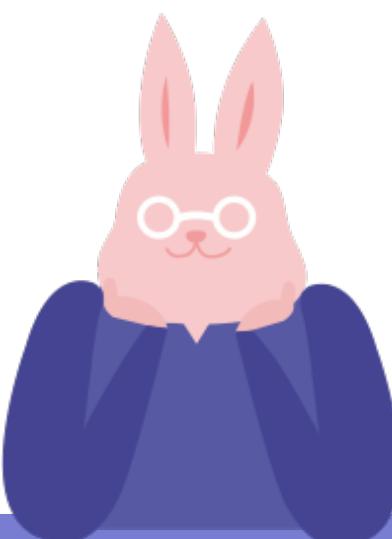


# Introduction to Problem Solving

2016. 11. 15

신현규



/\* elice \*/

[Files](#) [Settings](#)
[Left Pane](#) [Right Pane](#) [Run on Window](#)
[Share Code](#)
[History](#) [Reset](#)
[Run](#) [Submit](#) [Stop](#)

## 과제

이번 과제에서는 간단한 HEX color string to RGB 번역기를 작성합니다 ([예제](#)). RGB는 컴퓨터에서 색을 표현하는 방법 중에 하나로 (R, G, B) 빨강, 초록, 파랑의 세 색으로 이루어지며 R, G, B 모두 0에서 255의 값을 가집니다. HEX color string은 웹에서 색을 표시하기 위해 사용되는 방식으로 #RRGGBB 의 포맷을 가지며 RR, GG, BB는 16진수로 표기됩니다. HEX 코드를 입력받아 조금 더 해석하기 쉬운 RGB로 번역해서 출력하는 알고리즘을 작성해 주십시오.

### 입력

한 줄에 HEX color string을 입력받습니다. HEX 값은 '#과 3개의 대문자 16진수 숫자로 이루어집니다. 각각 세 개의 16진수는 모두 두 자리를 가집니다 (예: 186 → BA, 7 → 07, 211 → D3).

### 출력

입력받은 HEX 값에 대응하는 R, G, B 값을 한 줄에 하나씩 출력하면 됩니다.

hex2rgb.py ×

```

1 def main():
2     hex_color = input()
3
4     r, g, b = hex2rgb(hex_color)
5
6     print(r)
7     print(g)
8     print(b)
9
10    ### IMPLEMENT THIS FUNCTION ###
11    ### ONLY THIS FUNCTION WILL BE GRADED ###
12
13    def singleHexToInt(myHex):
14        return int(myHex, 16)
15
16    def twoHexToInt(myHex):
17        return singleHexToInt(myHex[0]) * 16 + singleHexToInt(myHex[1])
18
19    def hex2rgb(hex_color):
20        r = 0
21        g = 0
22        b = 0
23
24        R = hex_color[1:3]
25        r = twoHexToInt(R)

```



/\* elice \*/



한국정보올림피아드  
Korea Olympiad in Informatics



SAMSUNG  
COLLEGIATE  
PROGRAMMING  
CUP

CODEFORCES<sup>β</sup>  
Sponsored by Telegram

[TOPCODER]

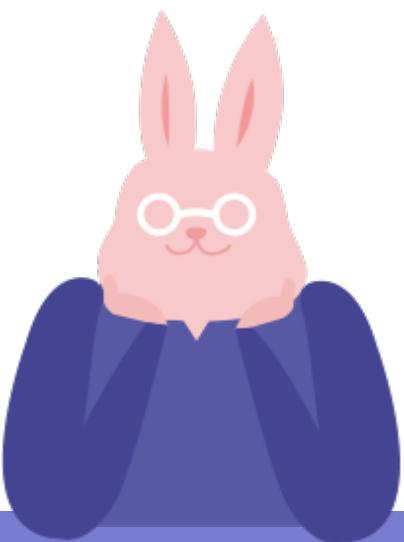


Google  
code jam

cout << "hello, world!" << endl;

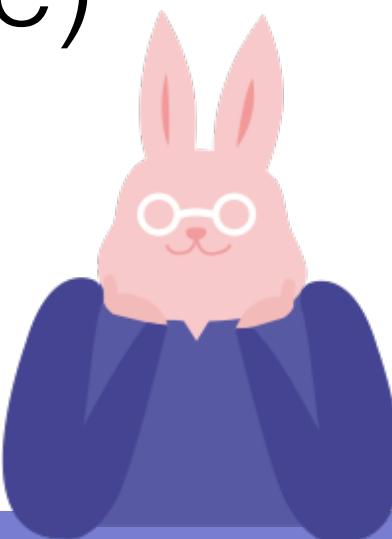


# 컴퓨터공학적으로 사고하기



# 커리큘럼

1. 재귀호출, 추상화
2. 시간복잡도, 알고리즘 정확성 증명, 정렬, 자료구조
3. 분할정복법, 탐욕적 기법
4. 동적계획법 1
5. 동적계획법 2
6. 그래프 이론 1
7. 그래프 이론 2
8. 세계 여러 기업의 입사 인터뷰 문제 도전 (+ NP-Complete)



# 강사 소개

## 신현규

KAIST KIXLAB 인턴

### 학력

- 2015 포항공과대학교 최우등 졸업

### 수상경력

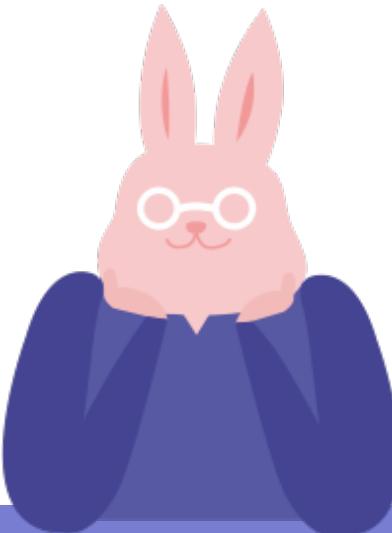
- 2012 ACM-ICPC 5<sup>th</sup> place
- 2011 POSTECH Programming Contest 1<sup>st</sup> place
- 2010 한국 정보올림피아드 경시대회 금상

### 실무경험

- Oracle Labs Intern @ California
- SAP Labs Intern @ Seoul

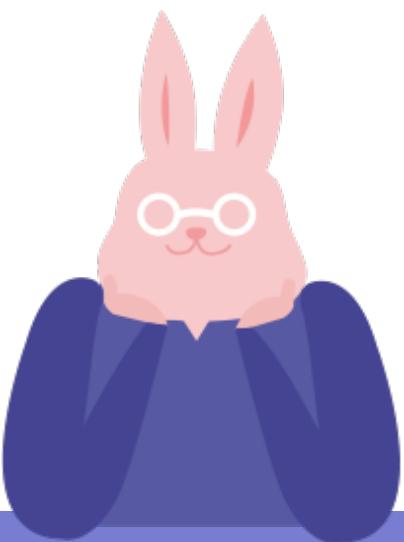
15.12. ~ 16.03.

15.07. ~ 15.08.



# 연락처

- E-mail : hyungyu.sh@kaist.ac.kr
- 카카오톡 : yougatup



# 프로그래밍

/\* elice \*/



## 과제

이번 과제에서는 간단한 HEX color string to RGB 번역기를 작성합니다 ([예제](#)). RGB는 컴퓨터에서 색을 표현하는 방법 중에 하나로 (R, G, B) 빨강, 초록, 파랑의 세 색으로 이루어지며 R, G, B 모두 0에서 255의 값을 가집니다. HEX color string은 웹에서 색을 표시하기 위해 사용되는 방식으로 #RRGGBB 의 포맷을 가지며 RR, GG, BB는 16진수로 표기됩니다. HEX 코드를 입력받아 조금 더 해석하기 쉬운 RGB로 번역해서 출력하는 알고리즘을 작성해 주십시오.

## 입력

한 줄에 HEX color string을 입력받습니다. HEX 값은 '#'과 3개의 대문자 16진수 숫자로 이루어집니다. 각각 세 개의 16진수는 모두 두 자리를 가집니다 (예: 186 -> BA, 7 -> 07, 211 -> D3).

## 출력

입력받은 HEX 값에 대응하는 R, G, B 값을 한 줄에 하나씩 출력하면 됩니다.

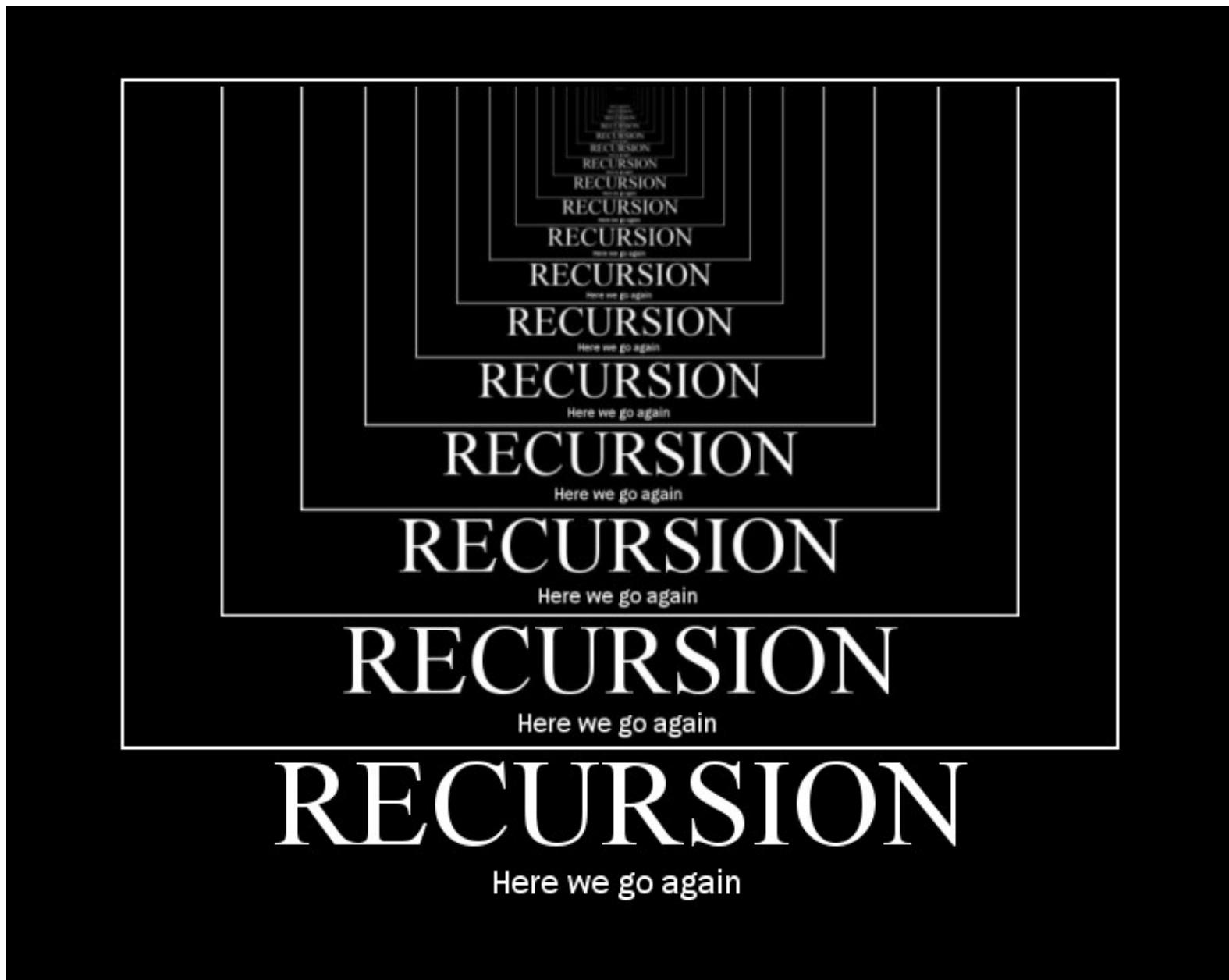
```

hex2rgb.py ×

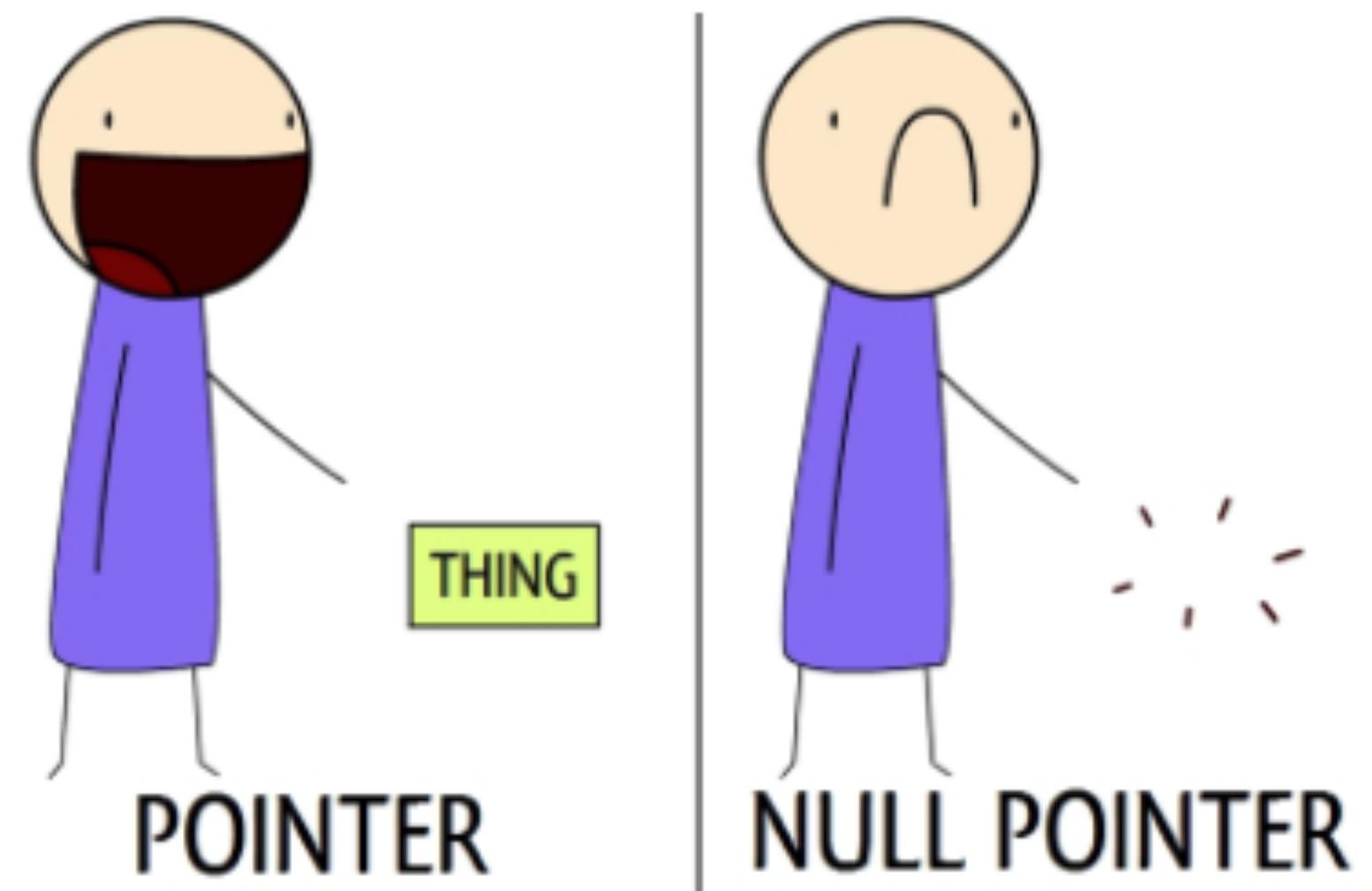
1 def main():
2     hex_color = input()
3
4     r, g, b = hex2rgb(hex_color)
5
6     print(r)
7     print(g)
8     print(b)
9
10    ### IMPLEMENT THIS FUNCTION ###
11    ### ONLY THIS FUNCTION WILL BE GRADED ###
12
13    def singleHexToInt(myHex):
14        return int(myHex, 16)
15
16    def twoHexToInt(myHex):
17        return singleHexToInt(myHex[0]) * 16 + singleHexToInt(myHex[1])
18
19    def hex2rgb(hex_color):
20        r = 0
21        g = 0
22        b = 0
23
24        R = hex_color[1:3]
25        r = twoHexToInt(R)

```

# 2가지 관문 (C/C++ programming)



# 1. 재귀호출



## 2. 포인트



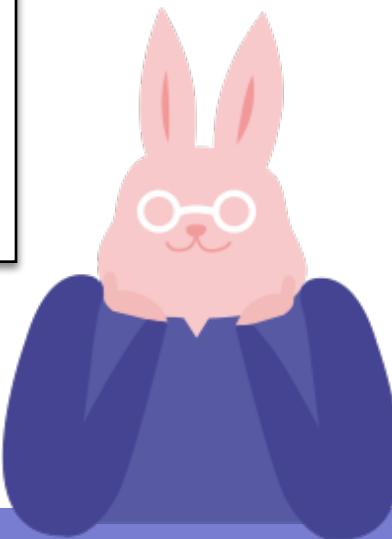
# 함수 : 프로그래밍의 최소 단위

문제 : 아래와 같이 '\*'으로 크기 n의 삼각형을 출력하시오

입력의 예

출력의 예

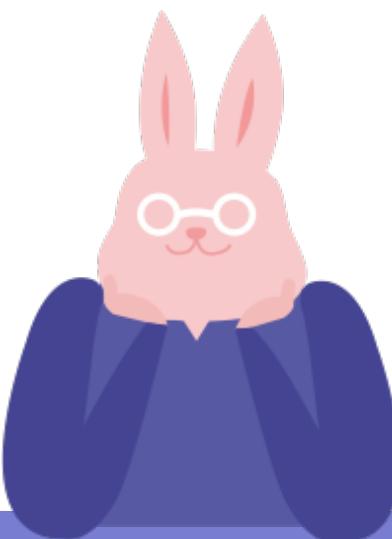
```
*  
**  
***  
****  
*****
```



# 함수 : 프로그래밍의 최소 단위

```
*  
**  
***  
****  
  
def printStarTriangle(x) :  
    for i in range(x) :  
        printString = ""  
  
        for j in range(i+1) :  
            printString = printString + "*"  
  
        print(printString)  
  
printStarTriangle(5)
```

전형적인 예제 : 2중 for문



# 함수 : 프로그래밍의 최소 단위

```
def printStars(n) :  
    printString = ""  
  
    for i in range(n) :
```

\* 머릿속 풀이를 **최대한 그대로** 코드로 옮긴다.  
\*,  
\*,  
\*,  
\*,  
\* 추상적 사고 먼저, 그 후 세부사항 고려 !  
\*,

\*\*\*\*\*

```
printStars(i+1)
```

```
printStarTriangle2(5)
```

함수를 최소단위로 한 코드



# [활동문제 1] 숫자 피라미드

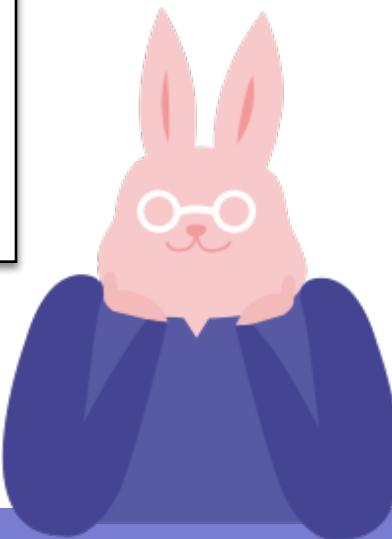
문제 : 아래와 같이 크기 n의 숫자 피라미드를 출력하시오

입력의 예

5

출력의 예

1  
234  
98765  
1234567  
765432198



# [활동문제 1] 숫자 피라미드

- 솔루션 : 어떤 함수가 필요한가 ?

printSpace (n) : null

n개의 공백을 출력

printRightDirection (start, cnt) : end

start부터 cnt개의 숫자를 오른쪽 방향으로 출력

다음으로 출력해야하는 숫자를 반환

printLeftDirection (start, cnt) : end

start부터 cnt개의 숫자를 왼쪽 방향으로 출력

다음으로 출력해야하는 숫자를 반환



# [활동문제 1] 숫자 피라미드

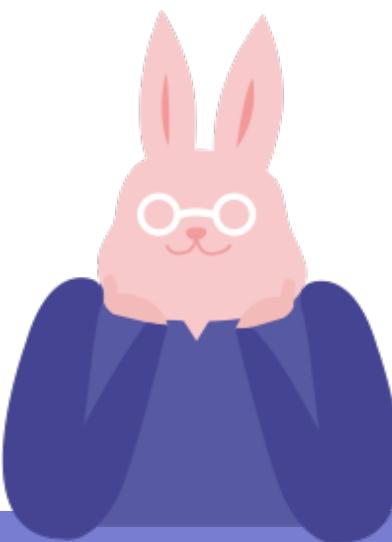
- 추상적 코드

```
number = 1
printCnt = 1

for i in range(n) :
    printSpace(n-i+1)

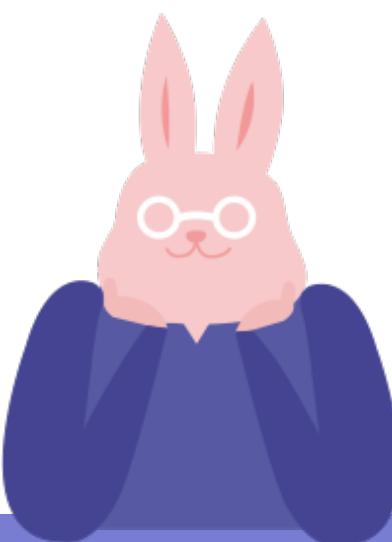
    if i % 2 == 0 :
        number = printLeftDirection(number, printCnt)
    else :
        number = printRightDirection(number, printCnt)

    printCnt = printCnt + 2
```



# 함수 : 프로그래밍의 최소 단위

- 결론
  - 함수를 최소 단위로 하여 **추상적 논리**에 집중하자
  - 각 함수가 수행하는 일을 명확히 하자
  - 올바른 논리가 세워진 다음 각 함수에 집중하자



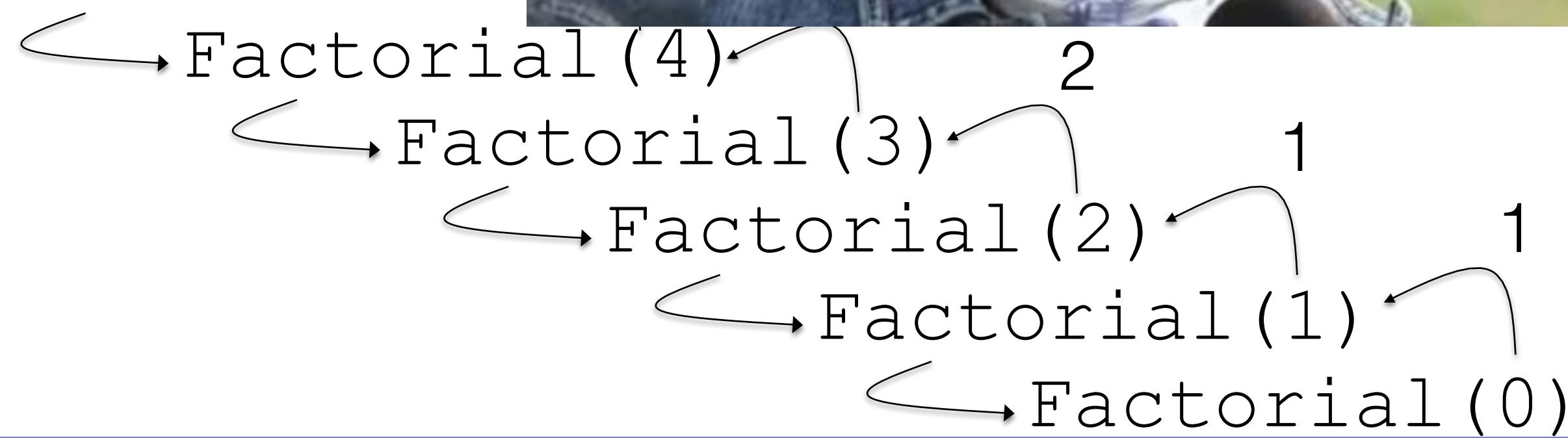
# 재귀함수

- 개념 : 자기 자신을 호출하는 함수

```
def Factorial(n) :
    if n == 0 :
        return 1
    else :
        return n * Factoria
```

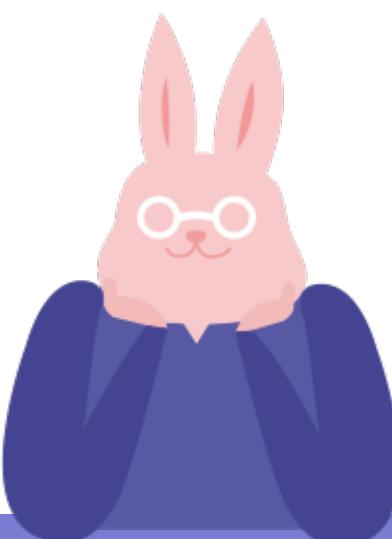
Answer = Factorial(5)

120



# 재귀함수 : 왜 필요한가 ?

- **수학적 귀납법**과 같은 원리로 사용됨
  - 명제  $P(k)$ 가 참이라고 가정할 때,  $P(k+1)$ 이 참임을 보임 혹은  
명제  $P(k')$ 가 참이라고 가정할 때,  $P(k)$ 이 참임을 보임 ( $k' < k$ )
- 함수  $f$ 의 값을 구하기 위해 함수  $f$ 의 논리를 그대로 사용



# 재귀호출 문제풀이 과정

1. 값을 구할 함수  $f$ 를 정의한다
  - Factorial( $n$ ) :  $n!$  을 반환하는 함수
  - Fibonacci( $n$ ) :  $F_n$  을 반환하는 함수
2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.
  - $\text{Factorial}(n) = n * \text{Factorial}(n-1)$
  - $\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$
3. 정답이 어느 함수의 값인지를 찾는다
  - $t! = \text{Factorial}(t)$
  - $F_t = \text{Fibonacci}(t)$

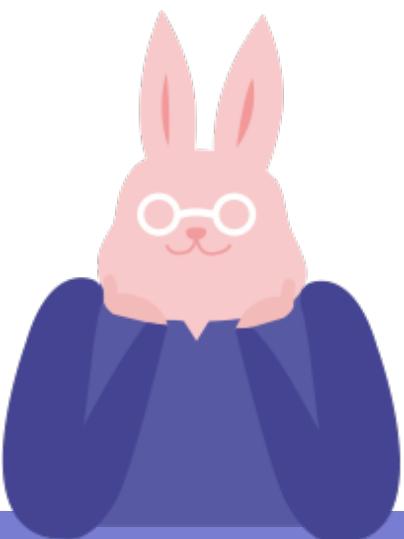


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

7을 찾자!

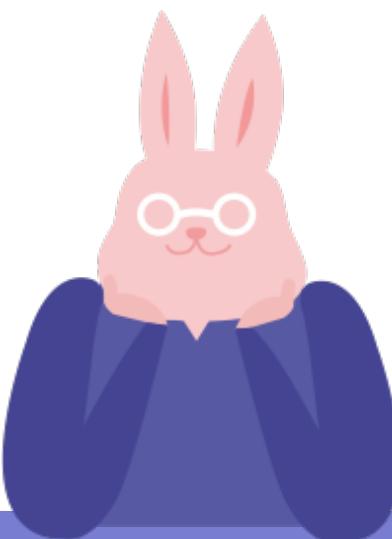


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

7을 찾자!

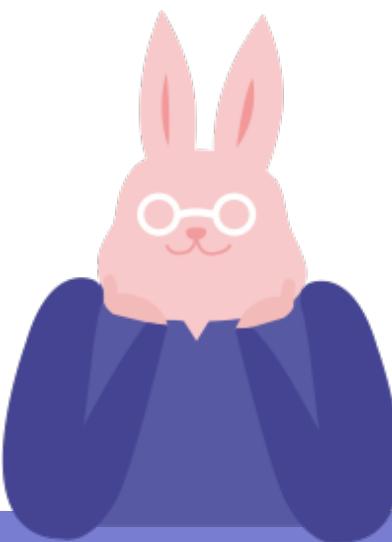


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

7을 찾자!



# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

8을 찾자!

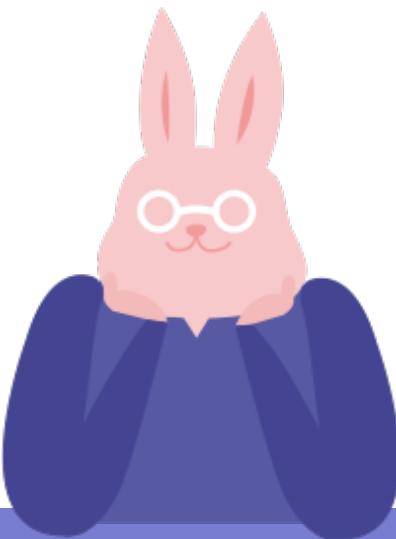


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

8을 찾자!

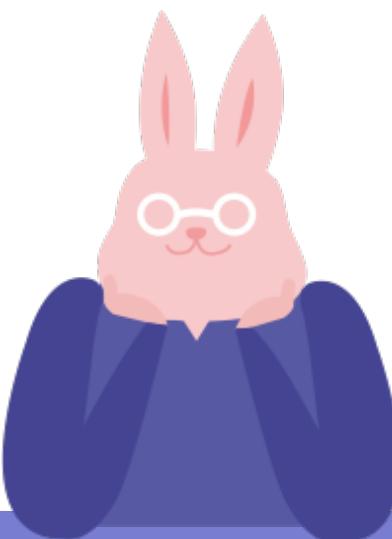


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

8을 찾자!

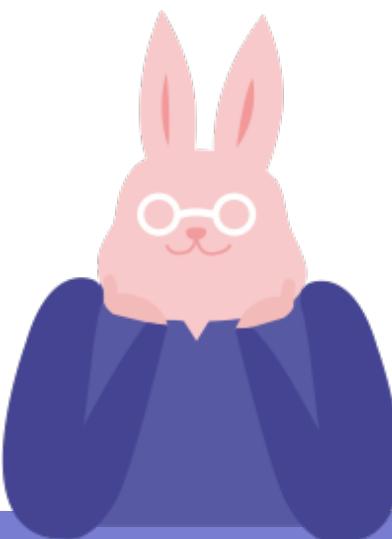


# Binary Search

- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17

8을 찾자!



# Binary Search

1. 값을 구할 함수 f를 정의한다

- binSearch(array, start, end, value)

: array의 [start, end]에 value가 있으면 True를 반환하는 함수



# Binary Search

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- binSearch(array, start, end, value) =

```
if start > end : return False
elif start == end :
    if array[start] == value :
        return True
    else :
        return False
else :
    mid = (start + end) / 2
    if array[mid] == value :
        return True
    elif array[mid] > value :
        return binSearch(array, start, mid-1, value)
    else :
        return binSearch(array, mid+1, end, value)
```



# Binary Search

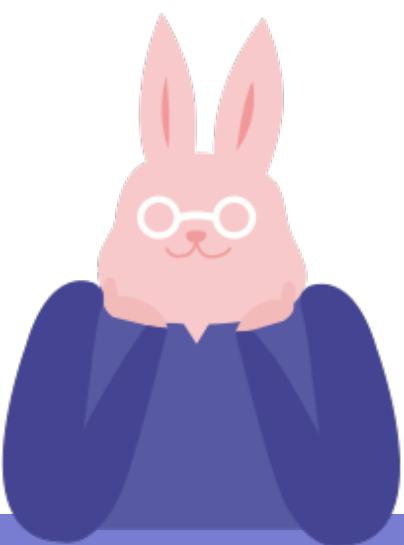
- 정렬된 숫자들이 주어졌을 때, 특정 숫자를 찾는 알고리즘

1      3      7      10      12      14      15      17  
True      Answer = binSearch(arr, 0, 7, 7)  
               $\leftarrow$  binSearch(arr, 0, 2, 7)      True  
               $\leftarrow$  binSearch(arr, 2, 2, 7)



# 최대공약수 구하기

- 두 숫자의 최대공약수를 구하라
1. 값을 구할 함수  $f$ 를 정의한다
    - $\text{GCD}(x, y) = x$ 와  $y$ 의 최대공약수를 반환하는 함수
  2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.
    - $\text{GCD}(x, y) =$



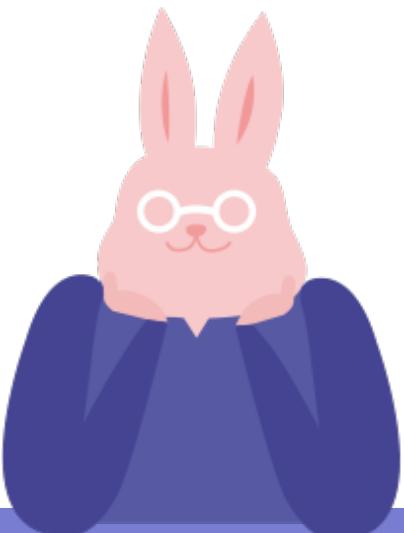
# 유클리드 호제 법

- $x > y$  라고 가정할 때,  $\text{GCD}(x, y)$ 의 귀납적 정의

$$\text{GCD}(x, y) = \begin{cases} y & \text{if } x \% y == 0 \\ \text{GCD}(y, x \% y) & \text{otherwise} \end{cases}$$

질문. 이 함수는 끝이 나는가?

**Yes. 숫자가 계속 작아진다.**



# [활동문제 2] 두 수의 최대공약수 구하기

문제 : 두 수의 최대공약수를 출력

입력의 예

6 4

출력의 예

2

입력의 예

7 3

출력의 예

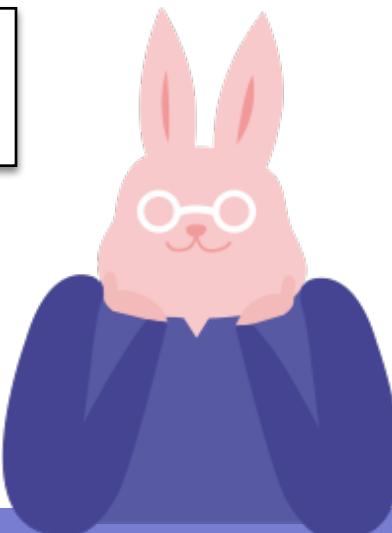
1

입력의 예

17 51

출력의 예

3



# [활동문제 3] 괄호 짹 맞추기

문제 : 주어진 괄호 문자열의 짹이 맞으면 Yes, 아니면 No 출력

입력의 예

((())

출력의 예

Yes

입력의 예

((()))((())()

출력의 예

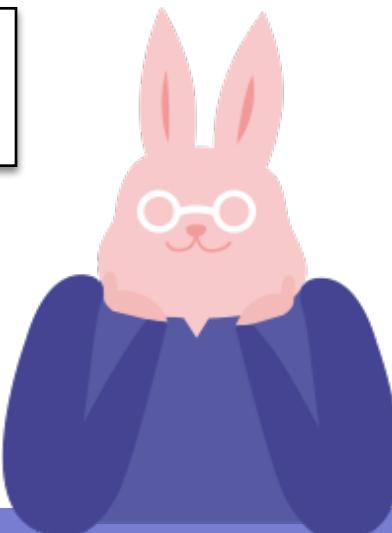
Yes

입력의 예

(())

출력의 예

No



# [활동문제 3] 괄호 짹 맞추기

1. 값을 구할 함수 f를 정의한다

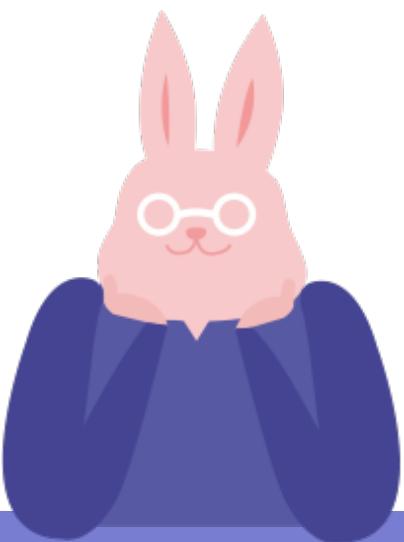
- Check(p)

: 괄호 문자열 p의 짹이 맞으면 True, 아니면 False를 반환하는 함수

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check(p) =

( ( ( ( ) ( ) ) ) ) ( ( ) ( ) )



# [활동문제 3] 괄호 짹 맞추기

1. 값을 구할 함수  $f$ 를 정의한다

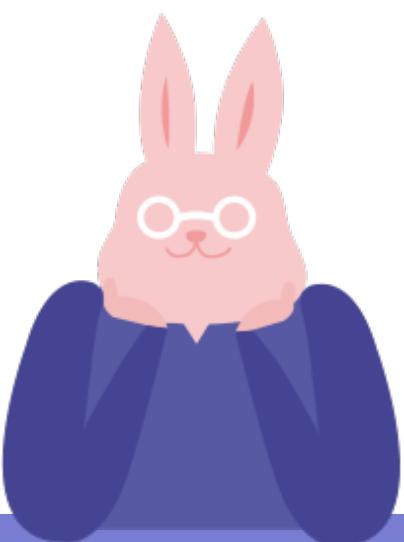
- Check( $p$ ,  $idx$ )

: 괄호  $p[idx]$ 의 짹의 위치를 반환하는 함수. 단  $p[idx]$ 는 '('라 가정.

2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check( $p$ ,  $idx$ ) =

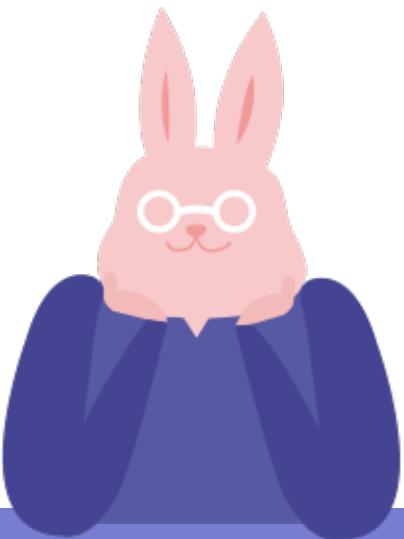
( ( ( ( ) ( ) ) ) ) ( ( ) ( ) )



# [활동문제 3] 괄호 짹 맞추기

2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.

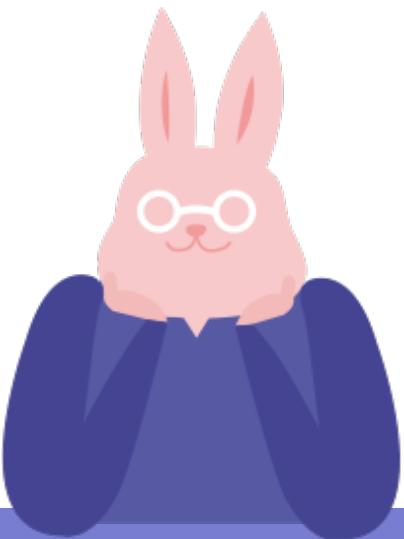
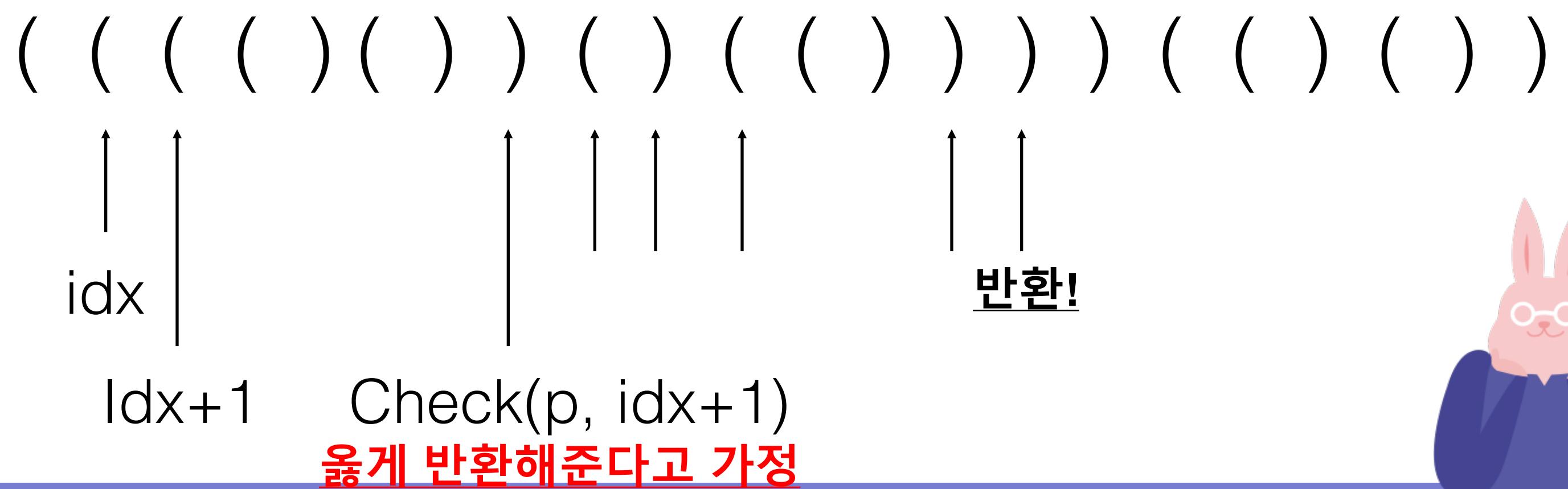
- $\text{Check}(p, \text{idx}) =$



# [활동문제 3] 괄호 짹 맞추기

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check(p, idx) =

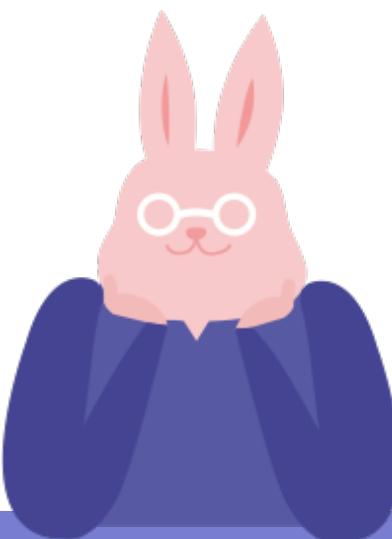


# [활동문제 3] 괄호 짹 맞추기

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check(p, idx) =

Case 1.  $p[idx+1] = ')'$       ( ( ) ...  
                return idx+1      ↑  
                                    idx



# [활동문제 3] 괄호 짹 맞추기

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check(p, idx) =

Case 2.  $p[idx+1] = '('$

$( ( ( \dots ) )$

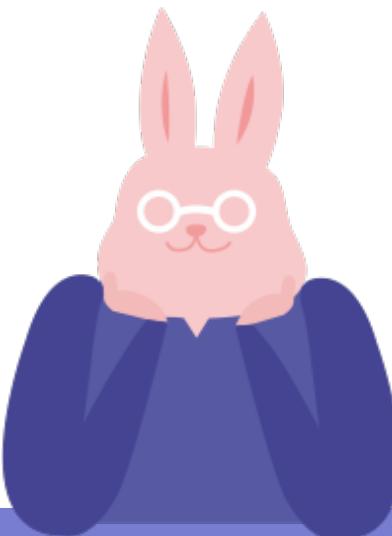
idx

j

$j = \text{Check}(p, idx+1)$

$\text{if } p[j+1] == ')'$

$\text{return } j+1$



# [활동문제 3] 괄호 짹 맞추기

2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.

- $\text{Check}(p, \text{idx}) =$

Case 2.  $p[\text{idx}+1] = '('$       ( ( ( ... ) (

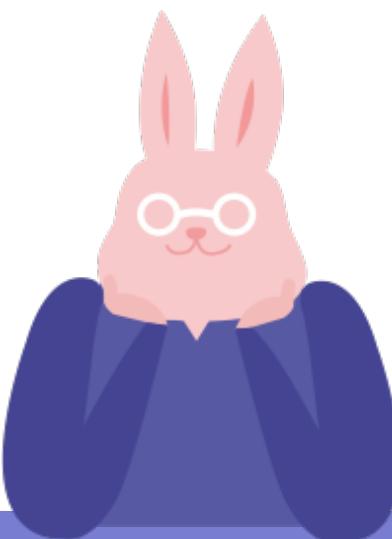
$\uparrow$      $\uparrow$

idx    j

j = Check(p, idx+1)

if p[j+1] == ')'

return j+1



# [활동문제 3] 괄호 짹 맞추기

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

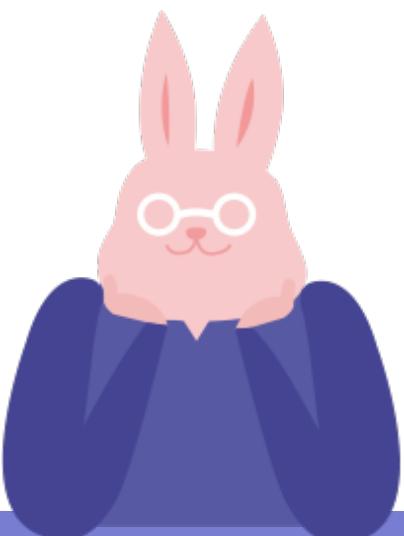
- Check(p, idx) =

Case 2.  $p[idx+1] = '('$        $((((...))$   
 $j = idx + 1$                            $\uparrow$                    $\uparrow$   
 $idx$      $j$

```

while (p[j] != ')')
    j = Check(p, j)
    j++
return j

```



# [활동문제 3] 괄호 짹 맞추기

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- Check(p, idx) =

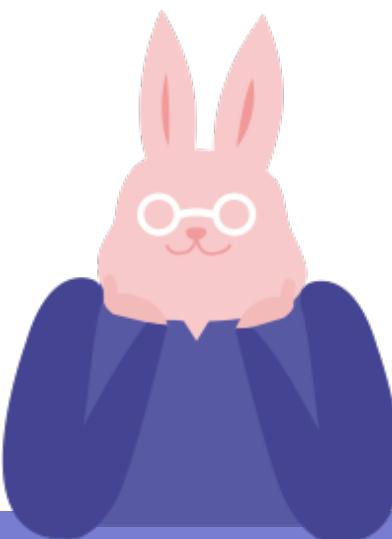
j = idx+1

While (p[j] != ')')

j = Check(p, j)

j++

return j

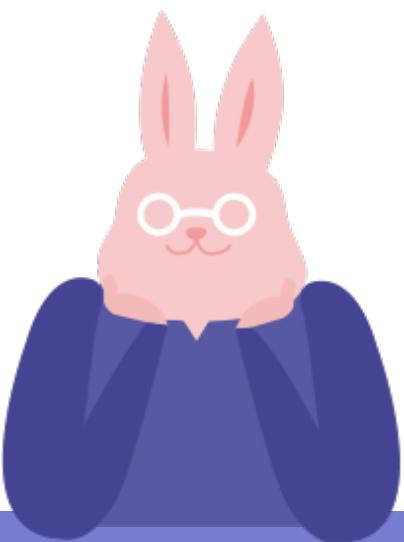


# [활동문제 3] 괄호 짹 맞추기

3. 정답이 어느 함수의 값인지를 찾는다
  - Check(p, 0) ?

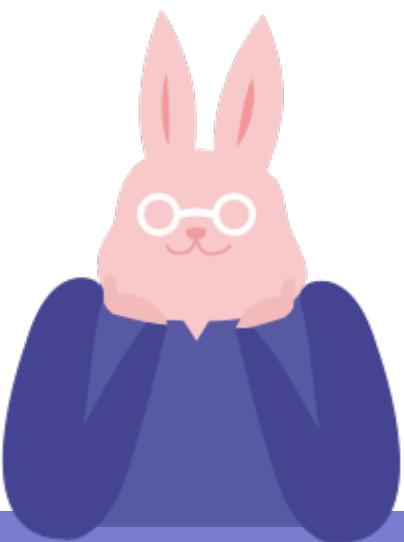
```
if Check (p, 0) = len (p) -1  
    return True  
else  
    return False
```

( ( ( ( ) ( ) ) ) ( ( ) ( ) ) )



# [활동문제 3] 괄호 짹 맞추기

- 코딩!



# [활동문제 4] 순열 출력하기

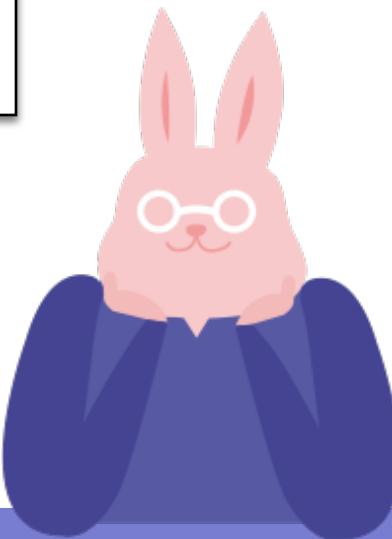
문제 : n, m이 주어질 때  $nPm$  의 모든 경우의 수를 출력하시오

입력의 예

4 3

출력의 예

abc  
abd  
acb  
acd  
adb  
adc  
bac  
...  
dcba



# [활동문제 4] 순열 출력하기

1. 값을 구할 함수 f를 정의한다

- P(buf, idx, n, m)

: buf[idx:(m-1)] 까지를 채워 출력하는 함수

2. f의 값을 구한다. 점화식의 형태를 얻을 것이다.

- $P(buf, idx, n, m) = \begin{cases} \text{if } idx == m : \\ \quad \text{print buf} \\ \text{else :} \\ \quad \text{for } i \text{ in } ('a' \dots ('a' + n)) : \\ \quad \quad \text{if } (!\text{isDuplicate}(buf, i)) : \\ \quad \quad \quad buf[idx] = i \\ \quad \quad \quad P(buf, idx+1, n, m) \\ \quad \quad \quad buf[idx] = 0 \end{cases}$



# [활동문제 4] 순열 출력하기

- P(buf, 0, 4, 3)

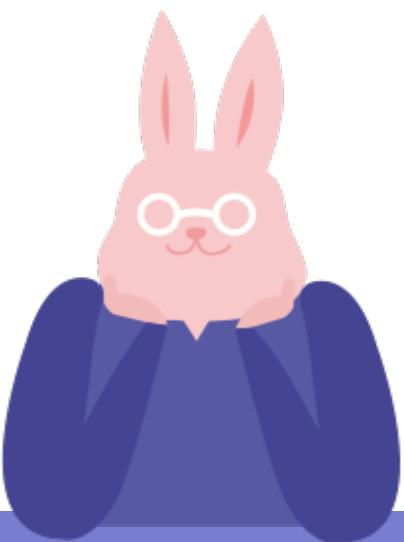
```
if idx == m :  
    print buf  
else :  
    for i in ('a'...('a' + n)) :  
        if(!isDuplicate(buf, i)) :  
            buf[idx] = i  
            P(buf, idx+1, n, m)  
            buf[idx] = 0
```



## [활동문제 4] 순열 출력하기

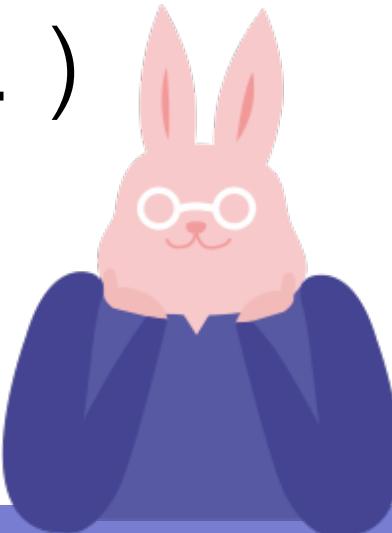
3. 정답이 어느 함수의 값인지를 찾는다

- $P(\text{buf}, 0, n, m)$  by definition



# 결론

- 함수를 최소 단위로 하여 **추상적 논리**에 집중하자
- 각 함수가 수행하는 일을 명확히 하자
- 재귀함수의 디자인
  1. 값을 구할 함수  $f$ 를 정의한다
  2.  $f$ 의 값을 구한다. 점화식의 형태를 얻을 것이다.
  3. 정답이 어느 함수의 값인지를 찾는다.
- 함수 정의에 의하여 모든게 결정된다 ( 문제 풀이 가능성, 난이도, ... )



감사합니다 😊

신현규

E-mail : [hyungyu.sh@kaist.ac.kr](mailto:hyungyu.sh@kaist.ac.kr)

Kakao : yougatup

