

1 자신의 이름과 학번을 출력하는 코드를 작성하시오

1

```
Name: ***  
Student number: ****
```

```
class PersonalInfo {  
    public static void main(String[] args) {  
        System.out.println("Name: ***");  
        System.out.println("Student number: ****");  
    }  
}
```

2 다음과 같이 입력받고 출력되도록 코드를 작성하세요

1

```
가로의 길이를 입력하세요 : 5  
세로의 길이를 입력하세요 : 5  
정사각형  
가로의 길이를 입력하세요 : 3  
세로의 길이를 입력하세요 : 4  
사각형
```

```
import java.util.Scanner;  
  
public class RectKind {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("가로의 길이를 입력하세요 : ");  
        int width = scanner.nextInt();  
        System.out.print("세로의 길이를 입력하세요 : ");  
        int length = scanner.nextInt();  
  
        if (width == length)  
            System.out.println("정사각형");  
        else  
            System.out.println("사각형");  
  
        scanner.close();  
    }  
}
```

3 아래와 같이 여러 개의 정수들에 대해 최대값으로 나눠 모든 값을 0~1 값으로 변환하세요(정규화)

2

```
원래 값들: 12 3 5 6 18  
정규화된 값들: 0.6666666666666666 0.16666666666666666 0.2777777777777778 0.3333333333333333 1.0
```

```
class Norm {  
    public static void main(String[] args) {  
        int[] original = { 12, 3, 5, 6, 18 };  
  
        int max = 0;  
  
        System.out.print("원래 값들: ");  
        for (int or : original) {  
            if (max < or)  
                max = or;  
            System.out.print(or + " ");  
        }  
  
        System.out.println(" ");  
  
        double normed;  
        System.out.print("정규화된 값들: ");  
        for (int i = 0; i < original.length; i++) {  
            normed = (double) original[i] / (double) max;  
            System.out.print(normed + " ");  
        }  
    }  
}
```

```

    }
}

```

4 다음 2차 부등식에 대해 해를 구하는 코드를 작성하세요.

2

$$2x^2 - 12x + 1 > 0$$

$$1 \leq x \leq 10$$

x 는 정수

```

public class GetSolutions {
    public static void main(String[] args) {
        int result = 0;
        System.out.println("Get the solutions for 2x^2-12x+1");
        for (int i = 1; i <= 10; i++) {
            result = 2 * i * i - 12 * i + 1;
            if (result > 0)
                System.out.println(i);
        }
    }
}

```

5 두 자연수를 입력받아 두 수를 포함 두 수 사이의 자연수를 모두 더한 후 출력하는 코드 작성하시오

2

```

Please input two positive integers: 1 3
Sum of the integers between 1 and 3: 6

```

```

import java.util.Scanner;

public class SumBetween {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Please input two positive integers: ");
        int first = scanner.nextInt();
        int second = scanner.nextInt();
        int smaller = 0;
        int bigger = 0;
        int sum = 0;

        if (first >= second) {
            smaller = second;
            bigger = first;
        } else {
            smaller = first;
            bigger = second;
        }

        for (int i = smaller; i <= bigger; i++) {
            sum += i;
        }

        System.out.println("Sum of the integers between " + smaller + " and " + bigger + ": " + sum);
        scanner.close();
    }
}

```

6 두 자연수를 입력받아 서로 소(공약수가 1뿐인 있는 경우)인지 확인하는 코드를 작성하시오

3

```

Please input two positive integers: 3 6
3 and 6 is not coprime

```

```

import java.util.Scanner;

public class Coprime {

```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Please input two positive digits: ");
    int first = scanner.nextInt();
    int second = scanner.nextInt();
    int smaller = 0;
    if (first >= second)
        smaller = second;
    else
        smaller = first;

    boolean coprime = true;
    for (int i = 2; i <= smaller; i++) {
        if (first % i == 0 && second % i == 0) {
            System.out.println(first + " and " + second + " is not coprime");
            coprime = false;
            break;
        }
    }
    if (coprime)
        System.out.println(first + " and " + second + " is coprime");
    scanner.close();
}
}

```

7 5개의 정수를 입력받아 배열 구성 후 배열 원소들을 출력하는 코드를 작성하시오

2

```

Please input 5 integers: 1 2 3 4 5
Array elements: 1 2 3 4 5

```

```

import java.util.Scanner;

public class MakeArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] intArray = new int[5];

        System.out.print("Please input 5 integers: ");

        for (int i = 0; i < intArray.length; i++) {
            intArray[i] = scanner.nextInt();
        }

        System.out.print("Array elements: ");
        for (int i = 0; i < intArray.length; i++) {
            System.out.print(intArray[i] + " ");
        }
        scanner.close();
    }
}

```

8 다음 두 행렬을 합하여 출력하는 코드를 작성하세요

2

1	2	3
4	5	6

7	8	9
10	11	12

```

public class ArraySum {
    public static void main(String[] args) {

        int[][] intArray1 = { { 1, 2, 3 }, { 4, 5, 6 } };
    }
}

```

```

int[][] intArray2 = { { 7, 8, 9 }, { 10, 11, 12 } };
int[][] arraySum = new int[2][3];

for (int i = 0; i < intArray1.length; i++)
    for (int j = 0; j < intArray1[i].length; j++)
        arraySum[i][j] = intArray1[i][j] + intArray2[i][j];

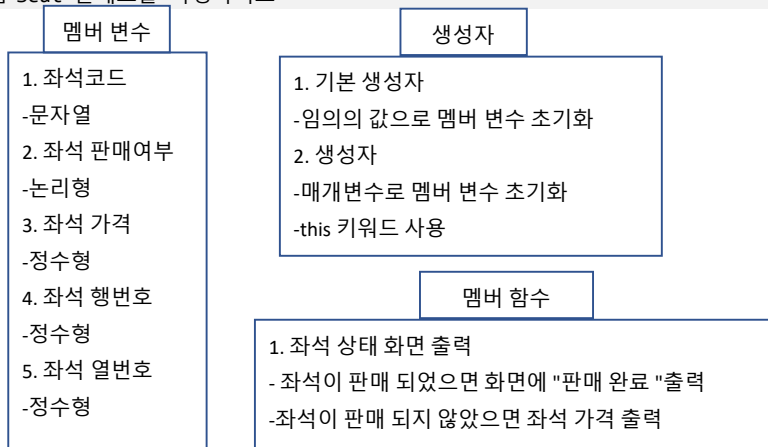
System.out.println("Array 1");
printArray(intArray1);
System.out.println("Array 2");
printArray(intArray2);
System.out.println("Sum of arrays");
printArray(arraySum);
}

public static void printArray(int[][] intArray) {
    for (int i = 0; i < intArray.length; i++) {
        for (int j = 0; j < intArray[i].length; j++)
            System.out.printf("%2d ", intArray[i][j]);
        System.out.println(" ");
    }
}
}

```

9 다음 Seat 클래스를 작성하시오

3



```

public class Seat {
    String code;
    boolean occupied;
    int price;
    int row;
    int column;

    public Seat() {
        code = "";
        occupied = false;
        price = 0;
        row = 0;
        column = 0;
    }

    public Seat(String code, boolean occupied, int price, int row, int column) {
        this.code = code;
        this.occupied = occupied;
        this.price = price;
        this.row = row;
        this.column = column;
    }
}

```

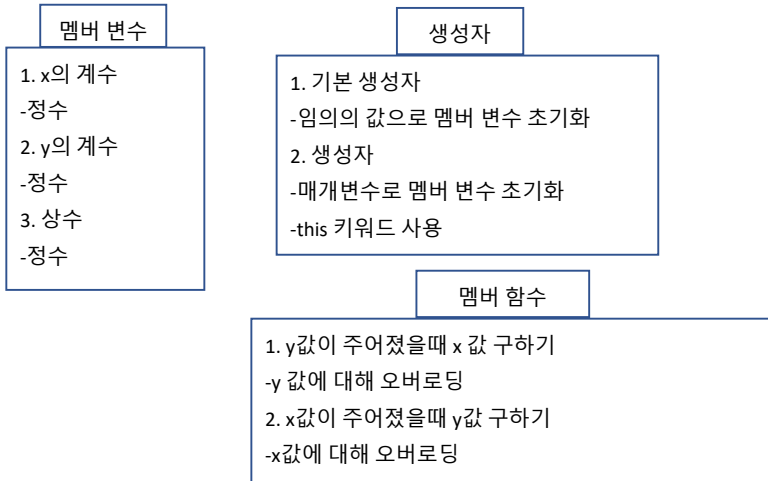
```

    }
    public void ShowStatus(){
        if(occupied){
            System.out.println("판매 완료");
        }
        else{
            System.out.println(price);
        }
    }
}
}

```

10 다음 일차방정식 클래스를 작성하시오 ($ax+by = c$)

3



```

public class LinearEquation {
    int firstCoeff;
    int secondCoeff;
    int con;

    public LinearEquation() {
        firstCoeff = 1;
        secondCoeff = 1;
        con = 1;
    }

    public LinearEquation(int firstCoeff, int secondCoeff, int con) {
        this.firstCoeff = firstCoeff;
        this.secondCoeff = secondCoeff;
        this.con = con;
    }

    public double getX(int y) {
        return (double) (con - secondCoeff * y) / (double) firstCoeff;
    }

    public double getX(double y) {
        return (double) (con - secondCoeff * y) / (double) firstCoeff;
    }

    public double getY(int x) {
        return (double) (con - firstCoeff * x) / (double) secondCoeff;
    }

    public double getY(double x) {
        return (double) (con - firstCoeff * x) / (double) secondCoeff;
    }
}

```

```

public static void main(String[] args) {
    LinearEquation le = new LinearEquation(2, 2, 3);
    System.out.println("2x+2y = 3");
    System.out.println("x = 2, y = " + le.getY(2));
    System.out.println("x = " + le.getX(2) + ", y = 2");
    System.out.println("x = 2.3, y = " + le.getY(2.3));
    System.out.println("x = " + le.getX(2.3) + ", y = 2.3");
}
}

```

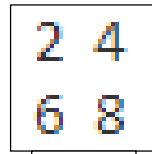
11 다음 MatrixEx 클래스의 main 함수가 오류 없이 실행 되도록 Matrix 클래스를 작성하세요

2

```

public class MatrixEx {
    public static void main(String[] args) {
        int[][] first = {{ 1, 2 }, { 3, 4 }};
        int[][] second = {{ 1, 2 }, { 3, 4 }};
        int[][] result = Matrix.add(first, second);
        Matrix.printMatrix(result);
    }
}

```



실행결과

```

class Matrix {
    public static int[][] add(int[][] first, int[][] second) {
        int[][] result = new int[first.length][first[0].length];

        for (int i = 0; i < first.length; i++)
            for (int j = 0; j < first[i].length; j++)
                result[i][j] = first[i][j] + second[i][j];

        return result;
    }

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

```

12 다음 학생 클래스를 작성하시오.

2

조건: 모든 멤버 변수는 private, 생성자 오버로딩, this 키워드 사용

학생 정보

1. 학생 이름
2. 학번
3. 상태
재학 또는 휴학 상태를 숫자로 표현
4. 현재까지 수강한 과목

기능

1. 학생 상태를 알려주기
재학, 일반휴학, 군휴학으로 알려줌
2. 과목 "x" 를 수강한 적이 있는지 알려주기
3. 이름 알려주기
4. 학번 알려주기

```

public class Student {
    private String name;
    private int num;
    private int state;
    private String[] taken;
}

```

```

public Student() {
    this(" ", 0, 0, null);
}

public Student(String name, int num, int state, String[] taken) {
    this.name = name;
    this.num = num;
    this.state = state;
    this.taken = taken;
}

public String getState() {
    switch (state) {
        case 0:
            return "재학";
        case 1:
            return "일반 휴학";
        case 2:
            return "군 휴학";
        default:
            return "미정";
    }
}

public boolean subjectTaken(String subject) {
    for (String sub : taken) {
        if (sub.equals(subject))
            return true;
    }
    return false;
}

public String getName() {
    return name;
}

public int getNum() {
    return num;
}
}

```

13 12번 문제의 학생 클래스를 이용하여 다음 학과 클래스를 작성하시오.

3

학과 정보

1. 학과 이름
 2. 학생들
- 학생 클래스의 객체들

기능

1. 과목 "x" 를 수강한 적이 있는지 학생들 이름 알려주기
2. 학과 이름 알려주기

```

public class Department {
    private Student[] students;
    private String name;

    public Department() {

    }

    public Department(String name, Student[] students) {
        this.name = name;
        this.students = students;
    }
}

```

```

public String[] getStudentsForSubject(String subject) {
    String[] subStudents = new String[0];

    int i = 0;
    for (Student student : students) {
        System.out.println(student.getName());

        if (student.subjectTaken(subject)) {
            subStudents[i] = student.getName();
            i++;
        }
    }
    return subStudents;
}

public String getName() {
    return name;
}
}

```

14 12번 문제의 학생 클래스를 변경한 후 변경된 학생클래스를 상속하여 다음 대학원생 클래스를 작성하세요.
조건: 생성자 오버로딩, this 및 super 키워드 사용, 메소드 오버라이딩

2

대학원생 정보	기능
1. 세부 전공	1. 학생 상태를 알려주기 석사 또는 박사 Student 클래스의 state 멤버 변수 값에 따라 결정 2. 세부 전공 알려주기

```

public class GraduateStudent extends Student {
    String field;

    public GraduateStudent() {
        super();
        field = "";
    }

    public GraduateStudent(String name, int num, int state, String[] taken, String field) {
        super(name, num, state, taken);
        this.field = field;
    }

    public String getState() {
        switch (super.state) {
            case 0:
                return "석사";
            case 1:
                return "박사";
            default:
                return "미정";
        }
    }

    public String getField() {
        return field;
    }
}

```