

자바 코딩테스트

# JAVA 코딩테스트 답안

류영표 강사

ryp1662@gmail.com

Copyright © "Youngpyo Ryu" All Rights Reserved.

This document was created for the exclusive use of "Youngpyo Ryu".

It must not be passed on to third parties except with the explicit prior consent of "Youngpyo Ryu".

# 시간 복잡도 계산하기

## 삼각형 모양의 별찍기

```
package test;

public class OperatorEx25 {

    public static void main(String[] args) {
        for (int i=0; i<5; i++) { //행
            for (int j=0; j<i+1; j++) { //열
                //i가 0일 때, j가 (1행 일때 * 1개)
                //i가 1일 때, j가 2(2행 일때 * 2개) ...
                // 이런식으로 진행되기 때문에 조건이 j<i+1 이렇게 된다.
                System.out.print('*');
            }
            System.out.println(); //한 줄 출력 후 줄바꿈.
        }
    }
}
```

## 피라미드 모양의 별찍기

```
package test; // 패키지 선언

public class OperatorEx25 {

    public static void main(String[] args) {
        int num = 1; // '*'을 출력할 개수를 저장하는 변수, 초기값은 1.

        for (int i = 0; i < 5; i++) { // 5개의 행을 출력하기 위한 반복
            for (int k = 4; k > i; k--) { // 각 행의 앞에 공백을 출력하여 가운데 정렬
                System.out.print(" "); // 공백 출력
            }

            for (int j = 0; j < num; j++) { // 현재 행에서 '*'을 출력하기 위한 반복
                System.out.print('*'); // '*' 출력
            }

            num = num + 2; // 다음 행에서 출력할 '*'의 개수를 홀수로 유지
            System.out.println(); // 현재 행이 끝나면 줄 바꿈
        }
    }
}
```

```
package test; // 패키지 선언

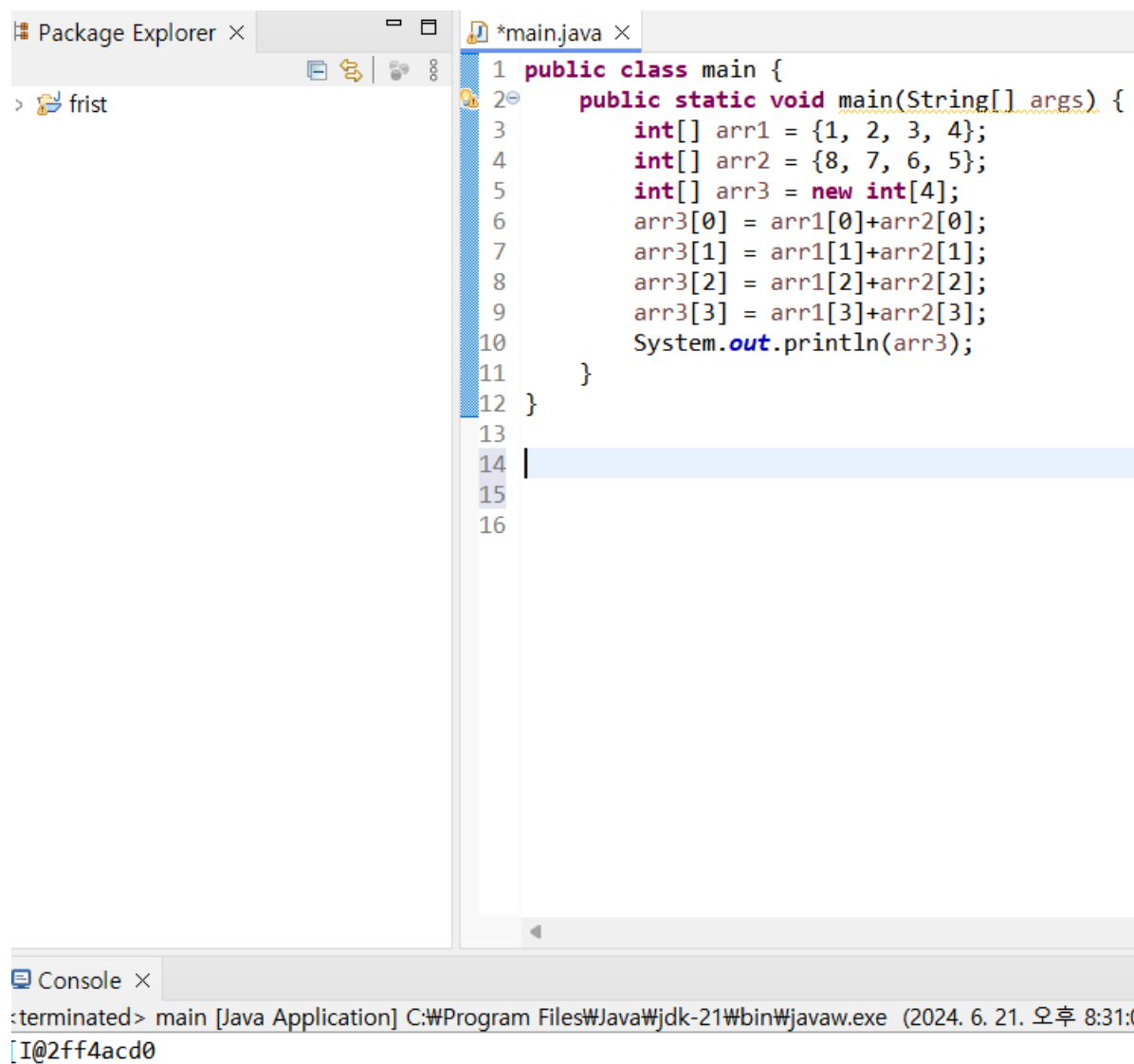
class Solution {
    public int solution(int n, int t) { // n: 초기 값, t: 반복 횟수
        int answer = 0; // 결과를 저장할 변수 초기화

        // t번 반복하여 n을 2배로 증가시킴
        for(int i = 0; i < t; i++) { // t번 만큼 반복
            n = n * 2; // n을 2배로 증가
        }

        answer = n; // 최종 결과를 answer 변수에 저장

        return answer; // 최종 결과 반환
    }

    public static void main(String[] args) {
        Solution sol = new Solution(); // Solution 클래스의 인스턴스 생성
        int result = sol.solution(5, 3); // 예시로 n=5, t=3을 전달
        System.out.println(result); // 결과 출력
    }
}
```



The screenshot shows an IDE with a Package Explorer on the left containing a package named 'frist'. The main editor displays a file named '\*main.java' with the following code:

```

1 public class main {
2     public static void main(String[] args) {
3         int[] arr1 = {1, 2, 3, 4};
4         int[] arr2 = {8, 7, 6, 5};
5         int[] arr3 = new int[4];
6         arr3[0] = arr1[0]+arr2[0];
7         arr3[1] = arr1[1]+arr2[1];
8         arr3[2] = arr1[2]+arr2[2];
9         arr3[3] = arr1[3]+arr2[3];
10        System.out.println(arr3);
11    }
12 }

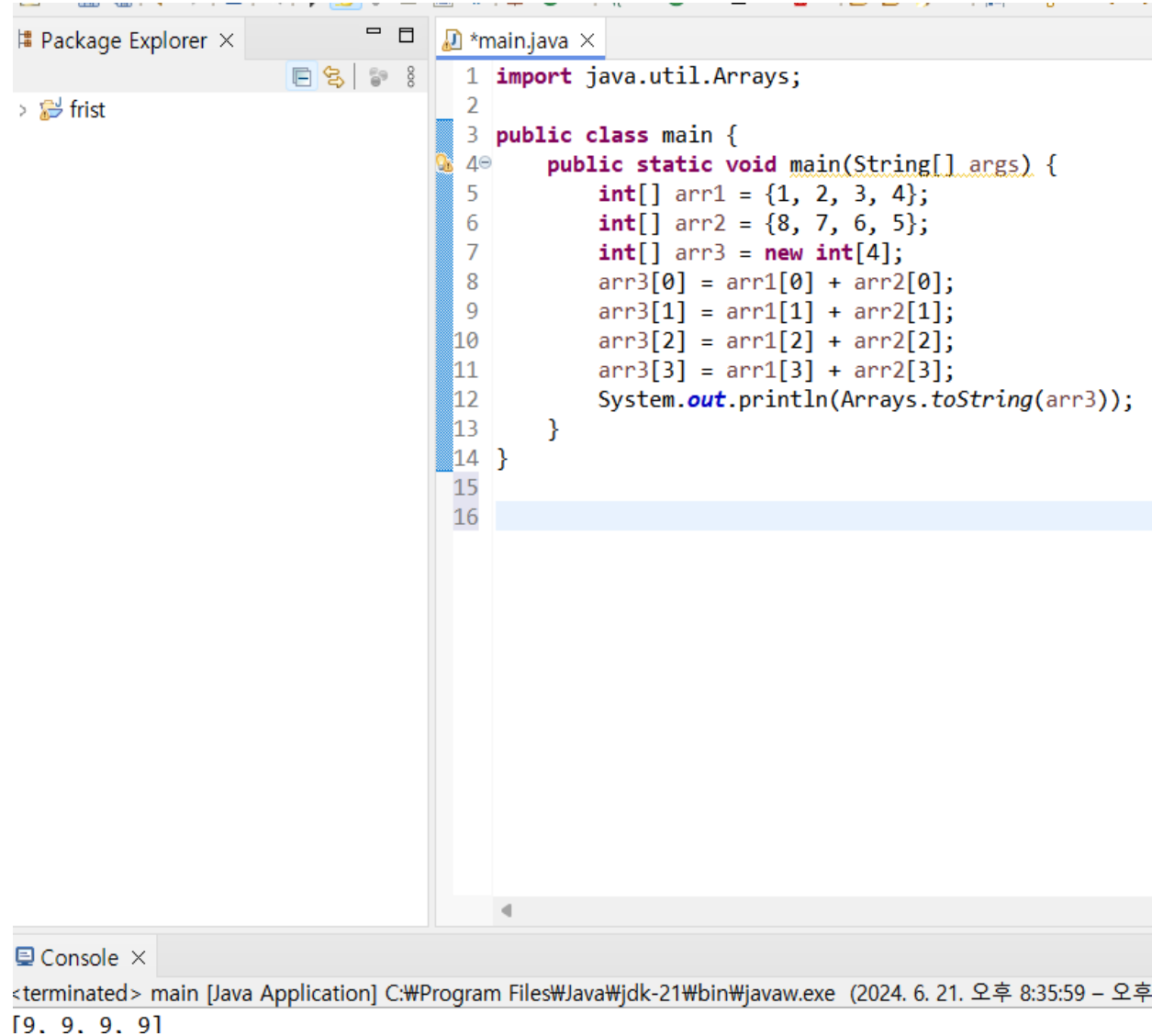
```

The Console window at the bottom shows the output of the program:

```

<terminated> main [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (2024. 6. 21. 오후 8:31:00)
[I@2ff4acd0

```



The screenshot shows an IDE with a Package Explorer on the left containing a package named 'frist'. The main editor displays a file named '\*main.java' with the following code:

```

1 import java.util.Arrays;
2
3 public class main {
4     public static void main(String[] args) {
5         int[] arr1 = {1, 2, 3, 4};
6         int[] arr2 = {8, 7, 6, 5};
7         int[] arr3 = new int[4];
8         arr3[0] = arr1[0] + arr2[0];
9         arr3[1] = arr1[1] + arr2[1];
10        arr3[2] = arr1[2] + arr2[2];
11        arr3[3] = arr1[3] + arr2[3];
12        System.out.println(Arrays.toString(arr3));
13    }
14 }

```

The Console window at the bottom shows the output of the program:

```

<terminated> main [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (2024. 6. 21. 오후 8:35:59 - 오후 8:36:00)
[9, 9, 9, 9]

```

# ArrayList - 배열 만들기1

```
import java.util.ArrayList;
import java.util.List;

public class Solution {
    public static void main(String[] args) {
        /*
         * 정수 n과 k가 매개변수로 주어집니다.
         * 1이상 n이하의 정수 중에서
         * k의 배수를 오름차순으로 저장한 배열을
         * 출력하는 문제입니다.
         * 10 , 3 > [3, 6, 9] / 15, 5 > [5, 10, 15]
         */
        int n = 15;
        int k = 5;
        List<Integer> answer = new ArrayList<Integer>();
        for (int i = 1; i <= n; i++) {
            if (i % k == 0) {
                answer.add(i);
            }
        }
        System.out.println(answer);
    }
}
```

# ArrayList - 배열 만들기2

```
import java.util.*;

class Solution {
    public int[] solution(int l, int r) {
        int[] answer = {};
        ArrayList<Integer> list = new ArrayList<>();
        int a = 0;
        for(int i = l; i <= r; i++) {
            String num = String.valueOf(i);
            String[] nums = num.split("");
            int numLength = nums.length;
            int count = 0;
            for (int j = 0; j < numLength; j++) {
                if (nums[j].equals("0") || nums[j].equals("5")) {
                    count++;
                }
            }
            if(count == numLength) {
                list.add(i);
            }
        }
        answer = list.stream().mapToInt(Integer::intValue).toArray();
        if(answer.length == 0) {
            answer = new int[] {-1};
            return answer;
        }
        return answer;
    }
}
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int N;
        N = sc.nextInt();

        int[] array = new int[N];

        for(int i=0; i<N; i++) {
            array[i] = sc.nextInt();
        }

        Arrays.sort(array);

        for(int value : array) {
            System.out.println(value);
        }
    }
}
```

```
import java.util.Scanner;

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

        Scanner in = new Scanner(System.in);

        int N = in.nextInt();
        int[] arr = new int[N];

        for(int i = 0; i < N; i++) {
            arr[i] = in.nextInt();
        }

        // Selection sort
        for(int i = 0; i < N - 1; i++) {
            for(int j = i + 1; j < N; j++) {
                if(arr[i] > arr[j]) {
                    int temp = arr[j];
                    arr[j] = arr[i];
                    arr[i] = temp;
                }
            }
        }

        for(int val : arr) {
            System.out.println(val);
        }
    }
}
```



# 백준 11650, 좌표 정렬하기

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.Comparator;
import java.util.StringTokenizer;

public class Main {

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int N = Integer.parseInt(br.readLine());
        int[][] arr = new int[N][2];
        StringBuilder sb = new StringBuilder();

        for(int i = 0; i < N; i++) {
            StringTokenizer st = new StringTokenizer(br.readLine());
            arr[i][0] = Integer.parseInt(st.nextToken());
            arr[i][1] = Integer.parseInt(st.nextToken());
        }

        //정렬
        Arrays.sort(arr, new Comparator<int[]>() {
            //sb에 정렬된 배열을 순서대로 저장해준다.
            for(int i = 0; i < N; i++) {
                sb.append(arr[i][0]).append(" ").append(arr[i][1]).append("\n");
            }
            System.out.println(sb);
        })
    }
}
```



# 문제 01) 배열 정리하기

```
import java.util.Arrays;
import java.util.Random;

class Solution {
    public static void main(String[] args) {
        // 배열을 랜덤 값으로 초기화
        int[] arr = new int[100000];
        Random random = new Random();
        for (int i = 0; i < arr.length; i++) {
            arr[i] = random.nextInt(100000); // 0부터 99999까지의 랜덤 값
        }

        // 버블 정렬 시간 측정
        long start = System.currentTimeMillis();
        int[] bubble = bubbleSort(arr);
        long end = System.currentTimeMillis();
        System.out.println("Bubble sort time: " + (end - start) / 1000.0 + "초");

        // 배열을 다시 초기화하여 동일한 값으로 정렬 수행
        for (int i = 0; i < arr.length; i++) {
            arr[i] = random.nextInt(100000);
        }

        // 내장 정렬 시간 측정
        start = System.currentTimeMillis();
        int[] sort = doSort(arr);
        end = System.currentTimeMillis();
        System.out.println("API sort time: " + (end - start) / 1000.0 + "초");

        // 두 배열이 같은지 확인
        System.out.println("Arrays are equal: " + Arrays.equals(bubble, sort));
    }
}
```

```
private static int[] bubbleSort(int[] org) {
    int[] arr = org.clone();
    int n = arr.length;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
    return arr;
}

private static int[] doSort(int[] org) {
    int[] arr = org.clone();
    Arrays.sort(arr);
    return arr;
}
}
```

# 문제) k번째 수 / 정렬(Level 1)

```
import java.util.Arrays;

class Solution {
    public int[] solution(int[] array, int[][] commands) {
        int[] answer = new int[commands.length];

        for (int i = 0; i < commands.length; i++) {
            int start = commands[i][0] - 1; // Start index
            int end = commands[i][1];       // End index (inclusive in copyOfRange)
            int k = commands[i][2] - 1;     // Kth position

            // Copy the specified range of the array and sort it
            int[] temp = Arrays.copyOfRange(array, start, end);
            Arrays.sort(temp);

            // Store the k-th element in the answer
            answer[i] = temp[k];
        }

        return answer;
    }
}
```

```
// Main method to test the solution
public static void main(String[] args) {
    Solution sol = new Solution();

    // Example input
    int[] array = {1, 5, 2, 6, 3, 7, 4};
    int[][] commands = {
        {2, 5, 3},
        {4, 4, 1},
        {1, 7, 3}
    };

    // Call the solution method and print the result
    int[] result = sol.solution(array, commands);
    System.out.println(Arrays.toString(result)); // Output: [5, 6, 3]
}
```

# 문제 ) 두개 뽑아서 정리하기

```
import java.util.ArrayList;
import java.util.Arrays;

class Solution {
    public int[] solution(int[] numbers) {
        ArrayList<Integer> list = new ArrayList<Integer>();

        for(int i=0; i<numbers.length; i++){
            for(int j=i+1; j<numbers.length; j++){
                int a = numbers[i]+numbers[j];
                if (list.indexOf(a) < 0){
                    list.add(a);
                }
            }
        }

        int[] answer = new int[list.size()];
        for (int i = 0; i < list.size(); i++) {
            answer[i] = list.get(i);
        }
        Arrays.sort(answer);

        return answer;
    }
}
```

```
class Solution {
    public int[] solution(int[] numbers) {
        Set<Integer> set = new HashSet<>();

        for(int i=0; i<numbers.length; i++) {
            for(int j=i+1; j<numbers.length; j++) {
                set.add(numbers[i] + numbers[j]);
            }
        }

        return set.stream().sorted().mapToInt(Integer::intValue).toArray();
    }
}
```

## 예제 1)

```
import java.util.*;

public class Main
{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int n = scan.nextInt();
        Integer numbers[] = new Integer[n];

        for(int i = 0; i < n; i++){
            numbers[i] = scan.nextInt();
        }

        Arrays.sort(numbers, Comparator.reverseOrder());

        for(int number : numbers){
            System.out.print(number + " ");
        }
    }
}
```

## 예제 2)

```
import java.util.*;

// 학생 클래스
class Student{
    String name;
    int score;

    public Student(String name, int score){
        this.name = name;
        this.score = score;
    }

    public String getName(){
        return this.name;
    }

    public int getScore(){
        return this.score;
    }
}

public class Main
{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int n = scan.nextInt();

        List<Student> students = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            String name = scan.next();
            int score = scan.nextInt();
            students.add(new Student(name, score));
        }

        // 정렬 후 출력
        students.stream().sorted(Comparator.comparing(Student::getScore))
            .forEach(s -> System.out.print(s.getName() + " "));

    }
}
```

## 예제 3)

```
import java.util.*;

public class Main
{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int n = scan.nextInt();
        int k = scan.nextInt();

        Integer arrA[] = new Integer[n];
        Integer arrB[] = new Integer[n];

        for (int i = 0; i < n; i++) {
            arrA[i] = scan.nextInt();
        }

        for (int i = 0; i < n; i++) {
            arrB[i] = scan.nextInt();
        }

        Arrays.sort(arrA); // 오름차순 정렬
        Arrays.sort(arrB, Comparator.reverseOrder()); // 내림차순 정렬

        for(int i = 0; i < k; i++){
            if(arrA[i] >= arrB[i]) break;

            int temp = arrA[i];
            arrA[i] = arrB[i];
            arrB[i] = temp;
        }

        long result = Arrays.stream(arrA).mapToInt(i -> i).sum();

        System.out.println("result = " + result);
    }
}
```

# 문제 05) 행렬의 곱셈

```
public class Solution {  
  
    public int[][] solution(int[][] arr1, int[][] arr2) {  
        // ❶ 행렬 arr1과 arr2의 행과 열의 수  
        int r1 = arr1.length;  
        int c1 = arr1[0].length;  
        int r2 = arr2.length;  
        int c2 = arr2[0].length;  
  
        // ❷ 결과를 저장할 2차원 배열 초기화  
        int[][] answer = new int[r1][c2];  
  
        // ❸ 첫 번째 행렬 arr1의 각 행과 두 번째 행렬 arr2의 각 열에 대해  
        for (int i = 0; i < r1; i++) {  
            for (int j = 0; j < c2; j++) {  
                // ❹ 두 행렬의 데이터를 곱해 결과 리스트에 더해줌  
                for (int k = 0; k < c1; k++) {  
                    answer[i][j] += arr1[i][k] * arr2[k][j];  
                }  
            }  
        }  
  
        return answer;  
    }  
}
```



# 문제06) 실패율

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

class Solution {
    public int[] solution(int N, int[] stages) {
        // 스테이지별 도달했으나 클리어하지 못한 사람 수
        int[] currentStages = new int[N + 1];
        // 스테이지별 도달한 사람 수
        int[] clearStages = new int[N + 1];

        for (int i = 0; i < stages.length; i++) {
            // 스테이지에 도달한 플레이어 수 증가
            for (int j = 0; j < stages[i]; j++) {
                clearStages[j] += 1;
            }
            // 도달 했으나 클리어하지 못한 플레이어 수 증가
            currentStages[stages[i] - 1] += 1;
        }

        // 스테이지 번호와 실패율을 저장할 map
        Map<Integer, Double> map = new HashMap<>();

        for (int i = 0; i < N; i++) {
            // 스테이지에 도달한 사람이 없거나 클리어한 사람이 없는 경우
            if (currentStages[i] == 0 || clearStages[i] == 0) {
                map.put(i + 1, 0.0);
            } else {
                // 실패율 계산하여 맵에 저장
                map.put(i + 1, (double) currentStages[i] / (double) clearStages[i]);
            }
        }

        List<Integer> list = new ArrayList<>(map.keySet());
        // 실패율을 기준으로 내림차순 정렬
        list.sort((o1, o2) -> Double.compare(map.get(o2), map.get(o1)));

        // 리스트를 배열로 변환하여 반환
        return list.stream().mapToInt(i -> i).toArray();
    }
}
```

## 문제 8, 올바른 괄호

```
class Solution {  
    boolean solution(String s) {  
        int openCount = 0;  
        int closeCount = 0;  
  
        for (int i = 0; i < s.length(); i++) {  
            if (s.charAt(i) == '(') {  
                openCount++;  
            } else if (s.charAt(i) == ')') {  
                closeCount++;  
            }  
            if (openCount < closeCount) {  
                return false;  
            }  
        }  
        if (openCount == closeCount) {  
            return true;  
        }  
        return false;  
    }  
}
```

스택(stack)을 사용하지 않은 경우

```
import java.util.*;  
  
class Solution {  
    boolean solution(String s) {  
        Stack<Character> stack = new Stack<>();  
        for (int i = 0; i < s.length(); i++) {  
            if (s.charAt(i) == '(') {  
                stack.push('(');  
            } else if (s.charAt(i) == ')') {  
                if (stack.isEmpty()) {  
                    return false;  
                }  
                stack.pop();  
            }  
        }  
        return stack.isEmpty();  
    }  
}
```

스택(stack)을 사용한 것

## 문제 8-1, 올바른 괄호

```
import java.util.Scanner;
import java.util.Stack;

public class Main {
    public String solution(String str) {
        String answer = "YES";
        Stack<Character> stack = new Stack<>();

        for(char x : str.toCharArray()) {
            if(x == '(') {
                stack.push(x);
            } else {
                // 닫는 괄호가 많은 상황
                if(stack.isEmpty()) {
                    return "NO";
                }
                stack.pop();
            }
        }

        // 여는 괄호가 많은 상황
        if(!stack.isEmpty()) {
            return "NO";
        }

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        String str = kb.next();
        System.out.println(T.solution(str));
    }
}
```

## 문제 8-2, 괄호문자 제거

```
import java.util.Scanner;
import java.util.Stack;

public class Main {
    public String solution(String str) {
        String answer = "";
        Stack<Character> stack = new Stack<>();

        for(char x : str.toCharArray()) {
            if(x == ')') {
                while(stack.pop() != '(');
            } else {
                stack.push(x);
            }
        }

        for(int i = 0; i < stack.size(); i++) {
            answer += stack.get(i);
        }

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        String str = kb.next();
        System.out.println(T.solution(str));
    }
}
```

# 문제 9, 후위식 연산

```
import java.util.Stack;
import java.util.Scanner;

public class Main {
    public int solution(String str) {
        int answer = 0;
        Stack<Integer> stack = new Stack<>();

        for(char x : str.toCharArray()) {
            if(Character.isDigit(x)) {
                stack.push(x - 48);
            } else {
                int rt = stack.pop();
                int lt = stack.pop();
                if(x == '+') {
                    stack.push(lt + rt);
                } else if(x == '-') {
                    stack.push(lt - rt);
                } else if(x == '*') {
                    stack.push(lt * rt);
                } else if(x == '/') {
                    stack.push(lt / rt);
                }
            }
        }

        answer = stack.get(0);

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        String str = kb.next();
        System.out.print(T.solution(str));
    }
}
```

# 문제 10, 후위식 연산(Postfix)

```
import java.util.Stack;
import java.util.Scanner;

public class Main {
    public int solution(String str) {
        int answer = 0;
        Stack<Character> stack = new Stack<>();

        for(int i = 0; i < str.length(); i++) {
            if(str.charAt(i) == '(') {
                stack.push('(');
            } else {
                stack.pop();
                if(str.charAt(i - 1) == '(') {
                    answer += stack.size();
                } else {
                    answer++;
                }
            }
        }

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        String str = kb.next();
        System.out.print(T.solution(str));
    }
}
```

# 문제 11, 짝지어 제거하기

```
import java.util.Stack;

class Solution {
    public int solution(String s) {
        // 문자를 저장할 스택
        Stack<Character> stack = new Stack<>();

        // 문자열 순회
        for (int i = 0; i < s.length(); i++) {
            if (stack.isEmpty()) { // 스택이 비어있다면 현재 문자를 스택에 추가
                stack.push(s.charAt(i));
            } else {
                char ch = s.charAt(i); // 현재 문자

                // 스택의 맨 위 문자와 현재 문자가 같다면 스택에서 제거
                if (stack.peek() == ch) {
                    stack.pop();
                } // 같지 않다면 현재 문자를 스택에 추가
                else {
                    stack.push(ch);
                }
            }
        }

        return stack.isEmpty() ? 1 : 0;
    }
}
```

```
import java.util.Stack;

class Solution
{
    public int solution(String s)
    {
        Stack<Character> stack = new Stack<>();
        //for-each문 사용
        for(char a : s.toCharArray()){
            //스택이 비어있지 않고, 맨 윗값이 넣으려는 a값과 동일하다면
            if(!stack.isEmpty() && stack.peek() == a){
                //스택 맨 위의 값 제거
                stack.pop();
            } //그 이외엔 push
            else stack.push(c);
        }
        return stack.isEmpty()? 1 : 0;
    }
}
```



## 문제 13, 크레인 인형 뽑기 게임

```
import java.util.*;
class Solution {
    public int solution(int[][] board, int[] moves) {
        int answer = 0;
        int len = board[0].length;
        Stack<Integer> st = new Stack<>();

        for(int mv: moves){
            mv -= 1;
            for(int i=0; i<len; i++){
                if(board[i][mv] != 0){ //인형집을 위치에 인형이 있는 경우 board[mv][i]
                    if(st.size() > 0 && st.peek() == board[i][mv]){ //지금 뽑은 인형과 마지막 인형이 같다면
                        st.pop();
                        answer += 2;
                    }else{
                        st.push(board[i][mv]); //인형 넣기
                    }
                    board[i][mv] = 0; //인형뽑았으니 0으로 세팅
                    break;
                }
            }
        }
        return answer;
    }
}
```

# 문제 13, 크레인 인형 뽑기 게임

```
import java.util.Stack;
import java.util.Scanner;
public class Main {
    public int solution(int[][] board, int[] moves) {
        int answer = 0;
        Stack<Integer> stack = new Stack<>();
        for(int pos : moves) {
            for(int i = 0; i < board.length; i++) {
                if(board[i][pos - 1] != 0) {
                    int tmp = board[i][pos - 1];
                    board[i][pos - 1] = 0;
                    if(!stack.isEmpty() && tmp == stack.peek()) {
                        answer += 2;
                        stack.pop();
                    } else {
                        stack.push(tmp);
                    }
                    break;
                }
            }
        }

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        int n = kb.nextInt();
        int[][] board = new int[n][n];
        for(int i = 0; i < n; i++) {
            for(int j = 0; j < n; j++) {
                board[i][j] = kb.nextInt();
            }
        }
        int m = kb.nextInt();
        int[] moves = new int[m];

        for(int i = 0; i < m; i++) {
            moves[i] = kb.nextInt();
        }
        System.out.print(T.solution(board, moves));
    }
}
```

## 문제 14, 후위식 연산(Postfix)

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

public class Main {
    public int solution(int n, int k) {
        int answer = 0;
        Queue<Integer> Q = new LinkedList<>();

        for(int i = 1; i <= n; i++) {
            Q.offer(i);
        }

        while(!Q.isEmpty()) {
            for(int i = 1; i < k; i++) {
                Q.offer(Q.poll());
            }
            Q.poll();
            if(Q.size() == 1) {
                answer = Q.poll();
            }
        }

        return answer;
    }

    public static void main(String[] args){
        Main T = new Main();
        Scanner kb = new Scanner(System.in);
        int n = kb.nextInt();
        int k = kb.nextInt();
        System.out.print(T.solution(n, k));
    }
}
```

## 문제 15, 공주 구하기

```
public class Main{
    public static int solution(int N, int K){
        int answer = 0;
        Queue<Integer> queue = new LinkedList<>();
        for(int i=1; i<=N; i++) queue.add(i);

        while(!queue.isEmpty()){
            for(int i=1; i<K; i++) queue.offer(queue.poll()); //k-1까지 뒤로 이동, k번째는 poll
            queue.poll();
            if(queue.size() == 1) answer = queue.poll(); //queue size가 1개일 경우 return
        }
        return answer;
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int N = scan.nextInt();
        int K = scan.nextInt();
        System.out.println(solution(N, K));
    }
}
```

# 문제 16, 응급실

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

class Person {
    int id;
    int priority;
    public Person(int id, int priority) {
        this.id = id;
        this.priority = priority;
    }
}

public class Main {
    public int solution(int n, int m, int[] arr) {
        int answer = 0;
        Queue<Person> Q = new LinkedList<>();

        for(int i = 0; i < n; i++) {
            Q.offer(new Person(i, arr[i]));
        }

        while(!Q.isEmpty()) {
            Person tmp = Q.poll();
            for(Person x : Q) {
                if(x.priority > tmp.priority) {
                    Q.offer(tmp);
                    tmp = null;
                    break;
                }
            }

            if(tmp != null) {
                answer++;
                if(tmp.id == m) {
                    return answer;
                }
            }
        }

        return answer;
    }
}
```

```
public static void main(String[] args){
    Main T = new Main();
    Scanner kb = new Scanner(System.in);
    int n = kb.nextInt();
    int m = kb.nextInt();
    int[] arr = new int[n];
    for(int i = 0; i < n; i++) {
        arr[i] = kb.nextInt();
    }
    System.out.print(T.solution(n, m, arr));
}
```

# Thank you.

자바 기초  
ryp1662@gmail.com

Copyright © “Youngpyo Ryu” All Rights Reserved.  
This document was created for the exclusive use of “Youngpyo Ryu”.  
It must not be passed on to third parties except with the explicit prior consent of “Youngpyo Ryu”.