



텍스트 파일을 열어서 파일 안의 스페이스 문자의 개수와 탭의 개수를 세는 프로그램을 작성하여 보자.

파일 이름을 입력하시오: proverbs.txt
스페이스 수 = 20, 탭의 수 = 0



Solution

```
def parse_file(path):
    infile = open(path)
    spaces = 0
    tabs = 0
    for line in infile:
        spaces += line.count(' ')
        tabs += line.count('\t')
    infile.close()

    return spaces, tabs

filename = input("파일 이름을 입력하시오: ");
spaces, tabs = parse_file(filename)
print("스페이스 수 = %d, 탭의 수 = %d" % (spaces, tabs))
```



텍스트 파일을 열어서 각 줄의 앞에 번호를 매겨서 다시 파일에 쓰는 프로그램을 작성해보자.

proverbs.txt

```
All's well that ends well.
Bad news travels fast.
Well begun is half done.
Birds of a feather flock together.
```

output.txt

```
1: All's well that ends well.
2: Bad news travels fast.
3: Well begun is half done.
4: Birds of a feather flock together.
```

Solution

```
infile = open("proverbs.txt")
outfile = open("output.txt", "w")
i = 1
for line in infile:
    outfile.write(str(i) + ": " + line)
    i = i + 1
infile.close()
outfile.close()
```

번호를 문자열로 변환하여 먼저 쓰고 :을 붙인 후에 원래의 문장을 추가한다.



각 문자 횟수 세기

파일 안의 각 문자들이 몇 번이나 나타나는지를 세는 프로그램을 작성하자.

proverbs.txt

```
All's well that ends well.
Bad news travels fast.
Well begun is half done.
Birds of a feather flock together.
```

```
{' ': 16, 'e': 12, 'o': 4, 'a': 7, 'u': 1, 'n': 4, 'k': 1,
'A': 1, 'r': 4, 'g': 2, 's': 7, 'b': 1, 'd': 4, 'v': 1, 'f': 5,
'w': 3, 'B': 2, 'h': 4, 'i': 2, 't': 7, 'l': 11, 'W': 1, '.': 4,
'"': 1, 'c': 1}
```

Solution

```
filename = input("파일명을 입력하세요: ").strip()
infile = open(filename, "r") # 파일을 연다.

freqs = {}

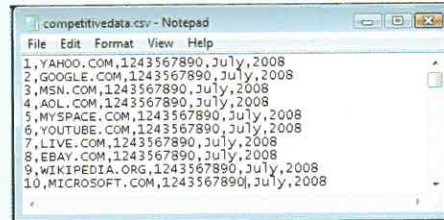
# 파일의 각 줄에 대하여 문자를 추출한다. 각 문자를 사전에 추가한다.
for line in infile:
    for char in line.strip():
        if char in freqs:
            freqs[char] += 1
        else:
            freqs[char] = 1
print(freqs)
infile.close()
```

딕셔너리에 있으면 하나 증가

딕셔너리에 없으면 1로 초기화



CSV(Comma Separated Values) 형식은 엑셀과 같은 스프레드 쉬트나 데이터베이스에서 가장 널리 사용되는 입출력 형식이다. 파이썬은 CSV 형식을 읽기 위해서 csv라고 하는 모듈을 제공한다. 이 모듈을 이용하면 CSV 파일을 쉽게 읽을 수 있다. 우리는 연습 삼아서 CSV 형식의 파일을 읽는 코드를 작성하여 보자.



data.csv

```
1/2/2014,5,8,red
1/3/2014,5,2,green
1/4/2014,9,1,blue
```

위의 파일을 읽어서 다음과 같이 출력하면 된다.

```
1/2/2014,5,8,red
  1/2/2014
    5
    8
    red
1/3/2014,5,2,green
  1/3/2014
    5
    2
    green
1/4/2014,9,1,blue
  1/4/2014
    9
    1
    blue
```

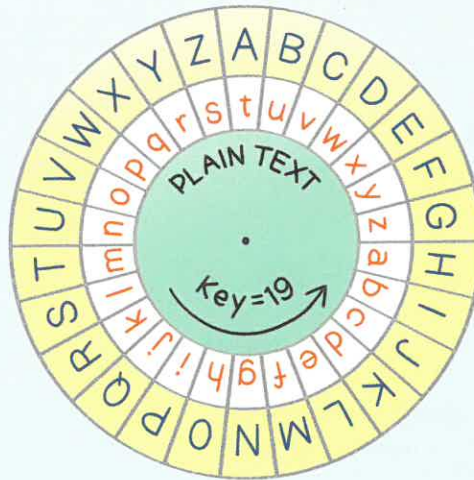
Solution

```
# 파일을 연다.  
f = open("C:\\test.csv", "r")  
  
# 파일 안의 각 줄을 처리한다.  
for line in f.readlines():  
  
    # 공백 문자를 없앤다.  
    line = line.strip()  
  
    # 줄을 출력한다.  
    print(line)  
  
    # 줄을 쉼표로 분리한다.  
    parts = line.split(",")  
  
    # 각 줄의 필드를 출력한다.  
    for part in parts:  
        print("    ", part)
```




시저 암호를 구현하여 보자. 로마의 유명한 정치가였던 줄리어스 시저(Julius Caesar, 100-44 B.C.)는 친지들에게 비밀리에 편지를 보내고자 할 때 다른 사람들이 알아보지 못하도록 문자들을 다른 문자들로 치환하였다. 시저 암호의 규칙을 표로 그려 보면 다음과 같다.

| 평문 | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 암호문 | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |



예를 들어 평문 “come to me”는 “FRPH WR PH”으로 바뀐다. 시저 암호 방식을 이용하여 파일을 암호화하고 복호화하는 프로그램을 작성하라.

평문: the language of truth is simple.
 암호문: wkh odqjxdjh ri wuxwk lv vlpsoh.
 복호문: the language of truth is simple.



```

key = 'abcdefghijklmnopqrstuvwxyz'

# 평문을 받아서 암호화하고 암호문을 반환한다.
def encrypt(n, plaintext):
    result = ''

    for l in plaintext.lower():
        try:
            i = (key.index(l) + n) % 26
            result += key[i]
        except ValueError:
            result += l

    return result.lower()

# 암호문을 받아서 복호화하고 평문을 반환한다.
def decrypt(n, ciphertext):
    result = ''

    for l in ciphertext:
        try:
            i = (key.index(l) - n) % 26
            result += key[i]
        except ValueError:
            result += l

    return result

n = 3
encrypted = encrypt(n, text)
text='the language of truth is simple.'
decrypted = decrypt(n, encrypted)
print ('평문: ', text)
print ('암호문: ', encrypted)
print ('복호문: ', decrypted)

```

오류처리구조이다. try블록에서 오류가 일어나면 except블록에서 해결한다. 12.7절을 참고하자.

```

평문: the language of truth is simple.
암호문: wkh odqjxdjh ri wuxwk lv vlpsoh.
복호문: the language of truth is simple.

```

