

## 고급통계프로그래밍 과제 #4

2017580034 통계학과 이유민

---

### Ex 8.4

In [1]:

```
def find(word, letter, a): # 3rd param = index
    index = a #받아온 index
    while index < len(word):
        if word[index] == letter:
            return index
        index = index + 1
    return -1
```

In [2]:

```
find('thinkpython', 'p', 0)
```

Out[2]:

5

In [3]:

```
find('thinkpython', 'p', 6)
```

Out[3]:

-1

---

### Ex 8.11

In [4]:

```
#8.11 -(1)
def any_lowercase1(s):
    for c in s:
        if c.islower():
            return True
        else:
            return False
```

In [5]:

```
any_lowercase1('banana')
```

Out[5]:

True

In [6]:

```
any_lowercase1('Banana')
```

Out[6]:

False

In [7]:

```
any_lowercase1('Ba1nana')
```

Out[7]:

False

`any_lowercase1` 함수는 `for`문으로 문자열을 돌면서 소문자인지를 판별한다. 따라서 문자열 내에 하나라도 소문자가 아닌 경우 `FALSE`를 반환하기 때문에 잘못된 함수이다.

In [8]:

```
#8.11 -(2)
def any_lowercase2(s):
    for c in s:
        if 'c'.islower():
            return 'True'
        else:
            return 'False'
```

In [9]:

```
any_lowercase2('banana')
```

Out[9]:

'True'

In [10]:

```
any_lowercase2('Banana')
```

Out[10]:

'True'

In [11]:

```
any_lowercase2('Ba1nana')
```

Out[11]:

'True'

(1)과 같이 for문을 돌면서 판별하고 있으나, 'c'.islower()은 문자 c를 의미하기 때문에 s에 들어가는 값과는 상관 없이 모든 값에 대해 TRUE를 반환하므로 잘못된 함수이다.

In [12]:

```
#8.11 -(3)
def any_lowercase3(s):
    for c in s:
        flag = c.islower()
    return flag
```

In [13]:

```
any_lowercase3('banana')
```

Out[13]:

True

In [14]:

```
any_lowercase3('Banana')
```

Out[14]:

True

In [15]:

```
any_lowercase3('Ba1nanA')
```

Out[15]:

False

any\_lowercase3 함수는 for문을 돌면서 소문자를 판별한 결과를 flag에 저장한다. 하지만 for문을 돌면서 flag의 값이 바뀌므로 마지막 flag의 값만 반환하게 된다. 따라서 s의 마지막 값이 소문자면 True, 대문자면 False를 반환하므로 잘못된 함수이다.

In [16]:

```
#8.11 -(4)
def any_lowercase4(s):
    flag = False
    for c in s:
        flag = flag or c.islower()
    return flag
```

In [17]:

```
any_lowercase4('banana')
```

Out[17]:

True

In [18]:

```
any_lowercase4('Banana')
```

Out[18]:

True

In [19]:

```
any_lowercase4('Ba1nanA')
```

Out[19]:

True

flag=False로 두고 for문을 시작하므로 소문자가 발견되지 않는 경우 함수는 False를 반환한다. 문자열에 소문자가 포함되어 있는 경우 True를 반환하고, 이는 flag값과 or로 연결되어 있으므로 올바른 함수이다.

In [20]:

```
#8.11 -(5)
def any_lowercase5(s):
    for c in s:
        if not c.islower():
            return False
    return True
```

In [21]:

```
any_lowercase5('banana')
```

Out[21]:

True

In [22]:

```
any_lowercase5('Banana')
```

Out[22]:

False

In [23]:

```
any_lowercase5('Ba1nanA')
```

Out[23]:

False

any\_lowercase5 함수는 not c.islower()인 경우 False를 반환하기 때문에 문자열이 모두 소문자인 경우만 True를 반환하므로 잘못된 함수이다.

---

## Ex 8.12

In [24]:

```
# 출력을 소문자로만 고려하는 경우 (문제에서 주어진 예시들이 모두 소문자이므로)
def rotate_word(word, n):
    words = 'abcdefghijklmnopqrstuvwxyz'
    x = []
    for i in word:
        y = words.find(i) + n
        x.append(words[y])
    return ''.join(x)
```

In [25]:

```
# 소문자&대문자를 모두 고려하여 출력하는 경우
def rotate_word(word, n):
    l_words = 'abcdefghijklmnopqrstuvwxyz'
    h_words = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    x = []
    for i in word:
        if i.islower() == True:
            y = l_words.find(i) + n
            x.append(l_words[y])
        else:
            y = h_words.find(i) + n
            x.append(h_words[y])
    return ''.join(x)
```

In [26]:

```
print(rotate_word('MeIoN', -10))
```

CubeD

In [27]:

```
print(rotate_word('melon', -10))
```

cubed

In [28]:

```
print(rotate_word('CHEER', 7))
```

JOLLY

In [29]:

```
print(rotate_word('cheer', 7))
```

jolly