    }  
}
```

```
JLab>java Array3  
java.lang.ArrayIndexOutOfBoundsException  
    at Array3.main(Array3.java:4)  
Exception in thread "main"  
JLab>
```

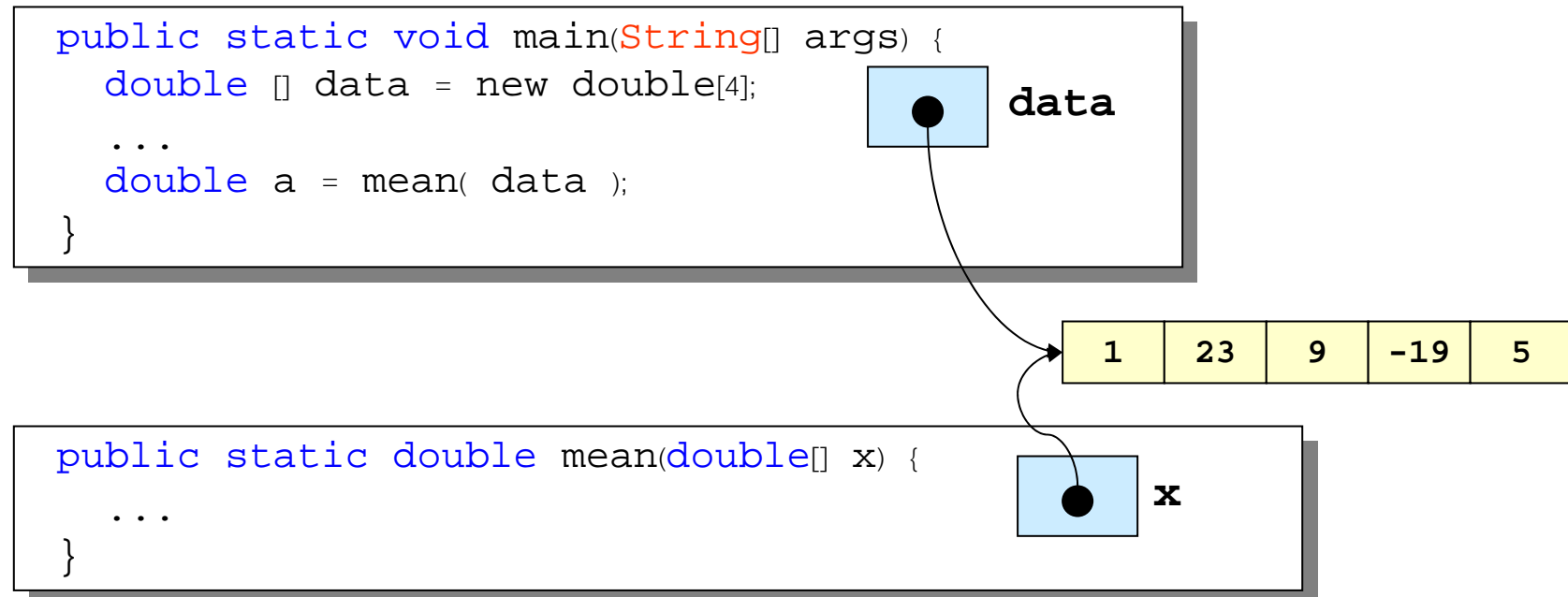

โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;
public class Month {
    public static void main(String [] args) {
        String month[] =
            { "JANUARY", "FEBRUARY", "MARCH", "APRIL",
              "MAY", "JUNE", "JULY", "AUGUST", "SEPTEMBER",
              "OCTOBER", "NOVEMBER", "DECEMBER" };
        int m = JLabIO.readInt("Enter month number : ");
        if (1<=m && m<=12) System.out.println(month[m-1]);
    }
}
```



การส่งอาร์เรย์ไปยังเมทอดอื่น

- ใช้ชื่ออาร์เรย์แทนทั้งอาร์เรย์ (ไม่มี [])
- การส่งอาร์เรย์จะไม่ได้ส่งข้อมูลของทั้งอาร์เรย์ไป แต่จะส่งตำแหน่ง (reference) ของอาร์เรย์ไปแทน



ถ้าอยากรู้ขนาดของอาเรย์

- เติม `.length` ตามหลังชื่ออาเรย์

```
public class Array2 {  
    public static void main(String [] args) {  
        int [] data = new int[100];  
        System.out.println( sum(data) );  
    }  
  
    public static int sum(int [] d) {  
        int sum = 0;  
        for(int i = 0; i < d.length; i++)  
            sum += d[i];  
        return sum;  
    }  
}
```

อย่าจำสลับกับความยาว `String` ซึ่งใช้ `.length()` เพราะเป็นเมทอด

โปรแกรมนับความถี่ตัวอักษรต่างๆ

```
import jlab.JLabIO;
import java.io.*;
public class CountChars {
    public static void main(String[] args) throws IOException {
        String txt;
        int[] freq = new int[26];
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));
        while ((txt = in.readLine()) != null) {
            int txtLen = txt.length();
            for (int i = 0; i < txtLen; i++) {
                char c = Character.toUpperCase(txt.charAt(i));
                if ('A' <= c && c <= 'Z') freq[c - 'A']++;
            }
        }
        for (int i = 0; i < freq.length; i++) {
            if (freq[i] > 0)
                System.out.println((char)'A' + i) + " --> " + freq[i]);
        }
        in.close();
    }
    ...
}
```

ในจาวา `char` เป็นจำนวนเต็มแบบ
ไม่ติดลบด้วย (unsigned integer)

เมท็อดหาค่าเฉลี่ย ค่าเบี่ยงเบน

```
public static double mean(double[] x) {  
    double sum = 0.0, m = 0.0;  
    if (x.length > 0) {  
        for (int i = 0; i < x.length; i++)  
            sum += x[i];  
        m = sum / x.length;  
    }  
    return m;  
}
```

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

```
public static double stddev(double[] x) {  
    double dev = 0;  
    double sumsq, m = mean(x);  
    if (x.length > 1) {  
        for (int i = 0; i < x.length; i++)  
            sumsq += (x[i] - m) * (x[i] - m);  
        dev = sum / (x.length - 1);  
    }  
    return dev;  
}
```

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

การค้นข้อมูลแบบลำดับ

- Requirement

- เมทอดเพื่อการค้นข้อมูลในอาเรย์

- Analysis

- รับ data เป็นอาเรย์ของจำนวนเต็ม
- รับ key เป็นจำนวนเต็มที่ต้องการค้นในอาเรย์
- ค้น index ตัวหนึ่งของ data ที่ข้อมูลมีค่าเท่ากับ key แต่ถ้าไม่ปรากฏ key ใน data ให้คืนค่า -1
- ใช้วิธีการค้นแบบลำดับ (sequential search)

```
public static int seqSearch( int [] data, int key )
```

	0	1	2	3	4	5	6	
data	23	2	3	14	5	99	9	key
								5

เรียก seqSearch(data, key) จะได้ 4

การค้นหาข้อมูลแบบลำดับ

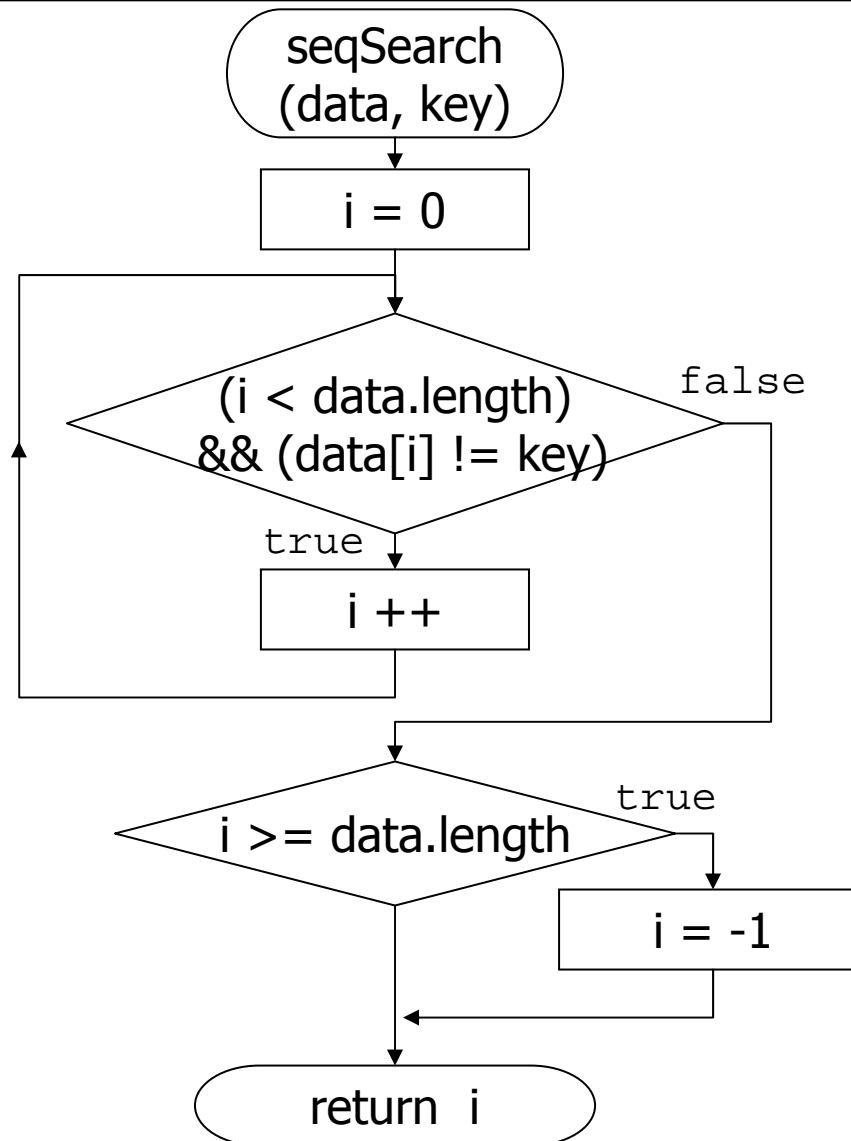
```
public static int sequentialSearch( int [] data, int key )
```

	0	1	2	3	4	5	6
data	23	2	3	14	5	99	9

key 5		
i	data[i]	data[i] == key
0	23	false
1	2	false
2	3	false
3	14	false
4	5	true
return i		

key 8		
i	data[i]	data[i] == key
0	23	false
1	2	false
2	3	false
3	14	false
4	5	false
5	99	false
6	9	false
7		
return -1		

ผังงานการค้นข้อมูลแบบลำดับ



เมทอดการค้นหาข้อมูลแบบลำดับ

```
public class TestSeqSearch {  
  
    public static int seqSearch(int[] data, int key) {  
        int i = 0;  
        while (i < data.length && data[i] != key) {  
            i++;  
        }  
        if (i >= data.length) i = -1;  
        return i;  
    }  
  
    // main is used for testing  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        System.out.println(seqSearch(data, 5));  
        System.out.println(seqSearch(data, 8));  
    }  
}
```

การค้นหาข้อมูลแบบลำดับ (ถอยหลัง)

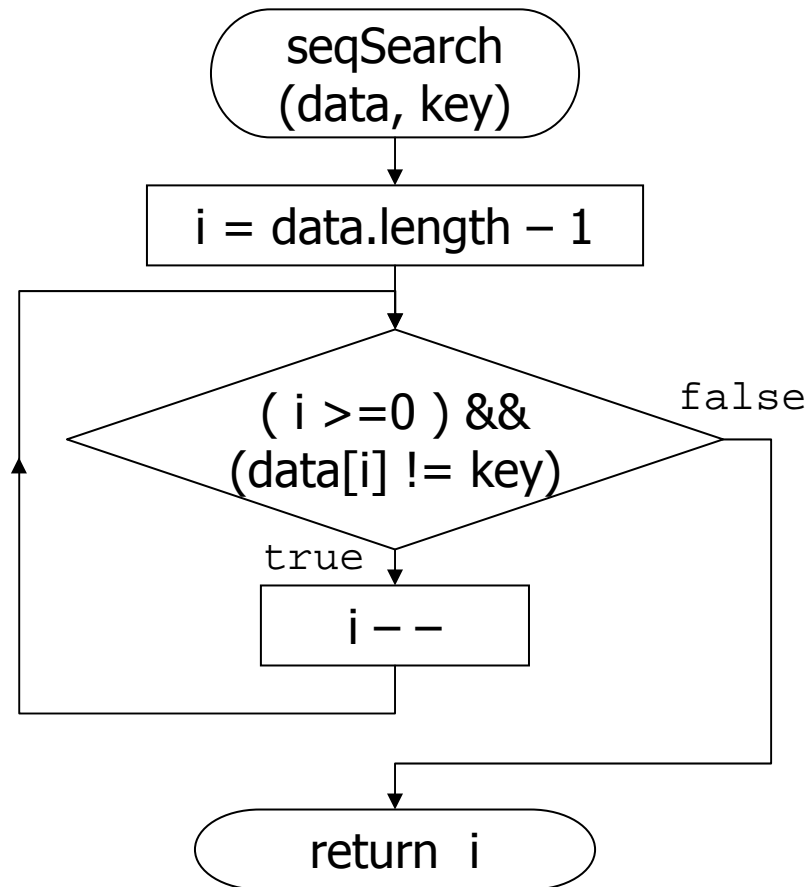
```
public static int sequentialSearch( int [] data, int key )
```

	0	1	2	3	4	5	6
data	23	2	3	14	5	99	9

key 5		
i	data[i]	data[i] == key
6	9	false
5	99	false
4	5	true
return i		

key 8		
i	data[i]	data[i] == key
6	9	false
5	99	false
4	8	false
3	14	false
2	3	false
1	2	false
0	23	false
-1		
return i		

ผังงานการค้นข้อมูลแบบลำดับ (ถอยหลัง)



เมทอดการค้นหาข้อมูลแบบลำดับ (ถอยหลัง)

```
public class TestSeqSearch {  
  
    public static int seqSearch(int[] data, int key) {  
        int i = data.length - 1;  
        while (i >= 0 && data[i] != key) {  
            i--;  
        }  
        return i;  
    }  
  
    // main is used for testing  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        System.out.println(seqSearch(data, 5));  
        System.out.println(seqSearch(data, 8));  
    }  
}
```

เขียนได้หลายแบบ

```
public static int seqSearch(int[] data, int key) {
    int i = data.length;
    while (i >= 0 && data[i] != key) i--;
    return i;
}
```

```
public static int seqSearch(int[] data, int key) {
    int i;
    for (i = data.length - 1;
         i >= 0 && data[i] != key;
         i--);
    return i;
}
```

อย่าลืมนะ

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (data[i] != key && i >= 0) i--;  
    return i;  
}
```

ตรวจสอบ $i \geq 0$ หลังการตรวจสอบ $data[i] \neq key$

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (i-- >= 0 && data[i] != key) ;  
    return i;  
}
```

i ถูกลดค่าก่อนที่จะใช้ $data[i]$

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (i >= 0 && data[i--] != key) ;  
    return i;  
}
```

i ถูกลดค่าทุกครั้งหลังเปรียบเทียบ
 $return i$ จะให้ผลผิด

การหาตำแหน่งที่มีค่ามากที่สุด

- Requirement
 - เมทอดเพื่อการหาตำแหน่งที่มีค่ามากสุดในอาเรย์
- Analysis
 - รับ data เป็นอาเรย์ของจำนวนเต็ม
 - คีน index ของอาเรย์ที่มีค่าไม่น้อยกว่าค่าใดๆ ในอาเรย์
 - ใช้วิธีการหาค่ามากสุดแบบลำดับ

```
public static int max( int [] data )
```

	0	1	2	3	4	5	6
data	23	2	3	14	5	99	9

เรียก max(data) จะได้ 5

เมท็อดการหาตำแหน่งที่มีค่ามากที่สุด

```
public class TestMax {
    public static int max(int[] data) {
        return max(data, data.length);
    }
    // find the max among the first k elements in the array
    public static int max(int[] data, int k) {
        int m = -1;
        if (k <= data.length && k > 1) {
            m = 0;
            for (int i = 1; i < k; i++)
                if (data[m] < data[i]) m = i;
        }
        return m;
    }
    public static void main(String[] args) {
        int[] data = {23, 2, 3, 14, 5, 99, 9};
        System.out.println(max(data));
        System.out.println(max(new int[0])); // zero size array
    }
}
```


การเรียงลำดับข้อมูล

- Requirement
 - เมท็อดเพื่อการเรียงลำดับข้อมูลในอาเรย์
- Analysis
 - รับ data เป็นอาเรย์ของจำนวนเต็ม
 - จัดลำดับข้อมูลใน data ให้เรียงลำดับจากน้อยไปมาก
 $data[0] \leq data[1] \leq data[2] \leq \dots$
 - ใช้วิธีการเรียงลำดับแบบเลือก (selection sort)

```
public static void selectionSort( int [] data )
```

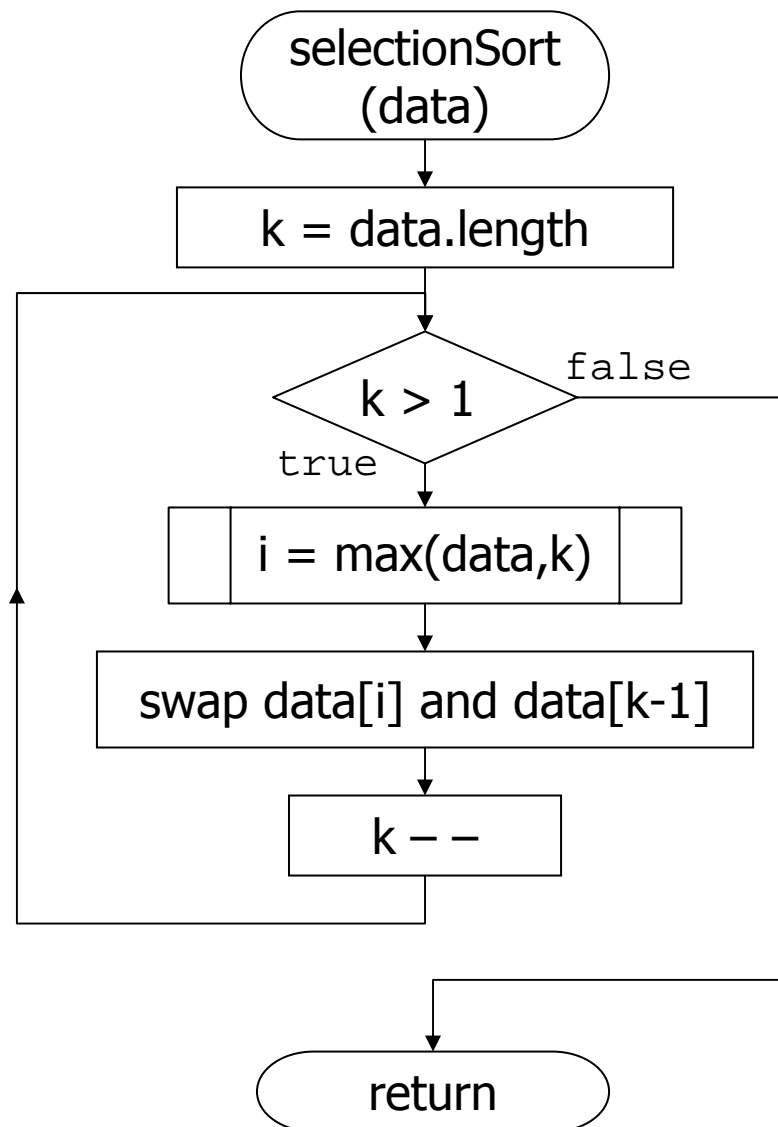
	0	1	2	3	4	5	6	
data	23	2	3	14	5	99	9	ก่อน
data	2	3	5	9	14	23	99	หลัง

การเรียงลำดับแบบเลือก

- มองอาเรย์เป็นสองส่วน
 - ส่วนขวา เรียงเสร็จแล้ว
 - ส่วนซ้าย ยังไม่เรียง
- ส่วนซ้าย
 - หาดตัวที่มีค่ามากที่สุด
 - สลับกับตัวสุดท้าย
- หลังสลับ
 - ส่วนซ้ายลดหนึ่งตัว
 - ส่วนขวาเพิ่มหนึ่งตัว
- เริ่มจากไม่มีส่วนขวา
หามากสุด + สลับ ไป
จนกระทั่งส่วนซ้ายเหลือตัว
เดียว ก็เสร็จ

23	2	3	14	5	99	9
23	2	3	14	5	9	99
23	2	3	14	5	9	99
9	2	3	14	5	23	99
9	2	3	14	5	23	99
9	2	3	5	14	23	99
9	2	3	5	14	23	99
5	2	3	9	14	23	99
5	2	3	9	14	23	99
3	2	5	9	14	23	99
3	2	5	9	14	23	99
2	3	5	9	14	23	99

ผังงานการเรียงลำดับแบบเลือก



เมท็อดการเรียงลำดับแบบเลือก

```
public class TestSort {
    public static void selectionSort(int[] data) {
        int i, t;
        for (int k = data.length; k > 1; k--) {
            i = max(data, k);
            t = data[i];
            data[i] = data[k - 1];
            data[k - 1] = t;
        }
    }
    public static void main(String[] args) {
        int[] data = {23, 2, 3, 14, 5, 99, 9};
        selectionSort(data);
        printArray(data);
    }
    public static void printArray(int[] data) {
        for (int i = 0; i < data.length; i++)
            System.out.print(data[i] + " ");
        System.out.println();
    }
    public static int max(int[] data, int k) {
        ...
    }
}
```

Arrays.binarySearch ของ Java API

- ค้นข้อมูล (ข้อมูลในอาเรย์ต้องเรียงลำดับแล้ว)

```
int binarySearch(byte[] a, byte key)
int binarySearch(short[] a, short key)
int binarySearch(int[] a, int key)
int binarySearch(long[] a, long key)
int binarySearch(float[] a, float key)
int binarySearch(double[] a, double key)
int binarySearch(char[] a, char key)
```

```
import java.util.Arrays;

public class TestSort {
    public static void main(String[] args) {
        int[] data = {2, 3, 5, 9, 14, 23, 99};
        System.out.println(Arrays.binarySearch(data, 5));
        System.out.println(Arrays.binarySearch(data, 8));
    }
}
```

Arrays.equals ของ Java API

- ตรวจสอบว่าอาร์เรย์สองแถวมีค่าเท่ากันหรือไม่

```
boolean equals(boolean[] a, boolean[] b)
boolean equals(byte[] a, byte[] b)
boolean equals(short[] a, short[] b)
boolean equals(int[] a, int[] b)
boolean equals(long[] a, long [] b)
boolean equals(float[] a, float [] b)
boolean equals(double[] a, double [] b)
boolean equals(char[] a, char [] b)
```

Arrays.fill ของ Java API

- เติมค่าเริ่มต้นให้ช่องต่างๆ เท่ากับค่าที่กำหนดให้

```
void fill(boolean[] a, boolean val)
void fill(byte[] a, byte val)
void fill(short[] a, short val)
void fill(int[] a, int val)
void fill(long[] a, long val)
void fill(float[] a, float val)
void fill(double[] a, double val)
void fill(char[] a, char val)
```

```
void fill(boolean[] a, int from, int to, boolean val)
void fill(byte[] a, int from, int to, byte val)
void fill(short[] a, int from, int to, short val)
void fill(int[] a, int from, int to, int val)
void fill(long[] a, int from, int to, long val)
void fill(float[] a, int from, int to, float val)
void fill(double[] a, int from, int to, double val)
void fill(char[] a, int from, int to, char val)
```

Arrays.sort ของ Java API

- เรียงลำดับข้อมูล

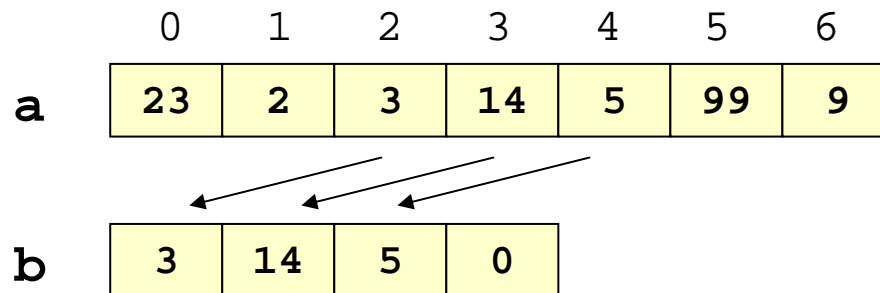
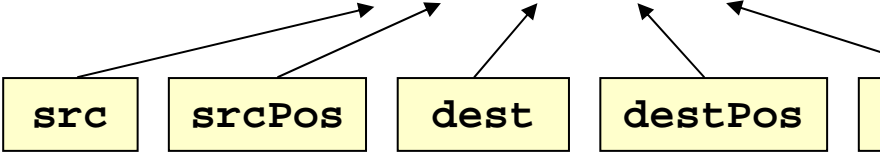
```
void sort(boolean[] a)
void sort(byte[] a)
void sort(short[] a)
void sort(int[] a)
void sort(long[] a)
void sort(float[] a)
void sort(double[] a)
void sort(char[] a)
```

```
void sort(boolean[] a, int from, int to)
void sort(byte[] a, int from, int to)
void sort(short[] a, int from, int to)
void sort(int[] a, int from, int to)
void sort(long[] a, int from, int to)
void sort(float[] a, int from, int to)
void sort(double[] a, int from, int to)
void sort(char[] a, int from, int to)
```


System.arraycopy ของ Java API

- copy ข้อมูลจากอาเรย์ไปยังอีกอาเรย์หนึ่ง

```
public class ArrayCopy {  
    public static void main(String[] args) {  
        int[] a = {23, 2, 3, 14, 5, 99, 9};  
        int[] b = new int[4];  
        System.arraycopy(a, 2, b, 0, 3);  
        ...  
    }  
}
```



อาร์เรย์หลายมิติ (Multidimensional Array)

```
int [] a = new int[7];  
a[2] = 3;
```

หนึ่งมิติ
(vector)

	0	1	2	3	4	5	6
a	0	0	3	0	0	0	0

```
int [][] b = new int[3][5];  
b[1][3] = 4;
```

สองมิติ
(matrix)

b	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	4	0
2	0	0	0	0	0

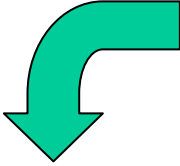
Initializer List

```
int [][] b = { {1, 2, 3, 4, 5},  
               {5, 4, 3, 2, 1},  
               {10, 20, 30, 40, 50} };
```

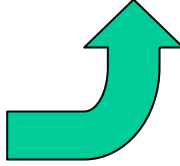
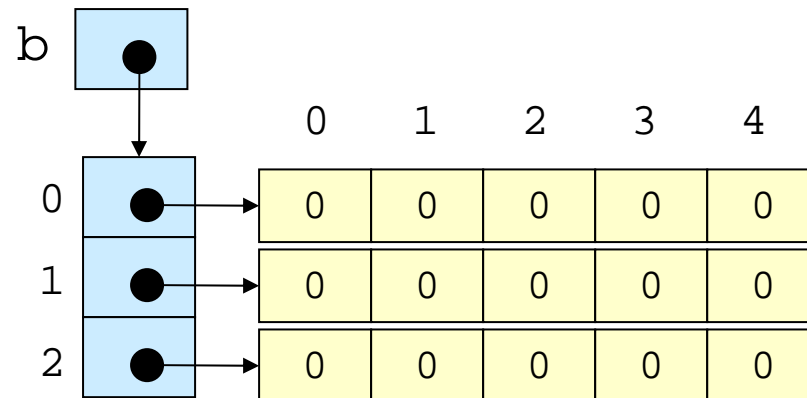
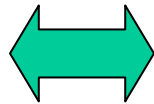
b	0	1	2	3	4
0	1	2	3	4	5
1	5	4	3	2	1
2	10	20	30	40	50

อาเรย์ของอาเรย์

```
int [][] b = new int[3][5];
```



b	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0



```
int [][] b;  
b = new int[3][];  
b[0] = new int[5];  
b[1] = new int[5];  
b[2] = new int[5];
```

เมทอดแสดงค่าในอาร์เรย์สองมิติ

```
public class TestPrintArray2 {  
    public static void main(String[] args) {  
        int[][] b = {{1, 2, 3, 4, 5},  
                    {5, 4, 3, 2, 1},  
                    {10, 20, 30, 40, 50}};  
  
        printArray2(b);  
    }  
    public static void printArray2(int[][] a) {  
        for (int i = 0; i < a.length; i++) {  
            for (int j = 0; j < a[i].length; j++)  
                System.out.print(a[i][j] + "\t ");  
            System.out.println();  
        }  
    }  
}
```

```
JLab>java TestPrintArray2
```

```
1         2         3         4         5  
5         4         3         2         1  
10        20        30        40        50
```

```
JLab>
```

การบวกเมทริกซ์

- Requirement
 - เมท็อดสำหรับการบวกเมทริกซ์
- Analysis
 - รับ a และ b เป็นอาร์เรย์สองมิติของ int
 - คืนอาร์เรย์สองมิติซึ่งเป็นผลบวกของ a กับ b

$$c_{ij} = a_{ij} + b_{ij}$$

เมท็อดสำหรับการบวกเมทริกซ์

```
public class TestAddMatrices {
    public static void main(String[] args) {
        int[][] b = {{1, 2, 3, 4, 5},
                    {5, 4, 3, 2, 1},
                    {10, 20, 30, 40, 50}};
        printArray2(addMatrices(b, b));
    }
    public static int[][] addMatrices(int[][] a,
                                      int[][] b) {
        int[][] c = new int[a.length][a[0].length];
        for (int i = 0; i < a.length; i++)
            for (int j = 0; j < a[i].length; j++)
                c[i][j] = a[i][j] + b[i][j];
        return c;
    }
    public static void printArray2(int[][] a) {
        ...
    }
}
```

การคูณเมทริกซ์

- Requirement
 - เมท็อดสำหรับการคูณเมทริกซ์
- Analysis
 - รับ a และ b เป็นอาร์เรย์สองมิติของ `int`
 - คืนอาร์เรย์สองมิติซึ่งเป็นผลคูณของ a กับ b
 - ถ้า a มีขนาด $[p \times q]$ และ b มีขนาด $[q \times r]$ ผลคูณของ $a \times b$ จะเป็นเมทริกซ์ขนาด $[p \times r]$

$$c_{ij} = \sum_{k=1}^q a_{ik} b_{kj}$$

เมท็อดสำหรับการคูณเมทริกซ์

```
public static int[][] mulMatrices(int[][] a,
                                   int[][] b) {

    int[][] c = new int[a.length][b[0].length];
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < b[i].length; j++) {
            c[i][j] = 0;
            for (int k = 0; k < b.length; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    return c;
}
```

$$c_{ij} = \sum_{k=1}^q a_{ik} b_{kj}$$

จำนวนฟีโบนัชชี

```
import jlab.JLabIO;
public class TestMulMatrices {
    public static void main(String[] args) {
        int[][] f1 = {{0, 1}, {1, 1}};
        int[][] fn = {{0, 1}, {1, 1}};
        int n = JLabIO.readInt("Fibonacci n = ");
        for (int i = 1; i < n; i++)
            fn = mulMatrices(f1, fn);
        System.out.println("f(" + n + ") = "
            + fn[0][1]);
    }

    public static int[][] mulMatrices(int[][] a,
        int[][] b) {
        ...
        }
    }
}
```

$$\dots \begin{bmatrix} f_{n-1} & f_n \\ f_n & f_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n$$

การ return ข้อมูลหลายๆ ค่า

- ใช้อาเรย์ในการคืนข้อมูลหลายๆ ตัวที่เป็นประเภทเดียวกัน (เช่นต้องการคืนรากของสมการทั้งสองตัว)

```
public static double [] quadraticRoot(double a,
                                     double b,
                                     double c) {

    double [] root;

    double t = Math.sqrt(b * b - 4 * a * c);
    if (t > 0) {
        root = new double[2];
        root[0] = (-b + t) / (2 * a);
        root[1] = (-b - t) / (2 * a);
    } else {
        root = null;
    }
    return root;
}
```

$$f(x) = ax^2 + bx + c$$

`null` เป็นค่าคงตัวในจาวา มักใช้แทน กรณี "แปลก" (ในที่นี้แทนการไม่มีรากจริง)