Object Oriented Programming

Lecture#4

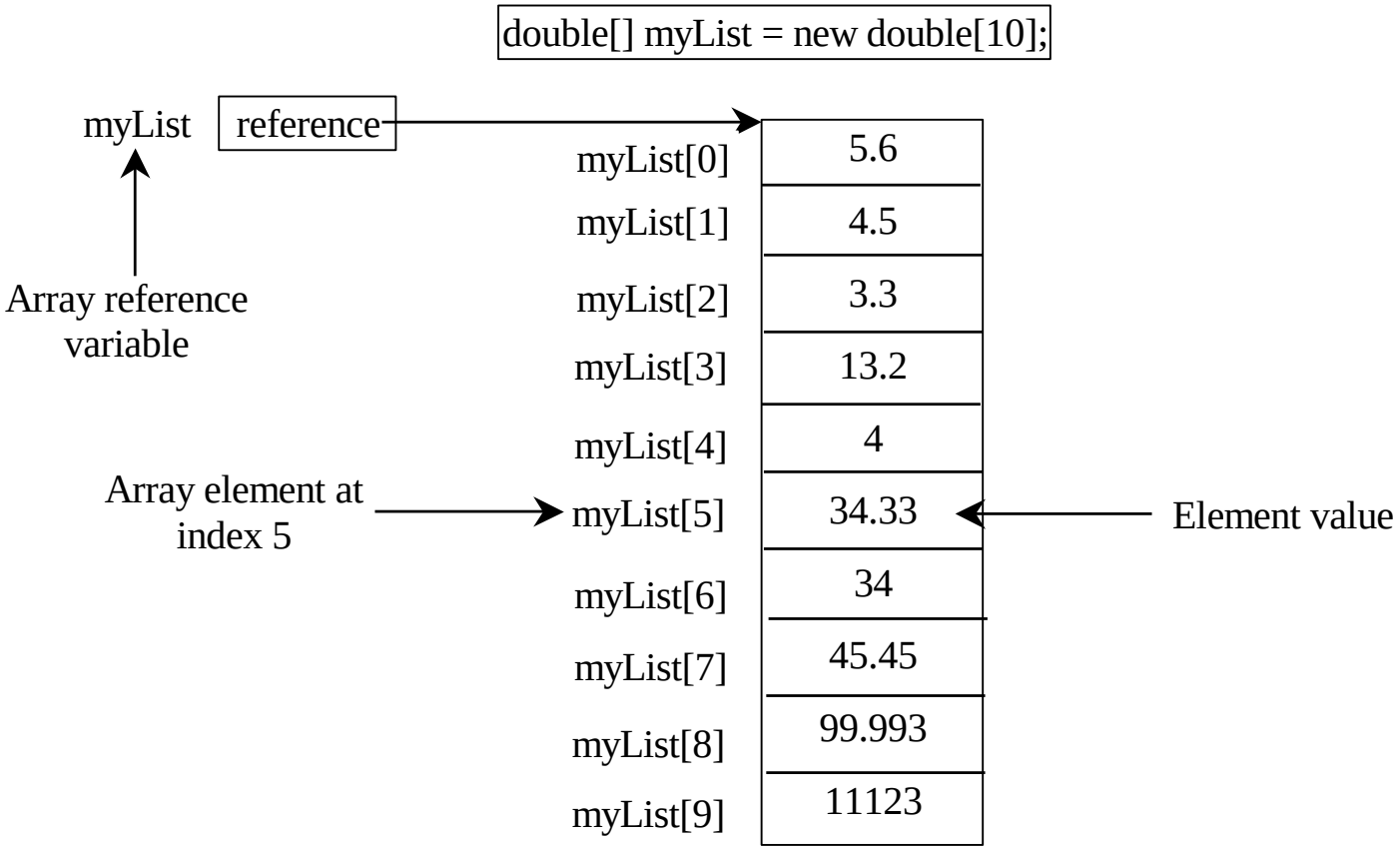
Array, Multidimensional Array

อาร์เรย์

- **ตัวแปรชุด (array) array** เป็นชนิดข้อมูลประเภทหนึ่งที่นำเอาชนิดข้อมูลข้อมูลพื้นฐานมาประยุกต์เป็นชนิดข้อมูลประเภทนี้ เช่น
  - ตัวอักษร(char)
  - ชนิดข้อมูลแบบเลขจำนวนเต็ม(int)
  - ชนิดข้อมูลแบบเลขจำนวนจริง(float)
- เมื่อประกาศโครงสร้างข้อมูลแบบอาร์เรย์(Array) จะเก็บข้อมูลต่างจากชนิดข้อมูลพื้นฐานทั่วไป คือ สามารถเก็บค่าภายในตัวแปรชนิดนี้ได้มากกว่า 1 ค่าซึ่งจำนวนค่าที่เก็บนั้นขึ้นอยู่กับขนาดของอาร์เรย์ที่ได้กำหนดไว้

ตัวอย่างของอาร์เรย์ myList

อาร์เรย์คือโครงสร้างข้อมูลที่มีชนิดข้อมูลประเภทเดียวกัน



ประเภทของตัวแปรชุด

- อาจแบ่งตามลักษณะของจำนวนตัวเลขของดัชนี คือ
  - 1. ตัวแปรชุด 1 มิติ (one dimension arrays หรือ single dimension arrays) เป็นตัวแปรชุดที่มีตัวเลขแสดงขนาดเป็นเลขตัวเดียว เช่น word[20] ,num[25] , x[15]
  - 2. ตัวแปรชุดหลายมิติ (multi-dimension arrays) เป็นตัวแปรชุดที่ชื่อมีตัวเลขแสดงขนาดเป็นตัวเลขหลายตัว ที่นิยมใช้กันมี 2 มิติ กับ 3 มิติ
    - 2.1 ตัวแปรชุด 2 มิติ มีเลขแสดงขนาด 2 ตัว เช่น a[3][5] , name[5][6]
    - 2.2 ตัวแปรชุด 3 มิติ มีเลขแสดงขนาด 3 ตัว เช่น a[3][5][6] , name[5][6][8]

การประกาศและสร้างตัวแปรอาร์เรย์

การประกาศตัวแปรอาร์เรย์

```
datatype[] arrayRefVar;
```

Example:

```
double[] myList;
```

การสร้างอาร์เรย์

```
arrayRefVar = new datatype[arraySize];
```

Example:

```
myList = new double[10];
```

`myList[0]` อ้างถึงสมาชิกตัวแรกในอาร์เรย์

`myList[9]` อ้างถึงสมาชิกในอาร์เรย์

การประกาศและสร้างอาร์เรย์ในหนึ่งขั้นตอน

```
datatype[] arrayRefVar = new  datatype[arraySize];
```

```
double[] myList = new double[10];
```

Array Initializers

- การประกาศ สร้าง และกำหนดค่าให้อาร์เรย์ในหนึ่งขั้นตอน

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

- การประกาศข้างต้นจะมีค่าเหมือนกับ

```
double[] myList = new double[4];
myList[0] = 1.9;
myList[1] = 2.9;
myList[2] = 3.4;
myList[3] = 3.5;
```

~~
```
double[] myList;
myList = {1.9, 2.9, 3.4, 3.5};
```
~~

อาร์เรย์

- เมื่ออาร์เรย์ถูกประกาศแล้วจะไม่สามารถเปลี่ยนขนาดของอาร์เรย์นอกจากนี้ยังสามารถหาขนาดของอาร์เรย์ได้จาก arrayRefVar.length
- เมื่ออาร์เรย์ถูกสร้าง จะมีการกำหนดค่าเริ่มต้นให้กับอาร์เรย์ด้วยค่าเริ่มต้น
  - 0 สำหรับชนิดข้อมูลแบบตัวเลขในprimitive data types,
  - '\u0000' สำหรับชนิดข้อมูลแบบตัวอักษร
  - false สำหรับชนิดข้อมูลแบบ boolean
- สมาชิกของอาร์เรย์สามารถเข้าถึงได้ด้วย index โดย indexของอาร์เรย์เริ่มที่ 0 ถึง ขนาดของอาร์เรย์ลบ 1(arrayRefVar.length-1)

```
arrayRefVar[index];
```

การประกาศตัวแปรชุด 1 มิติของข้อมูลประเภท integer

- int[] a=new int[10]; เป็นการประกาศตัวแปร array ชื่อ a เป็น array ของข้อมูลประเภท integer มีสมาชิกได้ จำนวน 10 ตัว คือ a[0] a[1] a[2] a[3] … a[9] โดยมีการจองเนื้อที่ในหน่วยความจำเปรียบเทียบได้ดังรูป

- โดยสมาชิกแต่ละตัวจะใช้เนื้อที่เท่ากับตัวแปรประเภท integer ที่ไม่ได้อยู่ใน array คือ 2 ไบต์ ต่อ ตัวแปร 1 ตัวดังนั้นเนื้อที่หน่วยความจำที่ใช้ทั้งหมดจึงเท่ากับจำนวนสมาชิก คูณด้วย 2 ไบต์

การประกาศตัวแปรชุด 1 มิติของข้อมูลประเภท integer

- การกำหนดค่าให้แก่ตัวแปร array อาจกำหนดพร้อมกับการประกาศ เช่น

```
int[] num1 ={56,25,89};
```

- เป็นการประกาศว่าตัวแปร num1 เป็น array ประเภท integer มีสมาชิก 3 ตัวโดย
  - num1[0] = 56;
  - num1[1]=25;
  - num1[2]=89;

```
int a[ ]={200,230};
```

- ประกาศว่า a เป็นตัวแปร array ประเภท integer ที่มีสมาชิก 2 ตัว โดย
  - a[0] มีค่า เป็น 200
  - a[1] มีค่าเป็น 230

การประกาศตัวแปรชุด 1 มิติของข้อมูลประเภท float

- float[] b=new float[10]; เป็นการประกาศตัวแปร array ของ ตัวแปรจำนวนที่มีทศนิยมได้ คือ float ในชื่อ b ซึ่งมีสมาชิกได้ 5 ตัว คือ b[0] b[1]... b[9] มีการจองเนื้อที่ในหน่วยความจำเปรียบเทียบได้ ดังรูป

- โดยสมาชิกแต่ละตัวใช้หน่วยความจำ 4 ไบต์ ดังนันทังหมดจะใช้หน่วยความจำ 4 คูณ 5 คือ40 ไบต์

การประกาศตัวแปรชุด 1 มิติของข้อมูลประเภท float

- การกำหนดค่าของตัวแปร array ประเภท float เป็นไปในลักษณะเดียวกับ array ประเภท integer ประกาศพร้อมกับกำหนดค่าให้เลยโดยล้อมรอบด้วย { } และค่าของสมาชิกแต่ละตัวคั่นด้วย , เช่น

  float[] num = {2.00,1.25,5.36,6.32,246.10};

  num[0] = 2.00

  num[1] = 1.25

  num[2] = 5.36

  num[3] = 6.32

  num[4] = 246.10

- หรือประกาศตัวแปรก่อนแล้วไปกำหนด ค่าภายหลัง เช่น float salary[10];
  - salary[0] = 25000.00;
  - salary[9] = 55600.00;

ตัวอย่างการประกาศตัวแปรชุด 1 มิติ

```
import java.util.Scanner;
public class TestArray {
     final static int MAX_SIZE=25;
     public static void main(String[] args){
        int list[]  = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                        21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
                        41, 42, 43, 44, 45
                        };
         int numPrinted;
         numPrinted = 0;
         for (int i = 0; i < MAX_SIZE; i++)
         {
            System.out.format("%3d", list[i]);
            if (numPrinted < 9)
               numPrinted++;
            else
            {
              System.out.format("\n");
              numPrinted = 0;
            }
         }

    }
}
```

```
/* Results:
  1  2  3  4  5  6  7  8  9 10
 21 22 23 24 25 26 27 28 29 30
 41 42 43 44 45
*/
```

ตัวอย่างการใช้งานตัวแปรชุด 1 มิติ

```java
import java.util.Scanner;
public class TestArray2 {
     final static int ARY_SIZE =5;
     public static void main(String[] args){
        // Local Declarations
        int[] sqrAry= new int[ARY_SIZE];
        // Statements
        for (int i = 0; i < ARY_SIZE; i++)
              sqrAry[i] = i * i;
        System.out.println("Element\tSquare\n");
        System.out.println("=======\t======\n");
        for (int i = 0; i < ARY_SIZE; i++)
            System.out.format("%5d\t%4d\n", i, sqrAry[i]);

     }
}
```

```
/* Results:
Element Square
======= ======
    0      0
    1      1
    2      4
    3      9
    4     16
*/
```

ตัวอย่างการใช้งานตัวแปรชุด 1 มิติ

```
import java.util.Scanner;
public class TestArray3 {
     final static int ARY_SIZE =5;
     public static void main(String[] args){
        Scanner in= new Scanner(System.in);
        int readNum;
        int[] numbers= new int[50];
        System.out.print("You may enter up to 50 integers:\n");
        System.out.print("How many would you like to enter? ");
        readNum= in.nextInt();
        if (readNum > 50)
          readNum = 50;
          System.out.print("\nEnter your numbers: \n");
          for (int i = 0; i < readNum; i++)
              numbers[i]=in.nextInt();
          System.out.print("\nYour numbers reversed are: \n");
          for (int i = readNum - 1, numPrinted = 0;i >= 0;i--)
          {
               System.out.format("%3d", numbers[i]);
               if (numPrinted < 9)
                  numPrinted++;
               else
               {
                   System.out.println("\n");
                   numPrinted = 0;
               }
          }
      }
}
```

```
/* Results:
You may enter up to 50 integers:
How many would you like to enter? 12

Enter your numbers:
1 2 3 4 5 6 7 8 9 10 11 12

Your numbers reversed are:
 12 11 10  9  8  7  6  5  4  3
  2  1
*/
```

ตัวอย่างการใช้งานตัวแปรชุด 1 มิติ

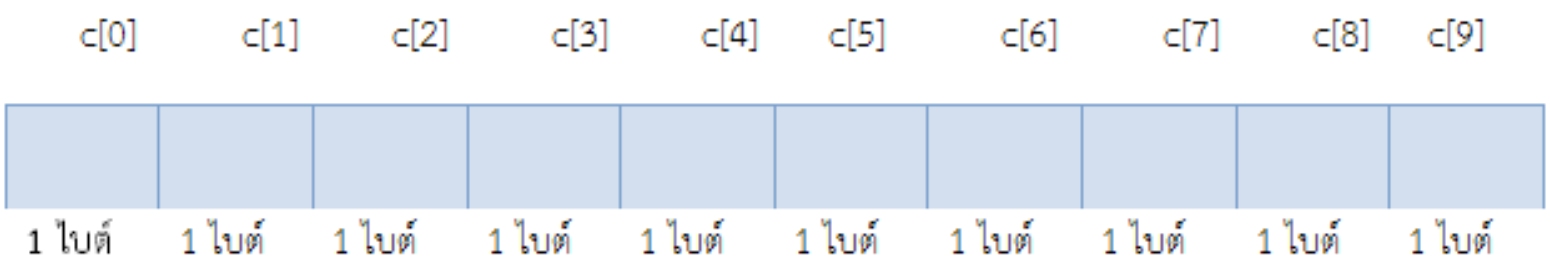
```
import java.util.Scanner;
public class TestArray4 {
     public static void main(String[] args){
         Scanner in= new Scanner(System.in);
         int i, n;
         float[] f= new float[40];
         f[0]=0;
         f[1]=1;
         System.out.println("Enter N:");
         n=in.nextInt();
         for(i=2;i<n;i++){
             f[i]=f[i-1]+f[i-2];
         }
         for(i=0;i<n;i++){
             System.out.format("f[%d]=%.0f\n", i, f[i]);
         }
     }
}
```

ตัวอย่างการใช้งานตัวแปรชุด 1 มิติ

```
import java.util.Scanner;
public class TestArray5 {
     public static void main(String[] args){
        Scanner in= new Scanner(System.in);
        int i, max, n;
        int[] x= new int[100];
        System.out.println("Enter N:");
        n=in.nextInt();
        for(i=0;i<n;i++){
           System.out.format("x[%d]=",i);
           x[i]=in.nextInt();
        }
        max=x[0];
        for(i=1;i<n;i++){
            if (max<x[i]){
                max=x[i];
            }
        }
        System.out.format("Max =%d\n", max);
     }
}
```

การประกาศตัวแปรชุด 1 มิติของข้อมูลประเภท char

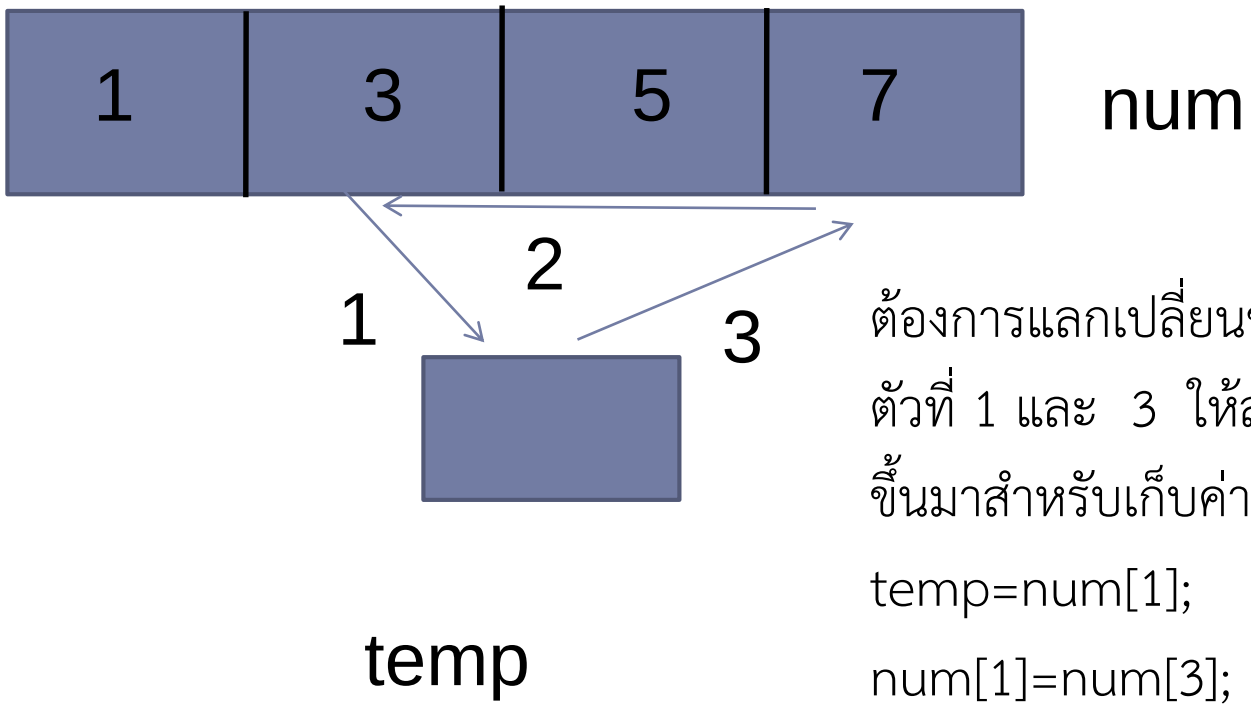
- char[] c = new char[10]; เป็นการประกาศตัวแปร array ชื่อ c เป็น array ของ ตัวแปรอักขระ char มีสมาชิกได้ 10 ตัว คือ a[0] a[1] … a[9] โดยการใช้เนื้อที่ในหน่วยความจำเปรียบเทียบได้ ดังรูป



โดยที่สมาชิกแต่ละตัวใช้เนื้อที่ 1 ไบต์ ดังนั้น array ประเภท char ชุดนี้ ใช้เนื้อที่เป็น10 bytes

การแลกเปลี่ยนข้อมูลระหว่างกันของอาร์เรย์

- int[] num = {1,3,5,7}



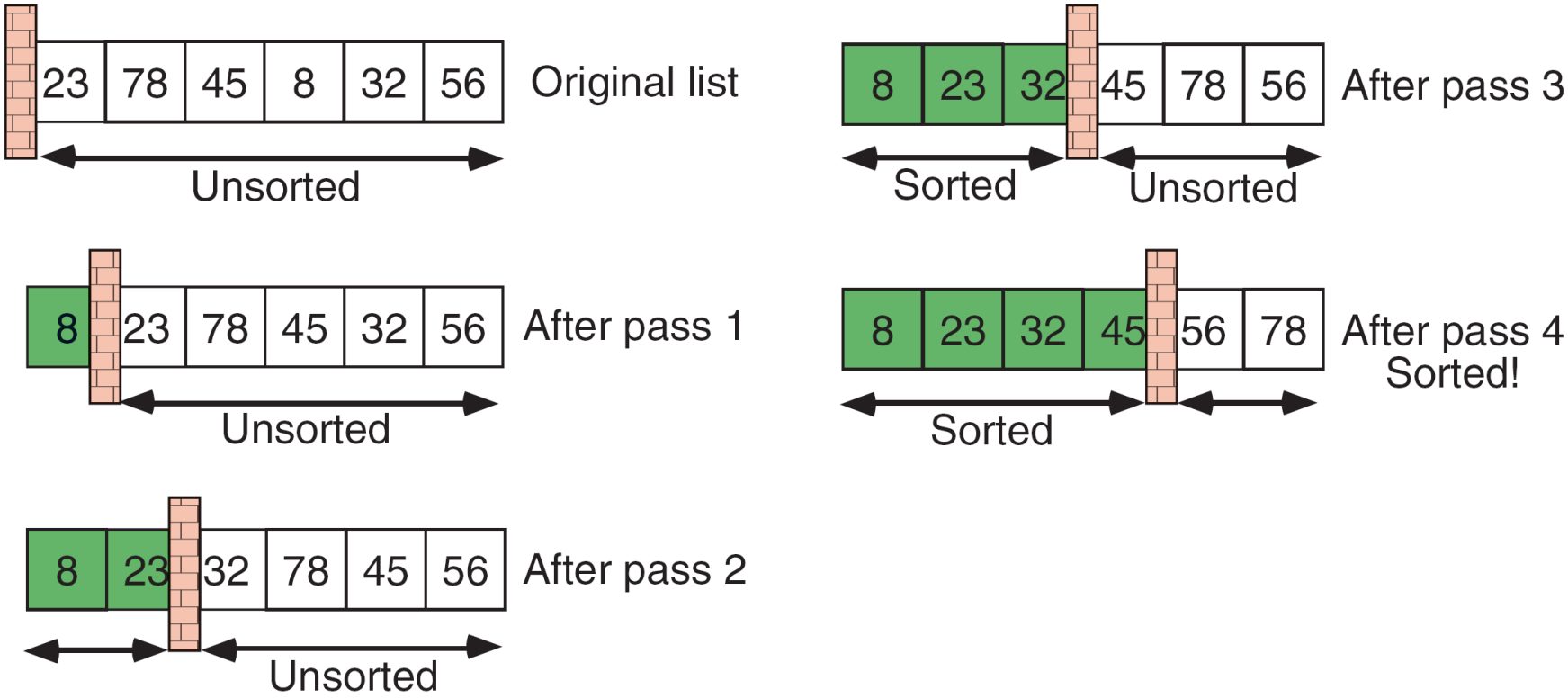
ต้องการแลกเปลี่ยนข้อมูลระหว่างอาร์เรย์ตัวที่ 1 และ 3 ให้สร้างตัวแปรชั่วคราวขึ้นมาสำหรับเก็บค่า จะได้การทำงานดังนี้

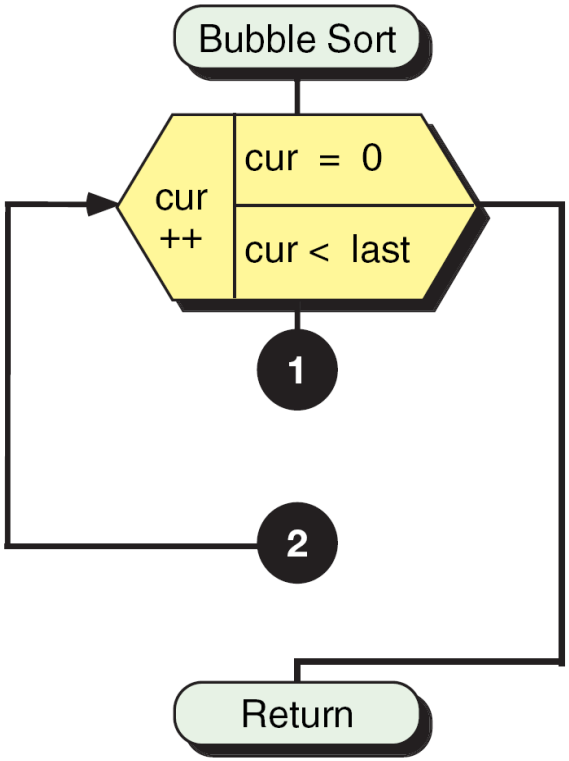
temp=num[1];

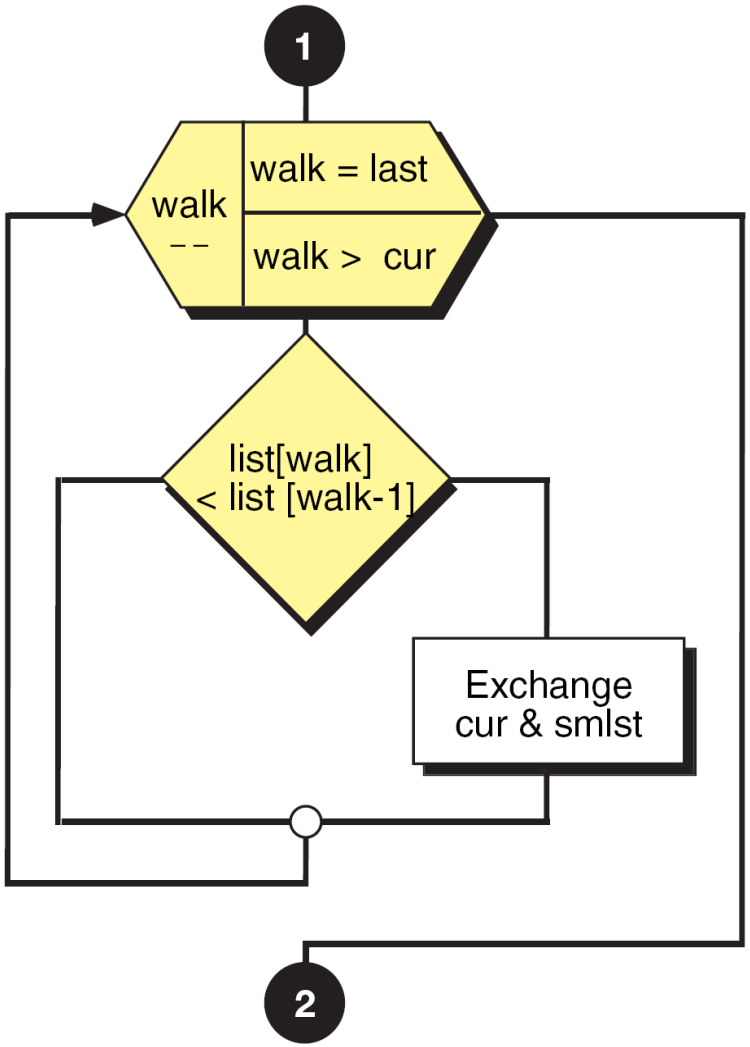
num[1]=num[3];

num3=temp

ตัวอย่างการประยุกต์ใช้งานอาร์เรย์กับการเรียงลำดับ




Bubble Sort
cur
++
cur = 0
cur < last
1
2
Return


1
walk
– –
walk = last
walk > cur
list[walk]
< list [walk-1]
Exchange
cur & smlst
2

```
public class NewClass16 {
    public static void main(String[] args) {
        int arr[] ={3,60,35,2,45,320,5};
        System.out.println("Array Before Bubble Sort");
        for(int i=0; i < arr.length; i++){
            System.out.print(arr[i] + " ");
        }
        System.out.println();
       int n = arr.length;
               int temp = 0;
                for(int i=0; i < n; i++){
                   for(int j=n-1; j >i ; j--){
                       if(arr[j] < arr[j-1]){
                           temp = arr[j];
                           arr[j] = arr[j-1];
                           arr[j-1] = temp;
                       }
                   }
                }
               System.out.println("Array After Bubble Sort");
               for(int i=0; i < arr.length; i++){
                   System.out.print(arr[i] + " ");
               }
        }
    }
```

Two-dimensional Arrays

```
// Declare array ref var
dataType[][] refVar;

// Create array and assign its reference to variable
refVar = new dataType[10][10];

// Combine declaration and creation in one statement
dataType[][] refVar = new dataType[10][10];

// Alternative syntax
dataType refVar[][] = new dataType[10][10];
```

Declaring Variables of Two-dimensional Arrays and Creating Two-dimensional Arrays

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

or

```
int matrix[][] = new int[10][10];
matrix[0][0] = 3;

for (int i = 0; i < matrix.length; i++)
  for (int j = 0; j < matrix[i].length; j++)
    matrix[i][j] = (int)(Math.random() * 1000);

double[][] x;
```

Two-dimensional Array Illustration

	0	1	2	3	4
0					
1					
2					
3					
4					

```
matrix = new int[5][5];
```

	0	1	2	3	4
0					
1					
2		7			
3					
4					

```
matrix[2][1] = 7;
```

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12

```
int[][] array = {
   {1, 2, 3},
   {4, 5, 6},
   {7, 8, 9},
   {10, 11, 12}
};
```

matrix.length? 5

matrix[0].length? 5

array.length? 4

array[0].length? 3

Declaring, Creating, and Initializing Using Shorthand Notations

เราสามารถประกาศ สร้าง และกำหนดค่าให้อาร์เรย์สองมิติได้ ดังตัวอย่าง

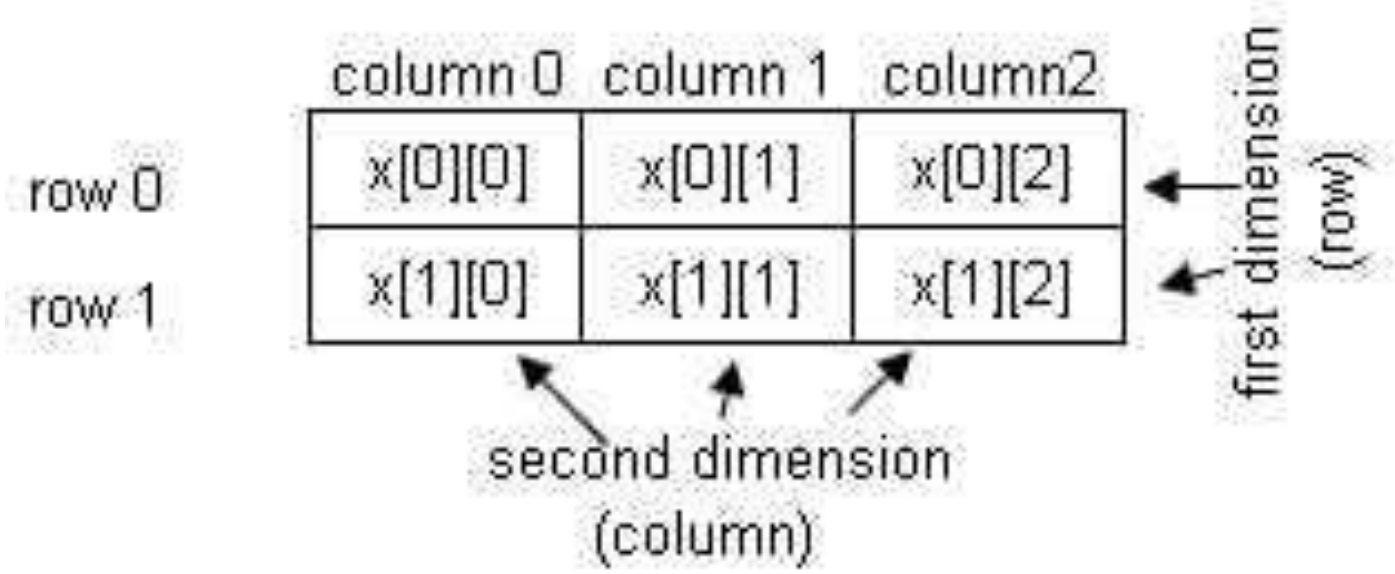
```
int[][] array = {
  {1, 2, 3},
  {4, 5, 6},
  {7, 8, 9},
  {10, 11, 12}
};
```

Same as

```
int[][] array = new int[4][3];
array[0][0] = 1; array[0][1] = 2; array[0][2] = 3;
array[1][0] = 4; array[1][1] = 5; array[1][2] = 6;
array[2][0] = 7; array[2][1] = 8; array[2][2] = 9;
array[3][0] = 10; array[3][1] = 11; array[3][2] =
    12;
```

การประกาศและกำหนดค่าตัวแปรชุด 2 มิติ(two dimension arrays)

- array 2 มิติ มีการจัดการจัดเก็บเปรียบเทียบคล้ายกับ ตาราง 2 มิติ มิติที่ 1 เปรียบเหมือนแถว(row) ของตาราง มิติที่ 2 เปรียบคล้ายกับสดมภ์(column)ของตาราง ดังรูป



การประกาศและกำหนดค่าตัวแปรชุด 2 มิติ(two dimension arrays)

```
import java.util.Scanner;
public class TestArray6 {
     public static void main(String[] args){
        Scanner in= new Scanner(System.in);
        int[][] table= new int[12][11];
        int row , col ;
        System.out.print("*** multiplication table ***");
        for(row=1;row <= 12;row++){
            System.out.print("\n");
            for(col=2;col<=12;col++){
                table[row-1][col-2]=row*col;
                System.out.format(" %3d ",table[row-1][col-2]);
            }
        }
     }
}
```

การประกาศและกำหนดค่าตัวแปรชุด 2 มิติ(two dimension arrays)

```
import java.util.Scanner;
public class TestArray7 {
     public static void main(String[] args){
         Scanner in= new Scanner(System.in);
         int i, j, n=3;
         int[][] C= new int[3][3];
         int A[][]={{1,2,3}, {4,5,6},{7,8,9}};
         int B[][]={{5,6,7}, {8,9,10},{11,12,13}};
         for(i=0;i<n;i++){
             for(j=0;j<n;j++){
                 C[i][j]= A[i][j]+B[i][j];
             }
         }
         System.out.print("Result:\n");
         for(i=0;i<n;i++){
             for(j=0;j<n;j++){
                  System.out.format("%d ",C[i][j]);
             }
             System.out.print("\n");
         }
     }
}
```

การประกาศและกำหนดค่าตัวแปรชุด 3 มิติ(Three dimension arrays)

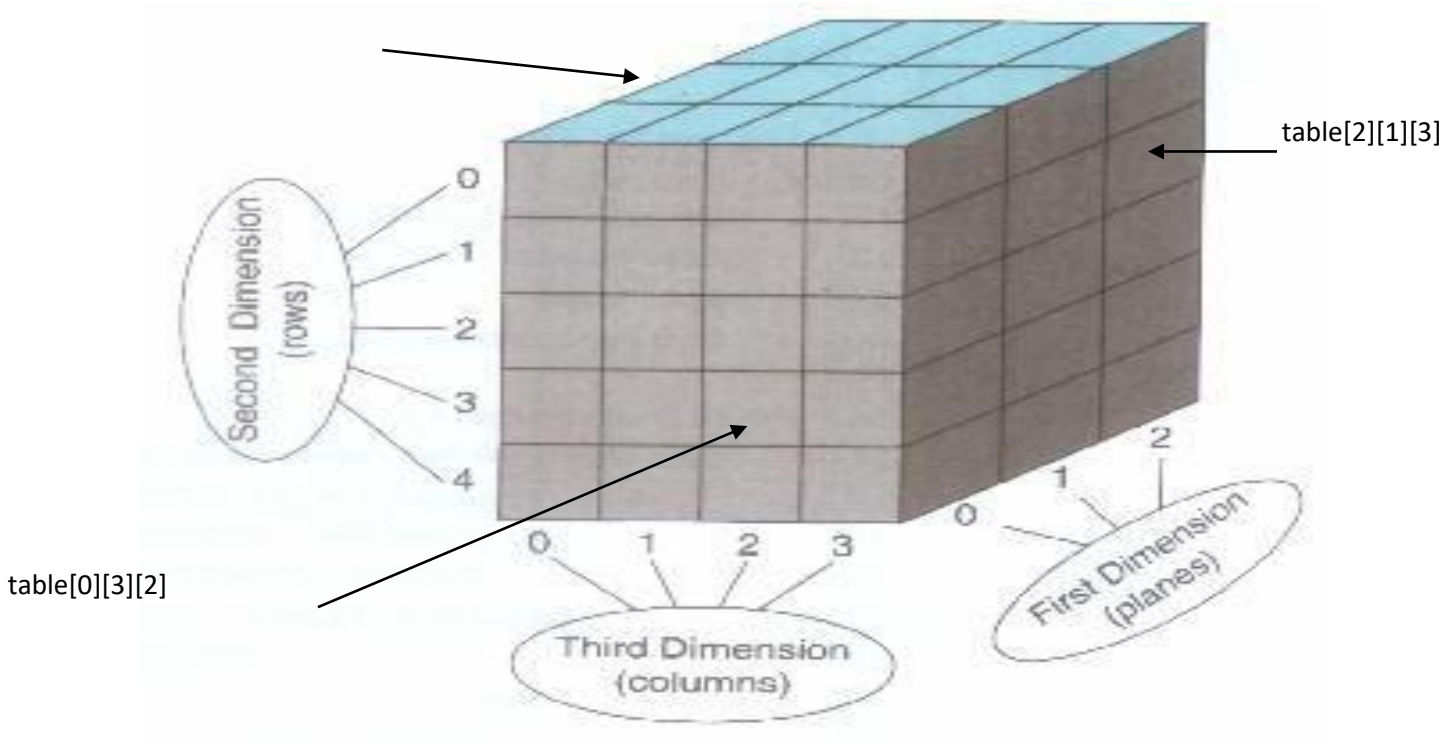
- **ตัวแปร array 3 มิติ** มีการประกาศ ดังนี้

```
type[][][] arrayname= new type[p] [r][c];
```

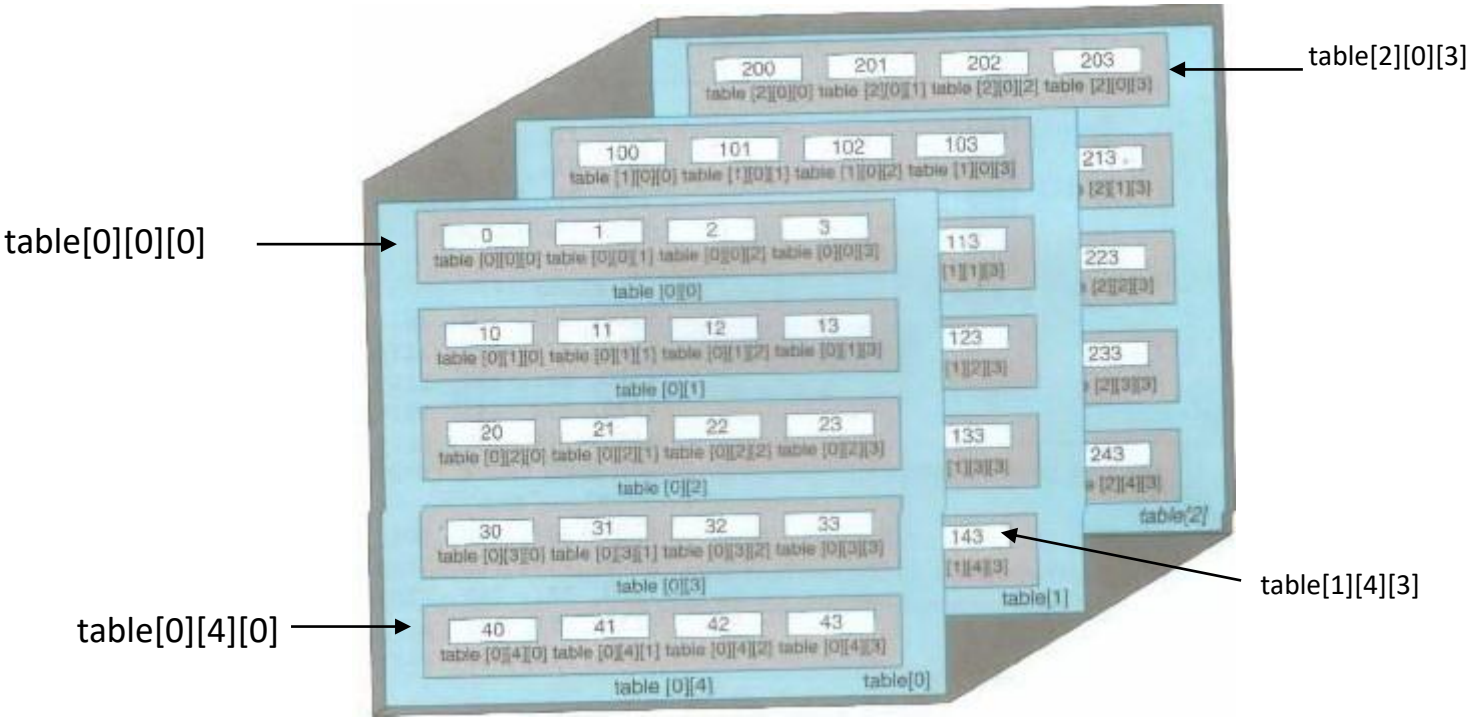
  - type คือ ชนิดของตัวแปร เช่น int ,float,char
  - arrayname คือชื่อของตัวแปร
  - r,c,p คือตัวเลขแสดงจำนวนในมิติที่ 1 มิติที่ 2 และมิติที่ 3 ของ array ตามลำดับ
  - โดยตัวเลขกำกับตำแหน่ง(ดัชนี)เป็นดังนี้
    - p เป็น 0,1,2 .. , p-1
    - r เป็น 0,1,2, ... ,r-1
    - c เป็น 0,1,2 ... ,c-1

การประกาศและกำหนดค่าตัวแปรชุด 3 มิติ(Three dimension arrays)

- ลักษณะของ array 3 มิติ อาจเปรียบเทียบว่าเป็น arrays of arrays ดังรูป



การประกาศและกำหนดค่าตัวแปรชุด 3 มิติ(Three dimension arrays)



การประกาศและกำหนดค่าตัวแปรชุด 3 มิติ(Three dimension arrays)

```
import java.util.Scanner;
public class TestArray8 {
     public static void main(String[] args){
        Scanner in= new Scanner(System.in);
        int[][][] arr= new int[3][4][5];
        int i, j, k, sum = 0;
        for(i = 0; i < 3; i++){
           for(j = 0; j < 4; j++){
               for(k = 0; k < 5; k++)
               {
                   arr[i][j][k]=in.nextInt();
                   sum = sum + arr[i][j][k];
               }
           }
         }
          System.out.format("sum is %d", sum);

     }
}
```

ตัวอย่างการใช้อาร์เรย์

Searching Arrays

Key	List							
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8

Binary Search

key is 11

key < 50

low mid high

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]

list 2 4 7 10 11 45 **50** 59 60 66 69 70 79

low mid high

[0] [1] [2] [3] [4] [5]

key > 7 list 2 4 7 10 11 45

low mid high

[3] [4] [5]

key == 11 list 10 11 45

Key	List							
8	1	2	3	4	6	7	8	9
8	1	2	3	4	6	7	8	9
8	1	2	3	4	6	7	8	9

key is 54

key > 50

low mid high

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]

list 2 4 7 10 11 45 **50** 59 60 66 69 70 79

low mid high

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]

key < 66 list 59 60 66 69 70 79

low mid high

[7] [8]

key < 59 list 59 60

low high

[6] [7] [8]

59 60

From Idea to Soluton

```
import java.util.Scanner;
public class NewClass17 {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int key, i;
        boolean found=false;
        int ansInd=0;
        int low = 0;
        int high = 7;
        int mid;
        int unsortedlist[] = {1, 4, 9, 3, 5, -3, 6, 2};
        int list[] = {-3,1, 2, 3, 4, 5, 6, 9 };
        System.out.println("Please enter number to be search:");
        key=sc.nextInt();
```

```
while(low<=high){
            mid = (low + high) / 2;
            System.out.format("mid=%d\n",mid);
            if (key == list[mid]){
                ansInd= mid;
                found=true;
                break;
            }
            else if (key < list[mid]){
                high = mid - 1;
            }
            else{
                low = mid + 1;
            }
        }
        if(found){
            System.out.format("%d is in position %d\n",key,ansInd+1);
        }
        else{
            System.out.print("Not found\n");
        }
    }
}
```

Selection Sort

swap

2 9 5 4 8 1 6

Select 9 (the largest) and swap it with 6 (the last) in the list

swap

2 6 5 4 8 1 9

The number 9 now is in the correct position and thus no longer need to be considered.

Select 8 (the largest) and swap it with 1 (the last) in the remaining list

swap

2 6 5 4 1 8 9

The number 8 now is in the correct position and thus no longer need to be considered.

Select 6 (the largest) and swap it with 1 (the last) in the remaining list

swap

2 1 5 4 6 8 9

The number 6 now is in the correct position and thus no longer need to be considered.

Select 5 (the largest) and swap it with 4 (the last) in the remaining list

2 1 4 5 6 8 9

The number 5 now is in the correct position and thus no longer need to be considered.

4 is the largest and last in the list. No swap is necessary

swap

2 1 4 5 6 8 9

The number 4 now is in the correct position and thus no longer need to be considered.

Select 2 (the largest) and swap it with 1 (the last) in the remaining list

1 2 4 5 6 8 9

The number 2 now is in the correct position and thus no longer need to be considered.

Since there is only one number in the remaining list, sort is completed

From Idea to Solution

```
for (int i = list.length - 1; i >= 1; i--) {
  select the largest element in list[0..i];
  swap the largest with list[i], if necessary;
  // list[i] is in place. The next iteration applies on list[0..i-1]
}
```

```
list[0] list[1] list[2] list[3] ...                list[10]
```

```
list[0] list[1] list[2] list[3] ...              list[9]
```

```
list[0] list[1] list[2] list[3] ...            list[8]
```

```
list[0] list[1] list[2] list[3] ...          list[7]
```

```
...
```

```
list[0]
```

Wrap it in a Method

```
/** The method for sorting the numbers */
public static void selectionSort(double[] list) {
  for (int i = list.length - 1; i >= 1; i--) {
    // Find the maximum in the list[0..i]
    double currentMax = list[0];
    int currentMaxIndex = 0;

    for (int j = 1; j <= i; j++) {
      if (currentMax < list[j]) {
        currentMax = list[j];
        currentMaxIndex = j;
      }
    }

    // Swap list[i] with list[currentMaxIndex] if necessary;
    if (currentMaxIndex != i) {
      list[currentMaxIndex] = list[i];
      list[i] = currentMax;
    }
  }
}
```

Insertion Sort

int[] myList = {2, 9, 5, 4, 8, 1, 6}; // Unsorted

The insertion sort algorithm sorts a list of values by repeatedly inserting an unsorted element into a sorted sublist until the whole list is sorted.

Step 1: Initially, the sorted sublist contains the first element in the list. Insert 9 to the sublist.

2 9 5 4 8 1 6

Step2: The sorted sublist is {2, 9}. Insert 5 to the sublist.

2 9 5 4 8 1 6

Step 3: The sorted sublist is {2, 5, 9}. Insert 4 to the sublist.

2 5 9 4 8 1 6

Step 4: The sorted sublist is {2, 4, 5, 9}. Insert 8 to the sublist.

2 4 5 9 8 1 6

Step 5: The sorted sublist is {2, 4, 5, 8, 9}. Insert 1 to the sublist.

2 4 5 8 9 1 6

Step 6: The sorted sublist is {1, 2, 4, 5, 8, 9}. Insert 6 to the sublist.

1 2 4 5 8 9 6

Step 7: The entire list is now sorted

1 2 4 5 6 8 9