



# Coding PPT

With Programmers coding Test

---

코드리뷰 깃주소

<https://github.com/WooSeungTae/CodeReview.git>

구성

CONTENTS

01

2018 KAKAO BLIND RECRUTMENT

02

2019 카카오 개발자 겨울 인턴십

03

2017 팀스타운 예상 대진표

04

이상한 문자 만들기

## [1차] 비밀지도

## 문제 설명

## 비밀지도

네오는 평소 프로도가 비상금을 숨겨놓는 장소를 알려줄 비밀지도를 손에 넣었다. 그런데 이 비밀지도는 숫자로 암호화되어 있어 위치를 확인하기 위해서는 암호를 해독해야 한다. 다행히 지도 암호를 해독할 방법을 적어 놓은 메모도 함께 발견했다.

1. 지도는 한 번의 길이가  $n$  인 정사각형 배열 형태로, 각 칸은 "공백"(" ") 또는 "벽"("#") 두 종류로 이루어져 있다.
2. 전체 지도는 두 장의 지도를 겹쳐서 얻을 수 있다. 각각 "지도 1"과 "지도 2"라고 하자. 지도 1 또는 지도 2 중 어느 하나라도 벽인 부분은 전체 지도에서도 벽이다. 지도 1과 지도 2에서 모두 공백인 부분은 전체 지도에서도 공백이다.
3. "지도 1"과 "지도 2"는 각각 정수 배열로 암호화되어 있다.
4. 암호화된 배열은 지도의 각 가로줄에서 벽 부분을 1, 공백 부분을 0으로 부호화했을 때 얻어지는 이진수에 해당하는 값의 배열이다.

#				#	$01001_{(2)} = 9$
#		#			$10100_{(2)} = 20$
#	#	#			$11100_{(2)} = 28$
#			#		$10010_{(2)} = 18$
	#		#	#	$01011_{(2)} = 11$



#	#	#	#		$11110_{(2)} = 30$
				#	$00001_{(2)} = 1$
#		#		#	$10101_{(2)} = 21$
#				#	$10001_{(2)} = 17$
#	#	#			$11100_{(2)} = 28$



#	#	#	#	#
#		#		#
#	#	#		#
#			#	#
#	#	#	#	#

## 입출력 예제

매개변수	값
n	5
arr1	[9, 20, 28, 18, 11]
arr2	[30, 1, 21, 17, 28]
출력	["#####", "# # #", "### #", "# ##", "#####"]

매개변수	값
n	6
arr1	[46, 33, 33, 22, 31, 50]
arr2	[27, 56, 19, 14, 14, 10]
출력	["#####", "### #", "## ##", " #####", " #####", "### # "]

# Coding 01

```
class Solution {  
    public String[] solution(int n, int[] arr1, int[] arr2) {  
        String[] answer = new String[n];  
        for(int i =0 ; i<n;i++) {  
            answer[i]=Long.parseLong(Long.toString(arr1[i]))+Long.parseLong(Long.toString(arr2[i]))+"";  
            answer[i]=String.format("%"+n+"s", answer[i]);  
            answer[i]=answer[i].replaceAll("0"," ");  
            answer[i]=answer[i].replaceAll("1","#");  
            answer[i]=answer[i].replaceAll("2","#");  
        }  
        return answer;  
    }  
}
```

## 크레인 인형뽑기 게임

## 문제 설명

게임개발자인 “조르디”는 크레인 인형뽑기 기계를 모바일 게임으로 만들려고 합니다.

“조르디”는 게임의 재미를 높이기 위해 화면 구성과 규칙을 다음과 같이 게임 로직에 반영하려고 합니다.



게임 화면은 “1 x 1” 크기의 칸들로 이루어진 “N x N” 크기의 정사각 격자이며 위쪽에는 크레인이 있고 오른쪽에는 바구니가 있습니다. (위 그림은 “5 x 5” 크기의 예시입니다). 각 격자 칸에는 다양한 인형이 들어 있으며 인형이 없는 칸은 빈칸입니다. 모든 인형은 “1 x 1” 크기의 격자 한 칸을 차지하며 **격자의 가장 아래 칸부터** 차례차례 쌓여 있습니다. 게임 사용자는 크레인을 좌우로 움직여서 멈춘 위치에서 가장 위에 있는 인형을 집어 올릴 수 있습니다. 집어 올린 인형은 바구니에 쌓이게 되는 데, 이때 바구니의 가장 아래 칸부터 인형이 순서대로 쌓이게 됩니다.

만약 같은 모양의 인형 두 개가 바구니에 연속해서 쌓이게 되면 두 인형은 터뜨리지면서 바구니에서 사라지게 됩니다. 위 상태에서 이어서 [5번] 위치에서 인형을 집어 바구니에 쌓으면 같은 모양 인형 두 개가 없어집니다.

크레인 작동 시 인형이 집어지지 않는 경우는 없으나 만약 인형이 없는 곳에서 크레인을 작동시키는 경우에는 아무런 일도 일어나지 않습니다. 또한 바구니는 모든 인형이 들어갈 수 있을 만큼 충분히 크다고 가정합니다. (그림에서는 화면표시 제약으로 5칸만으로 표현하였음)

게임 화면의 격자의 상태가 담긴 2차원 배열 board와 인형을 집기 위해 크레인을 작동시킨 위치가 담긴 배열 moves가 매개변수로 주어질 때, 크레인을 모두 작동시킨 후 터뜨려져 사라진 인형의 개수를 return 하도록 solution 함수를 완성해주세요.

## [제한사항]

- board 배열은 2차원 배열로 크기는 “5 x 5” 이상 “30 x 30” 이하입니다.
- board의 각 칸에는 0 이상 100 이하인 정수가 담겨있습니다.
  - 0은 빈 칸을 나타냅니다.
  - 1 ~ 100의 각 숫자는 각기 다른 인형의 모양을 의미하며 같은 숫자는 같은 모양의 인형을 나타냅니다.
- moves 배열의 크기는 1 이상 1,000 이하입니다.
- moves 배열 각 원소들의 값은 1 이상이며 board 배열의 가로 크기 이하인 자연수입니다.

## 입출력 예

board	moves	result
[[0,0,0,0,0],[0,0,1,0,3],[0,2,5,0,1],[4,2,4,4,2],[3,5,1,3,1]]	[1,5,3,5,1,2,1,4]	4

# Coding 02

```
import java.util.ArrayList;
class Solution {
    public int solution(int[][] board, int[] moves) {
        int answer = 0;
        ArrayList<Integer> al = new ArrayList<Integer>();
        for(int i =0 ; i < moves.length; i++) {
            for(int k =0; k < board.length; k++) {
                if(board[k][moves[i]-1] != 0) {
                    al.add(board[k][moves[i]-1]);
                    board[k][moves[i]-1] = 0;
                    break;
                }
            }
        }
        for(int k = 0 ; k < al.size(); k++) {
            if(k == al.size()-1) break;
            if(al.get(k) == al.get(k+1)) {
                al.remove(k);
                al.remove(k);
                answer += 2;
                k -= 1;
            }
        }
        return answer;
    }
}
```

## 예상 대진표

## 문제 설명

△△ 게임대회가 개최되었습니다. 이 대회는 N명이 참가하고, 토너먼트 형식으로 진행됩니다. N명의 참가자는 각각 1부터 N번을 차례대로 배정받습니다. 그리고, 1번↔2번, 3번↔4번, ... , N-1번↔N번의 참가자끼리 게임을 진행합니다. 각 게임에서 이긴 사람은 다음 라운드에 진출할 수 있습니다. 이때, 다음 라운드에 진출할 참가자의 번호는 다시 1번부터 N/2번을 차례대로 배정받습니다. 만약 1번↔2번 끼리 겨루는 게임에서 2번이 승리했다면 다음 라운드에서 1번을 부여받고, 3번↔4번에서 겨루는 게임에서 3번이 승리했다면 다음 라운드에서 2번을 부여받게 됩니다. 게임은 최종 한 명이 남을 때까지 진행됩니다.

이때, 처음 라운드에서 A번을 가진 참가자는 경쟁자로 생각하는 B번 참가자와 몇 번째 라운드에서 만나는지 궁금해했습니다. 게임 참가자 수 N, 참가자 번호 A, 경쟁자 번호 B가 함수 solution의 매개변수로 주어질 때, 처음 라운드에서 A번을 가진 참가자는 경쟁자로 생각하는 B번 참가자와 몇 번째 라운드에서 만나는지 return 하는 solution 함수를 완성해 주세요. 단, A번 참가자와 B번 참가자는 서로 붙게 되기 전까지 항상 이긴다고 가정합니다.

## 제한사항

- N :  $2^1$  이상  $2^{20}$  이하인 자연수 (2의 지수 승으로 주어지므로 부전승은 발생하지 않습니다.)
- A, B : N 이하인 자연수 (단,  $A \neq B$  입니다.)

## 입출력 예

N	A	B	answer
8	4	7	3

## 입출력 예 설명

## 입출력 예 #1

첫 번째 라운드에서 4번 참가자는 3번 참가자와 붙게 되고, 7번 참가자는 8번 참가자와 붙게 됩니다. 항상 이긴다고 가정했으므로 4번 참가자는 다음 라운드에서 2번이 되고, 7번 참가자는 4번이 됩니다. 두 번째 라운드에서 2번은 1번과 붙게 되고, 4번은 3번과 붙게 됩니다. 항상 이긴다고 가정했으므로 2번은 다음 라운드에서 1번이 되고, 4번은 2번이 됩니다. 세 번째 라운드에서 1번과 2번으로 두 참가자가 붙게 되므로 3을 return 하면 됩니다.

# Coding 03

Solution.java

```
1  class Solution
2  {
3      public int solution(int n, int a, int b){
4          int answer = 0;
5          for(int i=1; i<=Math.getExponent(n); i++) {
6              a=(a%2==0)?a/2:a/2+1;
7              b=(b%2==0)?b/2:b/2+1;
8              if(a==b) {
9                  answer=i;
10                 break;
11             }
12         }
13         return answer;
14     }
15 }
```



## 이상한 문자 만들기

### 문제 설명

문자열 `s`는 한 개 이상의 단어로 구성되어 있습니다. 각 단어는 하나 이상의 공백문자로 구분되어 있습니다. 각 단어의 짝수번째 알파벳은 대문자로, 홀수번째 알파벳은 소문자로 바꾼 문자열을 리턴하는 함수, `solution`을 완성하세요.

### 제한 사항

- 문자열 전체의 짝/홀수 인덱스가 아니라, 단어(공백을 기준)별로 짝/홀수 인덱스를 판단해야 합니다.
- 첫 번째 글자는 0번째 인덱스로 보아 짝수번째 알파벳으로 처리해야 합니다.

### 입출력 예

s	return
"try hello world"	"TrY HeLlO WoRlD"

### 입출력 예 설명

"try hello world"는 세 단어 "try", "hello", "world"로 구성되어 있습니다. 각 단어의 짝수번째 문자를 대문자로, 홀수번째 문자를 소문자로 바꾸면 "TrY", "HeLlO", "WoRlD"입니다. 따라서 "TrY HeLlO WoRlD"를 리턴합니다.

# Coding 04

```
import java.util.ArrayList;
class Solution {
    public String solution(String s) {
        String answer = "";
        String [] s1 = s.split(" ",-1);
        for(int i=0;i<s1.length;i++) {
            char [] arry = s1[i].toLowerCase().toCharArray();
            String a="";
            for(int k=0;k<arry.length;k++){
                if(k%2==0) {
                    a+=Character.toString((char) (arry[k]-32));
                }else {
                    a+=Character.toString(arry[k]);
                }
            }
            if(i==s.length()-1) {
                answer+=a;
            }
            answer+=a+" ";
        }

        return answer.substring(0,answer.length()-1);
    }
}
```



Coding PPT  
THANK YOU FOR WATCHING

- 우 승 태 -