

비주얼 프로그래밍

김승태 교수님

과제 4

20176359 신수현

[문제 1]

전자사전 만들기

1. 문제의 해결 방안

첫 while 문에 먼저 파일을 'r'형태로 열어줍니다. 그리고 검색할 단어를 입력받을 input문을 작성합니다. 추가적으로 검색한 단어가 없을 경우를 대비하기 위해 flag형태인 found = False를 설정해줍니다.

이후 두번째 while문을 작성하고, readline()을 해줍니다. input받은 검색할 단어를 word라고 설정했을 때, word == v.split(':')[0][- 1] (여기서 v= f.readline() 이다)이면, found를 True로 바꾸고 (단어를 찾았기 때문에 True로 변경합니다.) 단어와 뜻을 출력해주기 위해 v 를 출력해줍니다.

※ 여기서, v.split(':')[0][- 1] 의 의미는 단어장을 보면, abaca : n.마닐라삼(필리핀 주산) 이런식으로 나열되어 있는데, readline()마다 ':' 로 split 해주었을 때, 왼쪽에 있는게 단어이며, 오른쪽은 단어의 뜻입니다. 왼쪽 index는 [0] 이고, 공백을 없애고 단어만을 뽑기 위해 [- 1]을 해주었습니다.

오류가 있을 때를 대비하기 위해 두번째 while문을 try로 감싸고, except문을 마지막에 써주었습니다.

2. 소스코드

while True:

```
f = open("dict_test.txt", 'r')
```

```
word = input('단어?')
```

```
found = False
```

```
try:
```

```
    while True:
```

```
        v=f.readline()
```

```
        if v == "":
```

```
            break
```

```
        if word == v.split(':')[0][- 1]:
```

```
            print(v[- 1])
```

```
            found = True
```

```
            break
```

```
    if found == False:
```

```

        print(word + ' Not Found')

except TypeError:
    print('문제가 있어요')

f.close()

```

3. 결과 화면 캡처

[문제 2]

끝말 잇기

1. 문제 해결 방안

첫번 째 while문 작성전에 무조건 apple이 첫 단어이어야 하기 때문에 word = 'apple'로 지정해줍니다. 그리고 중복되는 단어를 찾기 위해 빈 리스트 word_list를 정의해줍니다.

첫번 째 while 문 안에는 사전에 있는 단어와 없는 단어를 구별하기 위해 flag 역할인 found = False를 설정해주고, 파일을 'r' 읽기 형태로 열어주고, 끝말잇기 단어를 입력받기 위해 input을 사용합니다. 단어가 5글자 이상인 것을 비교하기 위해 if문과 len함수를 이용하고, 단어가 5글자 미만인 경우에도 마찬가지로 작성합니다.

두번 째 while문 안에서는 v = f.readline()을 사용해, 한문장씩 읽어줍니다. 그리고 v.split에 단어 뜻을 제외한 단어 자체만 담아줍니다. 먼저 단어 길이가 5인것이 확인되면, 내가 입력한 단어의 첫번째 자리 글자와 전 단어의 마지막 단어가 일치하는지를 확인해줍니다(둘다 if문 사용). 일치하지 않는다면 “끝말잇기가 안되는 단어라고 출력합니다. 맞다면 “정답입니다”를 출력해주고, 내가 입력한 단어를 word_list에 append해주고 found=True 로 변경합니다. 만약에 입력한 단어가 word_list에 있는 단어라면 중복된 단어라고 출력합니다.

마지막으로 found = False이면 사전에 없는 단어라고 출력해줍니다.

2. 소스 코드

```
word = 'apple'
word_list = []
while True:
    found = False
    f = open("dict_test.txt", 'r')
    ques = input('%s 끝말잇기?' % word)

    #단어 짧은거
    if len(ques) < 5:
        print('단어가 짧아요(%s의 끝말을 이으세요).' % word)
        continue

    #단어가 긴거
    elif len(ques) > 5 :
        print('단어가 길어요(%s의 끝말을 이으세요).' % word)
        continue

    while True:
        v = f.readline()
        v_split = v.split(':')[0][-1]

        # 파일을 끝까지 읽은 경우 break
        if v == "":
            break

        if len(ques) == 5:
            #끝자와 첫글자가 맞는지 체크
            if ques[0] == word[4]:
                if ques == v_split:
                    #중복된 단어 비교하기
                    if ques in word_list:
                        print('중복된 단어입니다(%s의 끝말을 이으세요).' % word)
```

```

        found = True
        break
#중복되지 않은 단어일 경우
else:
    print("정답입니다(%s 의 끝말을 이으세요)." % ques)
    word_list.append(ques)
    word = ques
    found = True
    break

else:
    print("뵙. 끝말잇기가 안되는 단어입니다.")
    found = True
    break

#사전에 없는 단어
if found == False:
    print("사전에 없는 단어입니다(%s 의 끝말을 이으세요)." % word)

f.close()

```

3. 결과 화면 캡처

```

===== RESTART: C:\Users\queenSSH\Desktop\학교\금융공학\비주얼프로그래밍\과제\과제 4\과제
연습.py =====
apple 끝말잇기?eagle
정답입니다(eagle 의 끝말을 이으세요).
eagle 끝말잇기?error
정답입니다(error 의 끝말을 이으세요).
error 끝말잇기?orange
단어가 길어요(error의 끝말을 이으세요).
error 끝말잇기?rose
단어가 짧아요(error의 끝말을 이으세요).
error 끝말잇기?range
정답입니다(range 의 끝말을 이으세요).
range 끝말잇기?efqwe
사전에 없는 단어입니다(range 의 끝말을 이으세요).
range 끝말잇기?apple
뵙. 끝말잇기가 안되는 단어입니다.
range 끝말잇기?erase
정답입니다(erase 의 끝말을 이으세요).
erase 끝말잇기?eagle
중복된 단어입니다(erase의 끝말을 이으세요).
erase 끝말잇기?

```

[문제 3]

기후 분석

1. 문제의 해결 방안

먼저, “데이터 파일이 달라지면 표시되는 도시가 달라질 수 도 있다”는 걸 감안하기 위해서 code를 담을 배열과 code와 province를 맵핑할 딕셔너리 자료형을 선언합니다. 이후 code_arr와 code_dic을 이용하여 사용자 메뉴를 만듭니다.

첫번째 while 문에서는 파일을 처음부터 읽기 시작합니다. 첫번째 줄은 컬럼이므로 넘어가고, input을 이용해 도시 선택 번호를 입력받습니다. 이전년월과 현재읽은데이터년월 변수, 년월이 바뀔 때 그 전까지 카운트한 수를 체크하기 위한 변수 cnt, 평균 기온의 합 저장변수, 강수량 합 저장변수, 데이터 담을 배열 등을 전부 빈 리스트나 빈 문자열 형태로 정의합니다.

두번째 while문에서는 첫번째 줄을 건너뛴 두번째 줄부터 readline()을 해줍니다. 처음에 파일을 다 읽은 경우에 누적된 변수 값을 마지막으로 배열에 넣고 반복문에서 빠져나오기 위해 if v == "" 을 설정합니다. if문 밖에서, v를 ','로 split해준 것을 v로 잡고, code는 v의 0번째 index, date는 2번째 index를 '-'로 또한번 split한 값으로 저장합니다. (평균 기온이 빈 값인 경우를 대비하기 위해 v[3] == ""이면 continue 하는 줄을 추가합니다.) 처음에 데이터를 읽은 경우에는 이전년월이 존재하지 않으므로 last_ym을 current_ym으로 덮어줍니다. 만약 last 와 current ym가 서로 다르다면, date_arr에 last_ym을 append해주고 (몇년 몇월인지), 마찬가지로 temp_arr, prec_arr 를 append합니다. 그리고 다음 달로 넘어가기 전에 초기화를 해줍니다. 만약, last와 current ym이 같다면 계속 sum에 더해주는 식으로 진행합니다.

2. 소스코드

code와 province를 맵핑할 딕셔너리 자료형

```
code_dic = {}
```

code를 담을 배열

```
code_arr = []
```

```
f = open('weather.csv', 'r')
```

```
v = f.readline().split(",")
```

중복을 제거한 코드 리스트 받아온 후 코드 오름차순으로 정렬

while True:

 v = f.readline()

 if v == "":

 break

 v = v.split(",")

 code = int(v[0])

 province = v[1]

 if code not in code_arr:

 code_arr.append(code)

 code_dic[code] = province

code_arr.sort()

사용자 메뉴 만들기

menu_str = ""

for i in range(len(code_arr)):

 if i == 0:

 menu_str = str(i + 1) + ":" + code_dic[code_arr[i]]

 else:

 menu_str = menu_str + ", " + str(i + 1) + ":" + code_dic[code_arr[i]]

while True:

 f.seek(0)

 f.readline()

 sel = input("도시를 선택하세요 (%s) " % menu_str)

선택한 도시의 코드 얻어오기

sel_code = code_arr[int(sel) - 1]

이전년월과 현재읽은데이터년월

last_ym = ""

current_ym = ""

cnt = 0

```
sum_avg_temp = 0.0
```

```
sum_prec = 0.0
```

```
date_arr = []
```

```
temp_arr = []
```

```
prec_arr = []
```

```
while True:
```

```
    v = f.readline()
```

```
    #파일을 끝까지 읽은 경우, 지금까지 누적된 변수 값을 마지막으로 배열에 넣고 break
```

```
    if v == "":
```

```
        date_arr.append(last_ym)
```

```
        temp_arr.append(sum_avg_temp/cnt)
```

```
        prec_arr.append(sum_prec)
```

```
        break
```

```
    v = v.split(",")
```

```
    code = int(v[0])
```

```
    date = v[2].split("- ")
```

```
    # 선택한 도시코드와 동일한 데이터인가 체크
```

```
    if sel_code == code:
```

```
        current_ym = date[0] + "년 " + str(int(date[1])) + "월"
```

```
        if v[3] == "":
```

```
            continue
```

```
        avg_temp = float(v[3])
```

```
        prec = float(v[6])
```

```
        if last_ym == "":
```

```
            # 현재데이터년월을 이전년월에 갱신
```

```
            last_ym = current_ym
```

```
            sum_avg_temp = avg_temp
```

```
            sum_prec = prec
```

```
            cnt = 1
```


이전년월과 현재데이터년월이 다른 경우에는 지금까지 저장한 값을 배열에 담고 변수 초기화.

else:

if last_ym != current_ym:

date_arr.append(last_ym)

temp_arr.append(sum_avg_temp / cnt)

prec_arr.append(sum_prec)

last_ym = current_ym

sum_avg_temp = 0.0

sum_prec = 0.0

cnt = 0

sum_avg_temp += avg_temp

sum_prec += prec

cnt += 1

저장한 값 출력

print('\n%s 기후 분석\n\n평균 기온\n\n월별 강수량(mm)' % code_dic[sel_code])

for i in range(len(date_arr)):

print(date_arr[i] + '\n\n' + str(round(temp_arr[i], 1)) + '\n\n' + str(round(prec_arr[i], 1)))

print('\n')

3.결과 화면 캡처

===== RESTART: C:\Users\queenSSH\Desktop\학교\금융공학\비주얼프로그래밍\과제\과제 4\과제 연습.py =====

도시를 선택하세요 (1:Seoul, 2:Busan, 3:Daegu, 4:Gwangju, 5:Incheon, 6:Daejeon, 7:Ulsan, 8:Gyeonggi-do, 9:Gangwon-do, 10:Chungcheongbuk-do, 11:Chungcheongnam-do, 12:Jeollabuk-do, 13:Jeollanam-do, 14:Gyeongsangbuk-do, 15:Gyeongsangnam-do, 16:Jeju-do) 1

| Seoul 기후 분석 | 평균 | 기온 | 월별 강수량(mm) |
|-------------|------|----|------------|
| 2016년 1월 | -3.2 | | 1.0 |
| 2016년 2월 | 0.2 | | 47.6 |
| 2016년 3월 | 7.0 | | 40.5 |
| 2016년 4월 | 14.1 | | 76.8 |
| 2016년 5월 | 19.6 | | 160.5 |
| 2016년 6월 | 23.6 | | 54.4 |
| 2016년 7월 | 26.2 | | 358.2 |
| 2016년 8월 | 28.0 | | 67.1 |
| 2016년 9월 | 23.1 | | 33.0 |
| 2016년 10월 | 16.1 | | 74.8 |
| 2016년 11월 | 6.8 | | 16.7 |
| 2016년 12월 | 1.2 | | 61.1 |
| 2017년 1월 | -1.8 | | 14.9 |
| 2017년 2월 | -0.2 | | 11.1 |
| 2017년 3월 | 6.3 | | 7.9 |
| 2017년 4월 | 13.9 | | 61.6 |
| 2017년 5월 | 19.5 | | 16.1 |
| 2017년 6월 | 23.3 | | 66.6 |
| 2017년 7월 | 26.9 | | 621.0 |
| 2017년 8월 | 25.9 | | 297.0 |
| 2017년 9월 | 22.1 | | 35.0 |
| 2017년 10월 | 16.4 | | 26.5 |
| 2017년 11월 | 5.6 | | 40.7 |
| 2017년 12월 | -1.9 | | 34.8 |

| | | | |
|-----------|------|--|-------|
| 2018년 1월 | -4.0 | | 8.5 |
| 2018년 2월 | -1.6 | | 29.6 |
| 2018년 3월 | 8.1 | | 49.5 |
| 2018년 4월 | 13.0 | | 130.3 |
| 2018년 5월 | 18.2 | | 222.0 |
| 2018년 6월 | 23.1 | | 171.5 |
| 2018년 7월 | 27.8 | | 185.6 |
| 2018년 8월 | 28.8 | | 202.6 |
| 2018년 9월 | 21.5 | | 68.5 |
| 2018년 10월 | 13.1 | | 120.5 |
| 2018년 11월 | 7.8 | | 79.1 |
| 2018년 12월 | -0.6 | | 16.4 |
| 2019년 1월 | -0.9 | | 0.0 |
| 2019년 2월 | 1.0 | | 23.8 |
| 2019년 3월 | 7.1 | | 26.8 |
| 2019년 4월 | 12.1 | | 47.3 |
| 2019년 5월 | 19.4 | | 37.8 |
| 2019년 6월 | 22.5 | | 74.0 |
| 2019년 7월 | 25.9 | | 194.4 |
| 2019년 8월 | 27.2 | | 190.5 |
| 2019년 9월 | 22.6 | | 139.8 |
| 2019년 10월 | 16.4 | | 55.5 |
| 2019년 11월 | 7.6 | | 78.8 |
| 2019년 12월 | 1.4 | | 22.6 |
| 2020년 1월 | 1.6 | | 60.5 |
| 2020년 2월 | 2.5 | | 53.1 |
| 2020년 3월 | 7.7 | | 16.3 |
| 2020년 4월 | 10.9 | | 16.9 |