

2주차 개인스터디 결과물(오윤기)

20001번: 고무오리 디버깅

백준 문제 풀이에 힘들어하는 수진이를 위해 민우는 문제해결에 도움이 되는 고무오리를 준비했다. 민우가 준비한 고무오리는 신비한 능력이 존재하는데, 최근에 풀던 백준 문제를 해결해주는 능력이다. 신비한

[/> https://www.acmicpc.net/problem/20001](https://www.acmicpc.net/problem/20001)

BAE<K>JOON>
ONLINE JUDGE

while 고무오리 디버깅 끝 들어올 때 까지

1. 고무오리 디버깅 시작

- 스택 생성

1-1)문제 들어올경우

push

1-2)고무오리 들어올경우

if stack.size() == 0

 스택에 push 2번

else stack.pop

if stack size == 0 sysout 고무오리아 사랑해

아니면 sysout 힙구

```
public class Main20001_RubberDuck {
    public static Stack<Character> rubberDuck;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine();
        rubberDuck = new Stack<Character>();
        while (!input.equals("고무오리 디버깅 끝")) {
            input = sc.nextLine();
            if (input.equals("문제"))
                rubberDuck.push('p');
            if (input.equals("고무오리")) {
                if (rubberDuck.size() == 0) {
                    rubberDuck.push('p');
                    rubberDuck.push('p');
                } else {
                    rubberDuck.pop();
                }
            }
        }
        if (rubberDuck.size() == 0)
            System.out.println("고무오리아 사랑해");
        else
            System.out.println("힙구");

        sc.close();
    }
}
```

17608

17608번: 막대기

문제 아래 그림처럼 높이만 다르고 (같은 높이의 막대기가 있을 수 있음) 모양이 같은 막대기를 일렬로 세운 후, 왼쪽부터 차례로 번호를 붙인다. 각 막대기의 높이는 그림에서 보인 것처럼 순서대로 6, 9, 7, 6,

<https://www.acmicpc.net/problem/17608>

BAE<K>JOON>
ONLINE JUDGE

현재 가장 높은 높이보다 높은 높이가 들어올 때 최대 높이를 갱신해야하는 점을 생각하지 못해 시행착오가 있었다.

```
public class Main17608 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int N = Integer.parseInt(br.readLine());
        int[] stickArr = new int[N];

        for (int i = 0; i < N; i++)
```

```

        stickArr[i] = Integer.parseInt(br.readLine());

        int CNT = 1;
        int lastOne = stickArr[N - 1];
        for (int x = N - 2; x >= 0; x--) {
            if (lastOne < stickArr[x]) {
                CNT++;
                lastOne = stickArr[x];
            }
        }

        System.out.println(CNT);
    }
}

```

10799

10799번: 쇠막대기

여러 개의 쇠막대기를 레이저로 절단하려고 한다. 효율적인 작업을 위해서 쇠막대기를 아래에서 위로 겹쳐 놓고, 레이저를 위에서 수직으로 발사하여 쇠막대기들을 자른다. 쇠막대기와 레이저의 배치는 다음 조

<https://www.acmicpc.net/problem/10799>

BAEKJOON
ONLINE JUDGE

스택으로도 구현가능하지만 String 함수를 잘 활용하면 스택없이도 풀이가능.

```

public class Main10799 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String stick = sc.next();
        int answer = 0;
        int cnt = 0;
        for (int i = 0; i < stick.length(); i++) {
            char c = stick.charAt(i);
            if (c == '(') {
                if (stick.charAt(i + 1) == ')') {
                    continue;
                } else {
                    cnt++;
                }
            } else {
                if (stick.charAt(i - 1) == '(') {
                    answer += cnt;
                } else {
                    cnt--;
                    answer++;
                }
            }
        }
        System.out.println(answer);
        sc.close();
    }
}

```

```
}  
}
```

2161번: 카드1

N장의 카드가 있다. 각각의 카드는 차례로 1부터 N까지의 번호가 붙어 있으며, 1번 카드가 제일 위에, N번 카드가 제일 아래인 상태로 순서대로 카드가 놓여 있다. 이제 다음과 같은 동작을 카드가 한 장 남을

<https://www.acmicpc.net/problem/2161>

BAEKJOON
ONLINE JUDGE

1. 1~N까지 차례대로 큐에 삽입
2. 큐사이즈가 0이 될때까지
 - case 1. 큐에 1개만 있을경우 poll 출력
 - case 2 . 2개이상일 경우
 - 2-1) 큐 poll sysout
 - 2-2) 큐 poll 한거 offer

```
public class Main2161 {  
    public static void main(String[] args) {  
        //1  
        Scanner sc = new Scanner(System.in);  
        int N = sc.nextInt();  
        Queue<Integer> queue = new LinkedList<Integer>();  
  
        for (int i = 1; i <= N; i++)  
            queue.offer(i);  
        //2  
        while (!queue.isEmpty()) {  
            if (queue.size() == 1) //2-1  
                System.out.print(queue.poll());  
            else { // 2-2  
                System.out.print(queue.poll() + " ");  
                queue.offer(queue.poll());  
            }  
        }  
        sc.close();  
    }  
}
```