

# 25일 (금)

## 1. 프로그래머스 문제 풀이

### 1) 비밀지도

#### [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 적어놓은 메모도 함께 발견했다.

1. 지도는 한 변의 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.

저는 이 단 1 에 대응하는 값이 없습니다.

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

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



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

네오가 프로도의 비상금을 손에 넣을 수 있도록, 비밀지도의 암호를 해독하는 작업을 도와줄 프로그램을 작성하라.

## 입력 형식

입력으로 지도의 한 변 크기 `n` 과 2개의 정수 배열 `arr1` , `arr2` 가 들어온다.

- $1 \leq n \leq 16$
- `arr1` , `arr2` 는 길이 `n` 인 정수 배열로 주어진다.
- 정수 배열의 각 원소 `x` 를 이진수로 변환했을 때의 길이는 `n` 이하이다. 즉,  $0 \leq x \leq 2^n - 1$  을 만족한다.

## 출력 형식

원래의 비밀지도를 해독하여 `'#'` , 공백 으로 구성된 문자열 배열로 출력하라.

## 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	["#####", "### #", "## ##", " ##### ", " #####", "### # "]

[해설 보러가기](#)

```
def solution(n, arr1, arr2):  
  
    # arr1, arr2 이진법 변환 리스트  
    arr1_binary = []  
    arr2_binary = []
```

```

# arr1 이진법 변환      # 이진법 변환 메소드 bin()
for num in arr1:
    binary = bin(num)
    binary = binary[2:]

    if len(binary) != n:
        binary = binary.zfill(n)    # 문자열 앞에 0 채우기 메서드

    arr1_binary.append(binary)
# arr2 이진법 변환
for num in arr2:
    binary = bin(num)
    binary = binary[2:]

    if len(binary) != n:
        binary = binary.zfill(n)    # 문자열 앞에 0 채우기 메서드 zfill(총 숫자)

    arr2_binary.append(binary)

# final 지도 생성
answer_list = []

for a, b in zip(arr1_binary, arr2_binary):
    ls = []
    for i, j in zip(a, b):
        if int(i) + int(j) != 0:
            ls.append('#')
        else:
            ls.append(' ')

    answer_list.append(''.join(ls))    # TIL append 안에 join 메서드 가능

return answer_list

```

#다른 사람 풀이

```

def solution(n, arr1, arr2):
    answer = []
    for i, j in zip(arr1, arr2):
        a12 = str(bin(i|j)[2:])    # i|j 의 뜻은 뭘까
        a12=a12.rjust(n, '0')      # rjust 대신 zfill( ) 도 가능
        a12=a12.replace('1', '#')
        a12=a12.replace('0', ' ')
        answer.append(a12)
    return answer

```

— 배운점 —

## 1. bin( ) 함수

bin( ) 함수로 10진법에서 2진법(0b) 으로 쉽게 변경 가능하다 . ( 16진수는 hex( value ) → 0x , 8진수는 oct( value ) → 0o 을 이용하여 변경 가능하다.)

## 2. 문자열 앞에 0 채우기 - text.zfill( ) , rjust( ) , format( ) 함수

text.zfill( ) 은 문자열에서 맨 앞에 0을 채워주는 함수이다.

```
text = '2'

a = text.zfill(2)
b = text.zfill(10)
c = text.zfill(1)
print(a)
print(b)
print(c)

# 02
# 0000000002
# 2
```

rjust( ) 는 0 뿐만 아니라 다른 문자열을 더해줄 수 있다.

```
text = '2'

a = text.rjust(2, 'a')
b = text.rjust(2, '0')
c = text.rjust(10, 'd')
print(a)
print(b)
print(c)

# a2
# 02
# dddddddd2
```

`format( )`은 문자열이 아니라 정수 타입일 때도 원하는 만큼 갯수를 채울 수 있다.

```
target = 2

a = format(target, '03')
b = '{0:06d}'.format(target)

print(a)
print(b)

# 002
# 000002
```