

CHAPTER
ANS-10
อาเรย์สองมิติ
(Two Dimensional Arrays)

โจทย์ข้อที่ 1 [ระดับง่าย]

- 1) `int a[][] = {{0,0,0},{0,0,0}};`
- 2) `int b[][] = {{1,1},{1,1},{1,1},{1,1},{1,1}};`
- 3) `String s[][] = {"Java","Java","Java"};`
- 4) `String t[][] = {"Java","Java","Java"};`
- 5) `double em[][] = {};`

โจทย์ข้อที่ 2 [ระดับง่าย]

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

1			
2	3		
3	4	5	
4	5	6	7

1	2	3		
2	3			
3	4	5	6	
4				
5	6			

โจทย์ข้อที่ 3 [ระดับง่าย]

- 1) `boolean matrix[][] = new boolean[5][8];`
- 2) `String chess[][] = new String[8][8];`
- 3) `int tranMatrix[][] = new int[4][9];`
- 4) `double data[][] = new double[300][3];`
- 5) `n[7] = new int[4];`
- 6) `n[4] = new int[x * x + 1];`

โจทย์ข้อที่ 4 [ระดับปานกลาง]

```
public class RainStatistic {
    public static double[][] getTable() {
        double rain[][] = new double[12][1];
        for (int i = 0; i < 12; i++) {
            if (i == 1) {
                rain[i] = new double[28];
                // rain[i] = new double[29];
            } else if (i == 3 || i == 5 || i == 8 || i == 10) {
                rain[i] = new double[30];
            } else {
                rain[i] = new double[31];
            }
        }
        return rain;
    }

    public static void main(String[] args) {
        double x[][] = getTable();

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

โจทย์ข้อที่ 5 [ระดับง่าย]

- 1) `int x = num[51][49];`
- 2) `char c = code[9][60];`
- 3) `int var1 = bank[0][1];`
- 4) `double var2 = bank[1][bank[1].length - 1];`
- 5) `code[6][4] = x;`
- 6) `sName[0][9] = s1;`
- 7) `sName[1][2] = s2;`
- 8) `a[a.length - 1][a[a.length - 1].length - 1] = y;`

โจทย์ข้อที่ 6 [ระดับง่าย]

```
public class The2DArray {
    public static void setArray(int n[][]) {
        for (int i = 0; i < n.length; i++) {
            for (int j = 0; j < n[i].length; j++) {
                n[i][j] = (i + 1) * (j + 1);
            }
        }
    }

    public static void main(String[] args) {
        int x[][] = new int[5][6];
        setArray(x);
    } //End of main
} //End of class
```

โจทย์ข้อที่ 7 [ระดับปานกลาง]

```
public class EqualityOf2DArray {
    public static boolean isArrayEquals(int x[][], int y[][]) {
        if (x.length == y.length) {
            for (int i = 0; i < x.length; i++) {
                if (x[i].length != y[i].length) return false;
            }
            for (int i = 0; i < x.length; i++) {
                for (int j = 0; j < x[i].length; j++) {
                    if (x[i][j] != y[i][j]) return false;
                }
            }
            return true;
        } else {
            return false;
        }
    }

    public static void main(String[] args) {
        int m[][] = {{1},{2,3},{3,4,5,6},{4,5,6,7}};
        int n[][] = {{1},{2,3},{3,4,5},{4,5,6,7}};
        boolean ch = isArrayEquals(m, n);
    } //End of main
} //End of class
```

โจทย์ข้อที่ 8 [ระดับปานกลาง - ระดับยาก]

```
import java.util.Scanner;
public class DataExperiment {
    public static double max(double t[][]) {
        double maxData = t[0][0];
        for (int i = 0; i < t.length; i++) {
            for (int j = 0; j < t[i].length; j++) {
                if (t[i][j] > maxData) maxData = t[i][j];
            }
        }
        return maxData;
    } //End of max(...)

    public static double min(double t[][]) {
        double minData = t[0][0];
        for (int i = 0; i < t.length; i++) {
            for (int j = 0; j < t[i].length; j++) {
                if (t[i][j] < minData) minData = t[i][j];
            }
        }
        return minData;
    } //End of min(...)

    public static double middleRange(double t[][]) {
        return (max(t) + min(t)) / 2.0;
    } //End of avgRange(...)

    public static double meanOfMonth(double t[][], int m) {
        double sum = 0.0;
        int i = (m + 3) % 12;
        for (int j = 0; j < t[i].length; j++) {
            sum += t[i][j];
        }
        return sum / t[i].length;
    } //End of meanOfMonth(...)

    public static double dataOfDay(double t[][], int d, int m) {
        return t[(m + 3) % 12][d - 1];
    } //End of dataOfDay(...)
```

```

public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    double test[][] = new double[12][];
    for (int i = 0; i < test.length; i++) {
        if (i == 5) {
            test[i] = new double[28];
        } else if (i == 0 || i == 2 || i == 7 || i == 9) {
            test[i] = new double[30];
        } else {
            test[i] = new double[31];
        }
        for (int j = 0; j < test[i].length; j++) {
            System.out.print((j + 1) + "/" + ((i + 9) % 12) + " : ");
            test[i][j] = kb.nextDouble();
        } //End of for j
    } //End of for i

    //===== Method Call =====
    System.out.println(max(test));
    System.out.println(min(test));
    System.out.println(middleRange(test));
    System.out.println(meanOfMonth(test, 10));
    System.out.println(dataOfDay(test, 13, 6));

} //End of main(...)
} //End of Class

```

โจทย์ข้อที่ 9 [ระดับง่าย - ระดับยาก]

```

import java.util.Scanner;
public class MatrixOperation {

```

//ระดับปานกลาง เขียนเมทอด addMatrix(...)

```

public static double[][] addMatrix(double x[][], double y[][]) {
    double add[][] = new double[x.length][x[0].length];
    for (int i = 0; i < x.length; i++) {
        for (int j = 0; j < x[i].length; j++) {
            add[i][j] = x[i][j] + y[i][j];
        }
    }
    return add;
}

```

//ระดับปานกลาง เขียนเมทอด mulScalarMatrix(...)

```

public static double[][] mulScalarMatrix(double x[][], double n) {
    double mulS[][] = new double[x.length][x[0].length];
    for (int i = 0; i < x.length; i++) {
        for (int j = 0; j < x[i].length; j++) {
            mulS[i][j] = n * x[i][j];
        }
    }
    return mulS;
}

```

//ระดับยาก เขียนเมทอด transposeMatrix(...)

```

public static double[][] transposeMatrix(double x[][]) {
    double at[][] = new double[x[0].length][x.length];
    for (int i = 0; i < x.length; i++) {
        for (int j = 0; j < x[i].length; j++) {
            at[j][i] = x[i][j];
        }
    }
    return at;
}

```

//ระดับยาก เขียนเมทอด mulMatrix(...)

```

public static double[][] mulMatrix(double x[][], double y[][]) {
    double mul[][] = new double[x.length][y[0].length];
    for (int i = 0; i < mul.length; i++) {
        for (int j = 0; j < mul[i].length; j++) {
            for (int k = 0; k < y.length; k++) {
                mul[i][j] += x[i][k] * y[k][j];
            }
        }
    }
    return mul;
}

```

//ระดับง่าย เขียนเมทอด printMatrix(...)

```

public static void printMatrix(double x[][]) {
    for (int i = 0; i < x.length; i++) {
        for (int j = 0; j < x[i].length; j++) {
            System.out.print(x[i][j] + "\t");
        }
        System.out.println();
    }
}

```

//ระดับง่าย เขียนเมทอด main(...)

```

public static void main(String[] args) {

```

```

//Define and creat matrix
double a[][] = {{1,3,5,9},{2,6,4,3}};
double b[][] = {{3,6},{4,8},{1,0}};
double c[][] = {{0,2,4},{5,3,1},{7,11,2},{6,6,9}};

//Calculate matrix m
double t1[][] = transposeMatrix(mulScalarMatrix(a, 2.0));
double t2[][] = transposeMatrix(mulScalarMatrix(b, 0.25));
double t3[][] = mulMatrix(t1, t2);
double t4[][] = mulScalarMatrix(c, -1.0);
double m[][] = addMatrix(t3, t4);

```

```
//Print matrix
System.out.println("----- Matrix a -----");
printMatrix(a);
System.out.println("----- Matrix b -----");
printMatrix(b);
System.out.println("----- Matrix c -----");
printMatrix(c);
System.out.println("----- Matrix m -----");
printMatrix(m);
```

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

โจทย์ข้อที่ 10 [ระดับยาก]

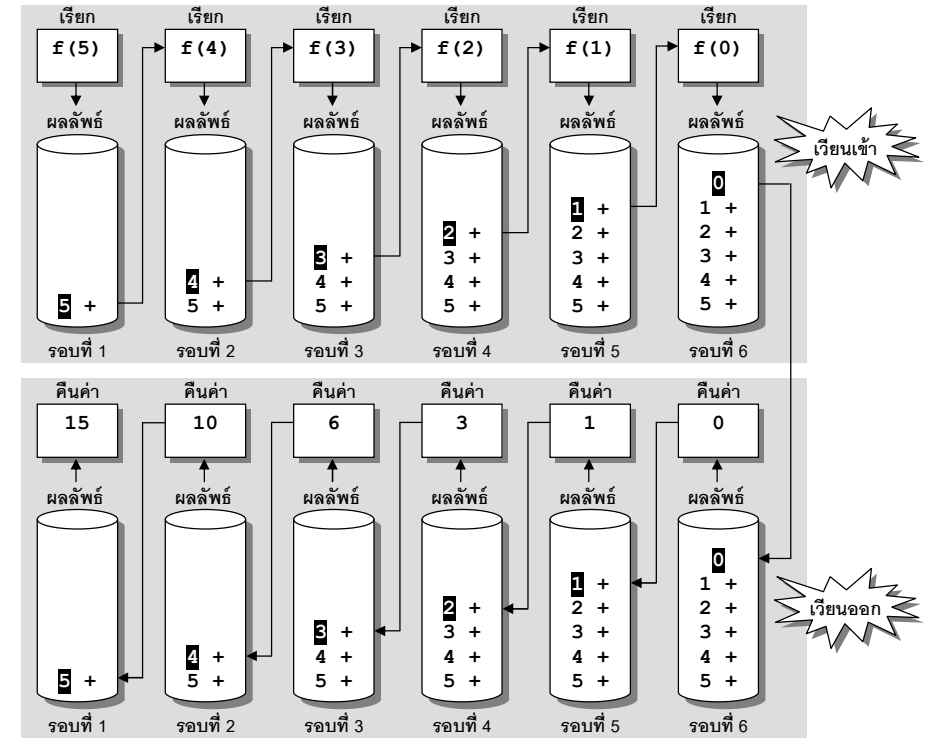
```
public static int[] splitRowOfArray(int a[][], int row) {
    int sp[];
    if (row > 0 && row <= a.length) {
        sp = new int[a[row - 1].length];
        for (int i = 0; i < sp.length; i++) {
            sp[i] = a[row - 1][i];
        }
    } else {
        sp = new int[0];
    }
    return sp;
}
```

โจทย์ข้อที่ 11 [ระดับยาก]

```
public static int[] splitColumnOfArray(int a[][], int col) {
    int sp[];
    if (col > 0 && col <= a[0].length) {
        sp = new int[a.length];
        for (int i = 0; i < sp.length; i++) {
            sp[i] = a[i][col - 1];
        }
    } else {
        sp = new int[0];
    }
    return sp;
}
```

CHAPTER
ANS-11การเวียนเกิด
(Recursion)

โจทย์ข้อที่ 1 [ระดับง่าย]



โจทย์ข้อที่ 2 [ระดับง่าย]

การเรียกใช้	คำตอบ
mul(4, 3)	12
mul(5, 7)	35
mul(20, 10)	200

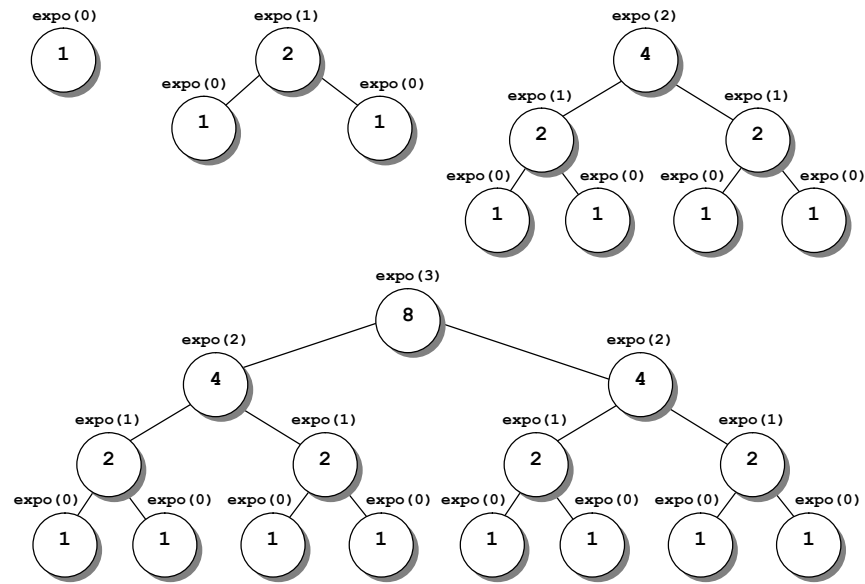
โจทย์ข้อที่ 3 [ระดับง่าย]

การเรียกใช้	คำตอบ
gcd(28, 16)	4
gcd(9, 14)	1
gcd(75, 30)	15

โจทย์ข้อที่ 4 [ระดับง่าย]

การเรียกใช้	คำตอบ
expo(4)	16
expo(7)	128
expo(11)	2048

จงแสดงรายละเอียดของ expo (0), expo (1), expo (2) และ expo (3)



โจทย์ข้อที่ 5 [ระดับง่าย]

การเรียกใช้และคำตอบ	
printX (4)	printX (7)
1234	1234567
123	123456
12	12345
1	1234
	123
	12
	1

โจทย์ข้อที่ 6 [ระดับปานกลาง]

พารามิเตอร์ m	พารามิเตอร์ n				
	0	1	2	3	4
0	1	2	3	4	5
1	2	3	4	5	6
2	3	5	7	9	11
3	5	13	29	61	125

โจทย์ข้อที่ 7 [ระดับง่าย]

$$f(n) = \sum_{i=1}^n \frac{1}{i^3} \quad n \geq 1$$

กรณีฐาน: $f(n) = 1 \quad n = 1$
 กรณีเวียนเกิด: $f(n) = f(n-1) + 1/n^3 \quad n > 1$

```
public static double f(int n) {
    if (n <= 1) {
        return 1.0;
    } else {
        return f(n - 1) + 1.0 / (n * n * n);
    }
}
```

โจทย์ข้อที่ 8 [ระดับง่าย]

```
public static int f(int n) {
    if (n <= 0) {
        return 0;
    } else {
        return f(n - 1) +
            (int) Math.pow(n, n) + (2 * n);
    }
}
```

โจทย์ข้อที่ 9 [ระดับง่าย]

```
public static double g(int n) {
    if (n <= 0) {
        return 0.0;
    } else if (n == 1) {
        return 1.0;
    } else {
        return g(n - 1) * (1.0 + 1.0 / n);
    }
}
```

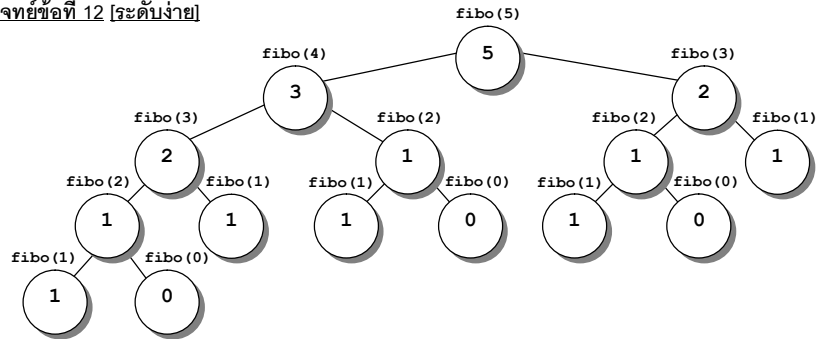
โจทย์ข้อที่ 10 [ระดับง่าย]

```
public static int fac(int n) {
    if (n < 2) {
        return 1;
    } else {
        return fac(n - 1) * n;
    }
}
```

โจทย์ข้อที่ 11 [ระดับง่าย]

```
public static int fibo(int n) {
    if (n < 2) {
        return n;
    } else {
        return fibo(n - 1) + fibo(n - 2);
    }
}
```

โจทย์ข้อที่ 12 [ระดับง่าย]



โจทย์ข้อที่ 13 [ระดับปานกลาง]

```
public static double pow(double a, int b) {
    if (b < 0) {
        return pow(a, b + 1) * 1.0 / a;
    } else if (b == 0) {
        return 1.0;
    } else {
        return pow(a, b - 1) * a;
    }
}
```

โจทย์ข้อที่ 14 [ระดับยาก]

```
public static double f(int n) {
    if (n <= 1) {
        return 1.0;
    } else {
        return 1.0 + (n / f(n - 1));
    }
}
```

โจทย์ข้อที่ 15 [ระดับยาก]

```
1) public static double g(int n, int m) {
    if (n <= 1) {
        return 1;
    } else {
        if (n % 2 == 0) {
            return g(n - 1, m) - Math.pow(n, m);
        } else {
            return g(n - 1, m) + Math.pow(n, m);
        }
    }
}
```

```
2) public static double h(int n) {
    if (n <= 1) {
        return 1;
    } else {
        if (n % 2 == 0) {
            return h(n - 1) + 1.0 / n;
        } else {
            return h(n - 1) - 1.0 / n;
        }
    }
}
```

โจทย์ข้อที่ 16 [ระดับปานกลาง]

```
public static int search(int a[], int k) {
    return search(a, k, 0);
}

public static int search(int a[], int k, int i) {
    if (i < a.length) {
        if (a[i] == k) {
            return i;
        } else {
            return search(a, k, i + 1);
        }
    } else {
        return -1;
    }
}
```

โจทย์ข้อที่ 17 [ระดับปานกลาง]

```
public class OverHundredNumber {
    public static void main(String[] args) {
        int d[] = { 99, 101, 13, 78, 200, 534, 47, 1234, 736 };
        System.out.println(overHundred(d));
    }
}
```

```
public static int overHundred(int d[]) {
    return overHundred(d, 0, 0);
}

public static int overHundred(int d[], int i, int count) {
    if (i < d.length) {
        if (d[i] > 100) {
            return overHundred(d, i + 1, count + 1);
        } else {
            return overHundred(d, i + 1, count);
        }
    } else {
        return count;
    }
}
```

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

โจทย์ข้อที่ 18 [ระดับปานกลาง]

```
public class TheString {
    public static void main(String[] args) {
        System.out.println(revStr("Computer"));
    }
}
```

```
public static String revStr(String s) {
    return revStr(s, "", s.length() - 1);
}

public static String revStr(String s, String rs, int i) {
    if (i >= 0) {
        rs += s.substring(i, i + 1);
        return revStr(s, rs, i - 1);
    } else {
        return rs;
    }
}
```

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

โจทย์ข้อที่ 19 [ระดับยาก]

```
public static void formulaAtoB(int a, int b) {
    formulaAtoB(a, b, 1);
}
```

```
public static void formulaAtoB(int a, int b, int i) {
    if (a <= b) {
        if (i <= 12) {
            System.out.println(a + " x " + i + " = " + (a * i));
            formulaAtoB(a, b, i + 1);
        } else {
            System.out.println("-----");
            formulaAtoB(a + 1, b);
        }
    } //End of if (a <= b)
}
```

โจทย์ข้อที่ 20 [ระดับยาก]

```
public static int[] addArray(int a[], int b[]) {
    int sum[] = new int[a.length];
    return addArray(a, b, sum, 0);
}

public static int[] addArray(int a[], int b[], int sum[], int i) {
    if (i < sum.length) {
        sum[i] = a[i] + b[i];
        return addArray(a, b, sum, i + 1);
    } else {
        return sum;
    }
}
```

โจทย์ข้อที่ 21 [ระดับยาก]

```
public static boolean isMatrixEquals(int a[][], int b[][]) {
    if (a.length == b.length && a[0].length == b[0].length) {
        return isMatrixEquals(a, b, 0, 0);
    } else {
        return false;
    }
}

public static boolean isMatrixEquals(int a[][], int b[][], int i,
    int j) {
    if (i < a.length) {
        if (j < a[i].length) {
            if (a[i][j] != b[i][j]) {
                return false;
            } else {
                return isMatrixEquals(a, b, i, j + 1);
            }
        } else {
            return isMatrixEquals(a, b, i + 1, 0);
        }
    } else {
        return true;
    }
}
```