[8월 4주차 스터디 이예나 개인 결과물]

백준 2564

```
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());
```

```
        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);

    }


}
```

백준 2605

```java
import java.io.BufferedReader;

public class Q2605 {
    static class Node{
        int value;
        int index;
        Node(int value,int index){
            this.value=value;
            this.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; //이동 값 저장
            input.add(i-move, input.get(i)); //위치에 삽입
            input.remove(i+1); // 기존 삭제
        }
            for(Node node: input) {
                System.out.print((node.index+1)+" ");
            }
```