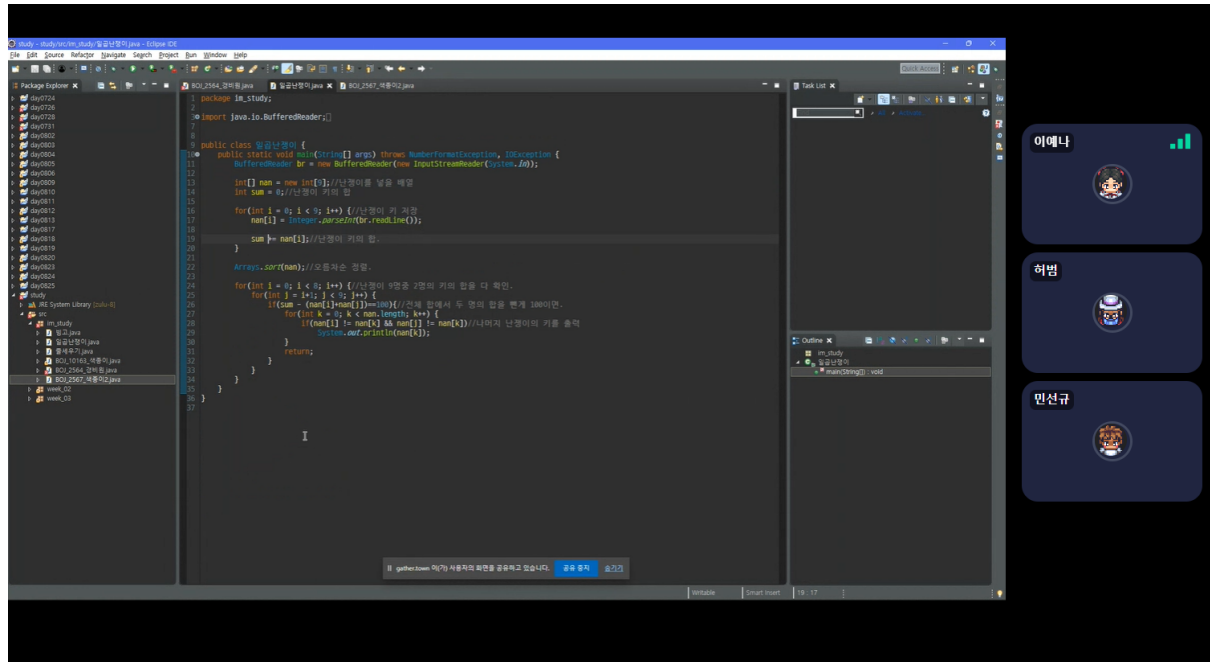


4주차 스터디 발표

- 멤버 : 민선규, 이에나, 허범, 현병욱
- 일시 : 21년 08월 28일 11시



1. 민선규 님.

2527-직사각형

```
package day0827;

import java.util.Scanner;

public class Baekjoon_2527 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 4; i++) { // 총 4번 확인하기 때문에 반복문 4번돌게 작성

            location[] arr = new location[4]; // 좌표를 4개 받기 때문에 크기가 4인 location형 배열
            생성
        }
    }
}
```

```

    for (int a = 0; a < 4; a++) {
        arr[a] = new location(sc.nextInt(), sc.nextInt()); // x, y좌표 4번 받기
    }

    if ((arr[1].x == arr[2].x && arr[1].y == arr[2].y) || (arr[0].x == arr[3].x && arr[0].y == arr[3].y) || (arr[0].x == arr[3].x && arr[1].y == arr[2].y)
        || (arr[1].x == arr[2].x && arr[0].y == arr[3].y)) { // 한 개의 직사각형을 기준으로
        꼭짓점 4개에 맞닿을 조건
        System.out.println('c'); // c를 출력하고
        continue; // 다음 반복문으로 넘어가기
    }

    if ((arr[1].x == arr[2].x && arr[1].y != arr[2].y) || (arr[0].x == arr[3].x && arr[0].y != arr[3].y) || (arr[0].x != arr[3].x && arr[1].y == arr[2].y)
        || (arr[1].x != arr[2].x && arr[0].y == arr[3].y)) { // 한 개의 직사각형을 기준으로
        변 4개에 맞닿을 조건
        System.out.println('b'); // b를 출력하고
        continue; // 다음 반복문으로 넘어가기
    }

    else if (arr[1].x < arr[2].x || arr[1].y < arr[2].y || arr[3].x < arr[0].x || arr[3].y < arr[0].y) { // 서로 전혀 겹치지 않을 조건
        System.out.println('d'); // d를 출력하고
        continue; // 다음 반복문으로 넘어가기
    }

    else {
        System.out.println('a'); // a 출력하기
    }
}

class location { // x, y 좌표를 저장할 location
    int x;
    int y;

    location(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

```

2851-슈퍼마리오

```

package day0827;

import java.util.Scanner;

```

```

public class Baekjoon_2851 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int[] arr = new int[10]; // 10개의 정수를 저장하는 배열 생성
        int sum = 0; // sum 변수를 0으로 초기화

        for (int i = 0; i < 10; i++) { // 배열에 10개의 값 넣기
            arr[i] = sc.nextInt();
        }

        for (int i = 0; i < 10; i++) {
            sum += arr[i]; // 앞에서 부터 차례차례 더하기
            if (sum > 100) { // 합이 100이 넘었을 때 실행
                if (100 - (sum - arr[i]) == sum - 100) { // 100에 똑같이 가까운 수가 2개라면 큰 값으로
                    break; // 반복문 중단
                }
                sum = (100 - (sum - arr[i]) < sum - 100) ? sum - arr[i] : sum; // 100에 제일 가까운 수 구하기
                break; // 반복문 중단
            }
        }
        System.out.println(sum);
    }
}

```

2. 이예나 님.

2564-경비원

```

package day0826;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;
import java.util.StringTokenizer;

public class Q2564 {
    static class Pos{
        int direction;
        int len;
        Pos(int direction, int len){
            this.direction = direction;
            this.len = len;
        }
    }
}

```

```

    }
}

static Pos[] store;
static Pos man;
static int minL;
static int width,height;
static int n;
public static void main(String[] args) throws IOException {

    BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
    String s = bf.readLine();
    StringTokenizer st = new StringTokenizer(s);
    width = Integer.parseInt(st.nextToken());
    height = Integer.parseInt(st.nextToken());
    n =Integer.parseInt(bf.readLine());
    store = new Pos[n+1];
    for(int i=0;i<n;i++) {
        s = bf.readLine();
        st= new StringTokenizer(s);
        int d = Integer.parseInt(st.nextToken());
        int len = Integer.parseInt(st.nextToken());
        store[i]=new Pos(d,len);

    }
    st = new StringTokenizer(bf.readLine());
    int y=Integer.parseInt(st.nextToken());
    int x = Integer.parseInt(st.nextToken());
    man = new Pos(y,x);

    makeLen();

}
static void makeLen() {
    int length =0;
    for(int i=0;i<n;i++) {
        Pos one = store[i];
        if(store[i].direction==man.direction) {//방향이 서로 같으면 길이 차이만 구하면 됨
            length += Math.abs(man.len - one.len);

        }
        else {
            int d= one.direction * man.direction; //그 외엔 곱이 모두 다름
            if(d==2) {
                int temp= height + man.len + one.len;
                if((width + height )*2 - temp<temp) {
                    temp = (width + height )*2 - temp;
                }
                length+=temp;
            }
            else if(d==3) {
                length+= man.len + one.len;
            }
            else if(d==4) {
                if(one.direction==1) {
                    length+=width-one.len + man.len;
                }
            }
        }
    }
}

```

```

        }
        else length +=width-man.len+one.len;
    }
    else if(d==6) {
        if(one.direction==2) {
            length +=height - man.len+ one.len;
        }
        else length +=height - one.len+man.len;
    }
    else if(d==8) {
        length+=width+height - one.len -man.len;
    }
    else if(d==12) {
        int temp =width + one.len +man.len;
        if((width + height )*2 - temp<temp) {
            temp = (width + height )*2 - temp;
        }
        length+=temp;
    }
}

}
System.out.println(length);

}
}

```

2605-줄세우기

```

package day0825;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.StringTokenizer;

public class Q2605 {
    static class Node{
        int value;
        int index;
        Node(int value,int index){
            this.value=value;
            this.index=index;
        }
        @Override
        public String toString() {
            return "[value=" + value + ", index=" + index + "]";
        }
    }
}

```

```

}
static List<Node>input;
public static void main(String[] args) throws IOException {
    BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
    int n = Integer.parseInt(bf.readLine());
    String s = bf.readLine();
    StringTokenizer st= new StringTokenizer(s);
    input = new ArrayList<>();
    int k=0;
    while(st.hasMoreTokens()) {
        input.add(new Node(Integer.parseInt(st.nextToken()),k));
        k++;
    }

    for(int i=0;i<n;i++) {

        int move = input.get(i).value;
        input.add(i-move, input.get(i));
        input.remove(i+1);
        //SwitchV(i,i-move);
        //System.out.println(Arrays.toString(input.toArray()));

    }
    for(Node node: input) {
        System.out.print((node.index+1)+" ");
    }
}
}

```

3. 허범 님.

2309-일곱난쟁이

```

package im_study;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

public class 일곱난쟁이 {
    public static void main(String[] args) throws NumberFormatException, IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        int[] nan = new int[9]; //난쟁이를 넣을 배열
    }
}

```

```

int sum = 0; //난쟁이 키의 합

for(int i = 0; i < 9; i++) { //난쟁이 키 저장
    nan[i] = Integer.parseInt(br.readLine());

    sum += nan[i]; //난쟁이 키의 합.
}

Arrays.sort(nan); //오름차순 정렬.

for(int i = 0; i < 8; i++) { //난쟁이 9명중 2명의 키의 합을 다 확인.
    for(int j = i+1; j < 9; j++) {
        if(sum - (nan[i]+nan[j])==100){ //전체 합에서 두 명의 합을 뺀게 100이면.
            for(int k = 0; k < nan.length; k++) {
                if(nan[i] != nan[k] && nan[j] != nan[k]) //나머지 난쟁이의 키를 출력
                    System.out.println(nan[k]);
            }
            return;
        }
    }
}
}
}
}

```

2567-색종이2

```

package im_study;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

public class BOJ_2567_색종이2 {
    public static void main(String[] args) throws NumberFormatException, IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        //배열 탐색을 위한 델타배열
        int[] dx = { -1, 0, 1, 0 };
        int[] dy = { 0, 1, 0, -1 };

        int N = Integer.parseInt(br.readLine());
        int ans = 0;

        int[][] map = new int[101][101]; //사각형을 입력받아서 기록할 배열

        for (int i = 0; i < N; i++) { //사각형 개수 만큼 반복하면서 입력받음.
            StringTokenizer st = new StringTokenizer(br.readLine());
            int x = Integer.parseInt(st.nextToken());
            int y = Integer.parseInt(st.nextToken());

            for (int j = x; j < x + 10; j++) { //사각형을 입력받아서 1로 채움.

```

```

        for (int k = y; k < y + 10; k++) {
            map[j][k] = 1;
        }
    }
}

for (int i = 1; i <= 100; i++) { //전체를 탐색하면서
    for (int j = 1; j <= 100; j++) {
        if (map[i][j] == 1) { //사각형을 만나면
            for(int dir = 0; dir < 4; dir++) { //4방향 탐색해서
                int x = i + dx[dir];
                int y = j + dy[dir];

                if(map[x][y] == 0) //경계선이면
                    ans++; //돌레 1씩 증가.
            }
        }
    }
}
System.out.println(ans);
}
}

```

4. 현병욱 님.(추후 업로드 예정)

3985-롤케이크

13300-방배정