forbeginners.html workshop.css

Programming ‘Python’ {

[Data Structure]

< Kim_jungyun>

}

Chapter 2. 재귀

Programming Language

forbeginners.html workshop.css

Table Of 'Contents' {

01 재귀함수 (recursion)

<재귀함수, 재귀함수 종료조건, 구현 예제>

02 백준 코딩 문제집

"""

https://www.acmicpc.net/workbook/view/12664

"""

}

Programming Language

forbeginners.html

workshop.css

01 {

[재귀함수 (recursion)]

< 재귀함수, 재귀함수 종료조건,
구현 예제 >

}

Programming Language

forbeginners.html workshop.css

재귀 함수 {

‘재귀 함수란 자기 자신을 다시 호출하는 함수를 의미한다.’

<예시>

```
def recursive_function():
    print(‘recursion’)
    recursive_function()
recursive_function()
```

과 같은 코드를 실행하면 종료조건이 없는 재귀함수이므로
recursion이 무한히 출력된다.

}

forbeginners.html workshop.css

재귀 함수 종료조건 {

‘재귀 함수에 종료조건을 부여해 유한하게 함수를 호출함.’

<예시>

```
def recursive_function(i):
    if i==100:
        return
    print(i,‘번째 recursion’, i+1, ‘번째 recursion을 호출’)
    recursive_function(i+1)

recursive_function(1)
```

100번 호출되면, 종료되도록 조건 생성

}

Programming Language

forbeginners.html

workshop.css

재귀함수 구현예제 {

```
'재귀적으로 구현한 n!'

def factorial_recursive(n):
    if n<=1:
        return 1
    return n*factorial_recursive(n-1)
```

```
'반복적으로 구현한 n!'

def factorial_iterative(n):
    result = 1
    for i in range(1, n+1):
        result *= 1
    return result
```

}

forbeginners.html

workshop.css

02 {

[백준 코딩 문제집]

< https://www.acmicpc.net/workbook/view/12664 >

<문제는 거의 못 풀었는데,
괜찮아보이는 거 위주로 뽑음>

}

Programming Language

forbeginners.html workshop.css

[25501] 재귀의 귀재{

‘팰린드롬(Palindrome)이란, 앞에서부터 읽었을 때와
뒤에서부터 읽었을 때가 같은 문자열을 말한다.’

예제 입력 1	예제 출력 1
AAA	1 2
ABBA	1 3
ABABA	1 3
ABCA	0 2
PALINDROME	0 1

Palindrome인 경우,
처음으로 1을 출력하고,
아니면 0을 출력

띄워쓴 후, recursion
검사 횟수를 출력한다.

}

Programming Language

forbeginners.html

workshop.css

How to solve{

```python
st=input()
cnt=0

def recursion(s, l, r):
    global cnt #전역변수를 함수에서 사용하는 법
    cnt+=1
    if l >= r: return 1
    elif s[l] != s[r]: return 0
    else: return recursion(s, l+1, r-1)

def isPalindrome(s):
    return recursion(s, 0, len(s)-1)

print(isPalindrome(st),cnt) #,가 자동으로 띄어쓰기 하나 됨
```

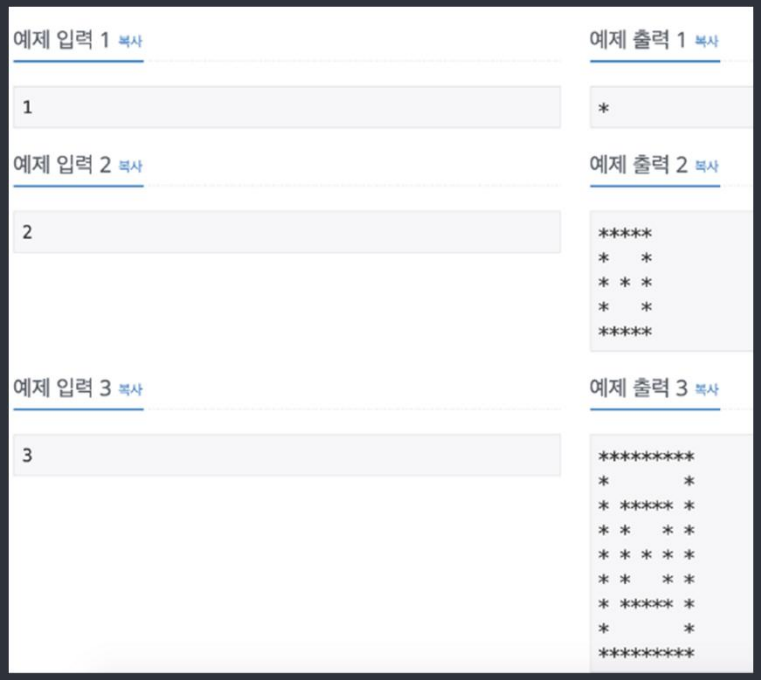
}

<답은 출력되는데 실패뜸(?)>

forbeginners.html

workshop.css

[10994] 재귀로 별찍기{



예제 입력 1 복사

1

예제 출력 1 복사

*

예제 입력 2 복사

2

예제 출력 2 복사

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

예제 입력 3 복사

3

예제 출력 3 복사

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

}

Programming Language

forbeginners.html

workshop.css

How to solve{

```
def draw(n, idx):
    if n == 1: #n이 1이면 하나만 출력
        starMap[idx][idx] = '*'
        return
    l = 4 * n -3 #가장 밖 테두리 길이 (4n-3 x 4n-3모양 정사각형)
    for i in range(idx, l+idx): #x,y좌표를 2씩 더해줌(작은 사각형)
        starMap[idx][i] = '*'
        starMap[idx+l-1][i] = '*'
        starMap[i][idx] = '*'
        starMap[i][idx+l-1] = '*'

    return draw(n-1, idx+2) #재귀로 변경할 조건
```

```
n = int(input())
lens = 4 * n -3

starMap = [[' '] * lens for _ in range(lens)]
#2차원 배열을 바로 만드는 법
draw(n,0) #idx를 재귀로 변경해야할 것

#출력
for i in range(lens):
    for j in range(lens):
        print(starMap[i][j], end="")
    print()
```

list comprehension

} 4n-3이 가장 밖테두리인데 한 칸씩 안의 별을 재귀함수를 통해 그려나가는 것 (별모양이 정사각형)