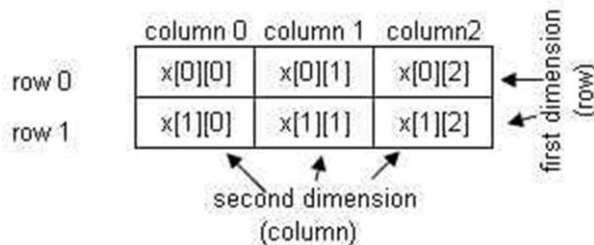


## แบบฝึกหัดปฏิบัติการคาบที่ 5: Multidimensional array and ArrayList

## คำสั่ง

1. ให้ศึกษาการสร้างอาร์เรย์จากตัวอย่างต่อไปนี้

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



ตัวแปร array 3 มิติ มีการประกาศ ดังนี้

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

type คือ ชนิดของตัวแปร เช่น int ,float,char

arrayname คือชื่อของตัวแปร

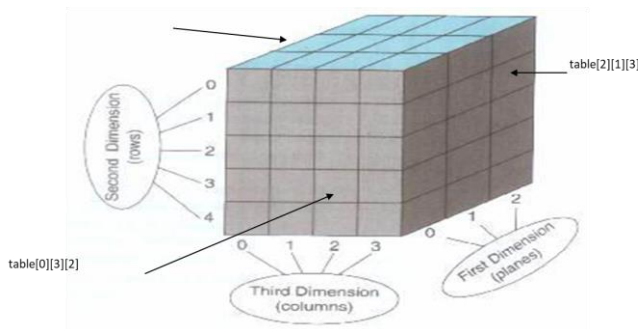
r,c,p คือตัวเลขแสดงจำนวนในมิติที่ 1 มิติที่ 2 และมิติที่ 3 ของ array ตามลำดับ

โดยตัวเลขกำกับตำแหน่ง(ดัชนี)เป็นดังนี้

p เป็น 0,1,2 .. , p-1

r เป็น 0,1,2, ... ,r-1

c เป็น 0,1,2 ... ,c-1



## Array Lists

- The ArrayList class manages a sequence of objects
- Can grow and shrink as needed
- ArrayList class supplies methods for many common tasks, such as inserting and removing elements
- The ArrayList class is a generic class: ArrayList<T> contains objects of type T:

```
ArrayList<BankAccount> accounts = new
ArrayList<BankAccount>();
accounts.add(new BankAccount(1001));
accounts.add(new BankAccount(1015));
accounts.add(new BankAccount(1022));
```

- size method yields number of elements
- Retrieving Array List Elements

- Use get method
- Index starts at 0
- `BankAccount anAccount = accounts.get(2);` // gets the third element of the array list
- Bounds error if index is out of range
- Most common bounds error:
  - `int i = accounts.size();`  
`anAccount = accounts.get(i);` // Error  
 //legal index values are 0. . i-1
- **Adding Elements**
- set overwrites an existing value
  - `BankAccount anAccount = new BankAccount(1729);`  
`accounts.set(2, anAccount);`
- add adds to the end
  - `accounts.add(new BankAccount(1001));`
- add can also be used to insert a new value at a given position (this pushes items at positions 1 to size-1 back by one):
  - `accounts.add(1, new BankAccount(1008));`

1. ให้เขียนโปรแกรมที่อ่านข้อมูลประกอบด้วยตัวเลขและเก็บข้อมูลดังกล่าวใน list หลังจากนั้นให้ประมวลผลตามรายการต่อไปนี้

พิมพ์ค่าเฉลี่ยของตัวเลขทั้งหมด  
 พิมพ์ตัวเลขที่มีค่าสูงสุดและต่ำสุด  
 ลบสมาชิกที่เป็นเลขคู่ออกจาก list

**ข้อมูลนำเข้า**

ข้อมูลที่ประกอบด้วยตัวเลข

**ข้อมูลส่งออก**

พิมพ์ค่าเฉลี่ยของตัวเลขทั้งหมด  
 พิมพ์ตัวเลขที่มีค่าสูงสุดและต่ำสุด  
 ลบสมาชิกที่เป็นเลขคู่ออกจากlist list

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
5 9 2 9 1 2 8 9 1 6	4.98
9 1 3 9 8 4 2 1 5 7	9
2 7 9 3 8 5 2 7 6 8	1
1 6 2 1 7 7 1 9 4 1	5 9 9 1 9 1 9 1 3 9 1 5 7 7 9 3 5 7 1 1 7 7 1 9 1 5 3 9 5 3 3
8 5 2 3 9 8 5 6 3 3	

## 2. ให้เขียนคลาส Star ที่ประกอบด้วยเมธอดต่อไปนี้

เมธอด addStars ที่รับ array list of strings เป็นพารามิเตอร์และเพิ่ม \* หลังสมาชิกแต่ละตัว เช่น array list ชื่อ list ที่มีสมาชิกจัดเก็บดังนี้ [the, quick, brown, fox] เมื่อเรียก addStars(list); ให้ array list มีสมาชิกดังนี้ [the, \*, quick, \*, brown, \*, fox, \*]

เมธอด removeStars ที่รับ array list of strings เป็นพารามิเตอร์และลบ \* หลังสมาชิกแต่ละตัว เช่น array list ชื่อ list ที่มีสมาชิกจัดเก็บดังนี้ [the, \*, quick, \*, brown, \*, fox, \*] เมื่อเรียก removeStars(list); ให้ array list มีสมาชิกดังนี้ [the, quick, brown, fox]

## ข้อมูลนำเข้า

คำแต่ละคำ

บรรทัดต่อมาแสดงจำนวนคำสั่ง n คำสั่ง ตามด้วยคำสั่งเมื่อ 1 คือ addStars 2 คือ removeStars

## ข้อมูลส่งออก

พิมพ์รายการสมาชิกใน ArrayList ภายหลังการประมวลผลคำสั่งแต่ละคำสั่ง

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
the quick brown fox 1 2	[the, *, quick, *, brown, *, fox, *] [the, quick, brown, fox]

## 3. ให้เขียนคลาส SortMergeList ที่ประกอบด้วยเมธอดต่อไปนี้

เขียน method intersect ที่ รับพารามิเตอร์ 2 ตัวที่เป็น array lists of integers สองรายการที่เรียงลำดับและส่งกลับรายการใหม่ที่มีแสดงเฉพาะสมาชิกที่พบในรายการทั้งสอง

ตัวอย่าง: ถ้ารายการชื่อ list1 และ list2 เริ่มต้นจัดเก็บ:

[1, 4, 8, 9, 11, 15, 17, 28, 41, 59]

[4, 7, 11, 17, 19, 20, 23, 28, 37, 59, 81]

ดังนั้นการเรียกใช้ intersect (list1, list2) จะส่งคืนรายการ: [4, 11, 17, 28, 59]

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
1 4 8 9 11 15 17 28 41 59 0 4 7 11 17 19 20 23 28 37 59 81 0	4 11 17 28 59

#### 4. [Algorithms]

ร้านคอมพิวเตอร์ K.I.B. ต้องการขยายฐานลูกค้าไปยังเมืองใหม่ โดยเมืองดังกล่าวมีการวางผังเมืองเป็นพื้นที่สี่เหลี่ยมย่อยจำนวน  $M \times N$  พื้นที่ (M แถว N หลัก) และจากการสำรวจสำมะโนประชากรทำให้ทราบจำนวนประชากรในแต่ละพื้นที่ (ดูภาพประกอบด้านล่าง)

เนื่องจากร้าน K.I.B. ต้อง การเปิดศูนย์บริการลูกค้าเพียงร้านเดียวในเมืองนี้ ยิ่งไปกว่านั้นพื้นที่บริการที่ร้านให้บริการลูกค้าได้จะครอบคลุมบริเวณที่ ประกอบด้วยสี่เหลี่ยมย่อยจำนวน  $K \times K$  พื้นที่ (K แถว K หลัก) เท่านั้น ทางร้านจึงพยายามหาพื้นที่บริการที่ดีที่สุด ซึ่งในที่นี้หมายถึงพื้นที่บริการที่มีประชากรรวมกันมากที่สุด

5	9	2	9	1	2	8	9	1	6
9	1	3	9	8	4	2	1	5	7
2	7	9	3	8	5	2	7	6	8
1	6	2	1	7	7	1	9	4	1
8	5	2	3	9	8	5	6	3	3

ภาพประกอบตัวอย่างโจทย์ แสดงผลการหาทำเลตั้งศูนย์บริการลูกค้าในพื้นที่ขนาด  $2 \times 2$  ( $K = 2$ ) ของผังเมืองขนาด  $5 \times 10$  ในที่นี้บริเวณที่ถูกเน้นคือพื้นที่บริการที่ดีที่สุด

จงเขียนโปรแกรมที่มีประสิทธิภาพในการหาจำนวนประชากรรวมในทำเลพื้นที่บริการที่ดีที่สุด

##### ข้อมูลนำเข้า

- บรรทัดแรกเป็นเลขจำนวนเต็มบวกสองตัวบอกจำนวนแถว (M) และจำนวนหลัก (N) ตามลำดับ โดยที่  $2 \leq M, N \leq 1000$
- บรรทัดที่สองระบุขนาดพื้นที่บริการของร้าน (K) โดยที่  $0 < K < M$  และ  $0 < K < N$
- บรรทัดที่สามถึง  $M+2$  ระบุจำนวนประชากรในแถวที่ 1 ถึง M ตามลำดับ ข้อมูลแต่ละบรรทัดประกอบด้วยตัวเลขจำนวนเต็มบวก N จำนวน ซึ่งระบุจำนวนประชากรของพื้นที่สี่เหลี่ยมย่อย N หลัก เรียงจากซ้ายไปขวาในแถวนั้น ๆ แต่ละจำนวนถูกคั่นด้วยช่องว่าง โดยประชากรในแต่ละพื้นที่สี่เหลี่ยมย่อยมีจำนวนไม่เกิน 2,000 คน

##### ข้อมูลส่งออก

จำนวนประชากรภายในพื้นที่บริการที่ดีที่สุด

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
5 10 2 5 9 2 9 1 2 8 9 1 6 9 1 3 9 8 4 2 1 5 7 2 7 9 3 8 5 2 7 6 8 1 6 2 1 7 7 1 9 4 1 8 5 2 3 9 8 5 6 3 3	31

## 5. [Application]

จงสร้างคลาส Employee ที่ประกอบด้วยสมาชิกต่อไปนี้

- มี Instance variable แบบ **private ประเภท** String ชื่อ firstname สำหรับเก็บชื่อพนักงาน
- มี Instance variable แบบ **private ประเภท** String ชื่อ lastname สำหรับเก็บนามสกุลพนักงาน
- มี Instance variable แบบ **private ประเภท** String ชื่อ id สำหรับเก็บรหัสพนักงาน
- มี Instance variable แบบ **private ประเภท** double ชื่อ salary สำหรับบันทึกค่าเงินเดือนของพนักงาน
- มี Constructor ที่รับข้อมูลดังนี้

```
public Employee(String firstname,String lastname,String id, double sal)
```

Method earning () มีการประมวลผลโดยนำเงินเดือนของพนักงานแต่ละคนมาทำการหักภาษี 5% ของเงินเดือน แล้ว return เงินเดือนที่หักภาษีแล้ว

method bonus ( int year ) มีการประมวลผลโดยรับพารามิเตอร์ year ซึ่งเป็นระยะเวลาการทำงานของพนักงานเข้ามา เพื่อทำการตรวจสอบว่าถ้าระยะเวลาเกิน 5 ปี จะคืนค่าโบนัสเป็น 12 เท่าของเงินเดือน ไม่เช่นนั้นจะคืนค่าโบนัสเป็น 6 เท่าของเงินเดือน

จากนั้นทำการสร้าง ArrayList ชื่อ arrayEarn สำหรับเก็บข้อมูลของพนักงานทุกคน ท้ายสุดให้สร้าง method printEmp(ArrayList a) ที่ทำการพิมพ์รายงานที่มีรูปแบบดังด้านล่างนี้ออกทางจอภาพ

```
Fist name    Last name    Earning    Bonus
```

## 6. [Algorithms]

จงเขียนเมธอด isConsecutiveFour(int[] values) เพื่อตรวจสอบอาร์เรย์สองมิติต่อไปนี้ว่ามีตัวเลขตัวเดียวกันเรียงต่อกัน

ครบสี่ตัวหรือไม่ ในแนวนั่ง แนวนอน หรือแนวทะแยง **public boolean** isConsecutiveFour(int[] values)

จากนั้นให้เขียนโปรแกรมทดสอบที่ผู้ใช้สามารถป้อนจำนวนแถวและคอลัมน์ของอาร์เรย์สองมิติและป้อนค่าภายในอาร์เรย์ โดยเมื่อป้อนค่าไปแล้วให้เรียกใช้เมธอด isConsecutiveFour(int[] values) โดยการคืนค่า true กรณีที่ภายในอาร์เรย์มีตัวเลข 4 ตัวเรียงต่อกัน กรณีอื่นให้คืนค่าเป็น false

ตัวอย่าง เมื่อส่งอาร์เรย์ต่อไปนี้เข้าไปที่เมธอดจะคืนค่า true ออกมา

0 1 0 3 1 6 1 0 1 6 8 6 0 1 5 6 2 1 8 2 9 6 5 6 1 1 9 1 1 3 6 1 4 0 7 3 3 3 3 4 0 7	0 1 0 3 1 6 1 0 1 6 8 6 0 1 5 5 2 1 8 2 9 6 5 6 1 1 9 1 1 5 6 1 4 0 7 3 5 3 3 4 0 7	0 1 0 3 1 6 1 0 1 6 8 6 0 1 5 6 2 1 6 2 9 6 5 6 6 1 9 1 1 3 6 1 4 0 7 3 6 3 3 4 0 7	0 1 0 3 1 6 1 0 1 6 8 6 0 1 9 6 2 1 8 2 9 6 9 6 1 1 9 1 1 3 9 1 4 0 7 3 3 3 9 4 0 7
--	--	--	--

ข้อมูลนำเข้า จำนวนแถวและคอลัมน์ของอาร์เรย์สองมิติและป้อนค่าภายในอาร์เรย์

ข้อมูลส่งออก คืนค่า 1 กรณีที่ภายในอาร์เรย์มีตัวเลข 4 ตัวเรียงต่อกัน กรณีอื่นให้คืนค่าเป็น 0

ข้อมูลนำเข้า	ข้อมูลส่งออก
6 7 0 1 0 3 1 6 1 0 1 6 8 6 0 1 5 6 2 1 8 2 9 6 5 6 1 1 9 1 1 3 6 1 4 0 7 3 3 3 3 4 0 7	1