Python Programming and Practice

Anagram generator & interpreter

Proposal

Progress Report : 1

Date : 23/11/26
Name : Jingang Kim
ID :233575

1. Introduction

1) Background

Anagram is a literary device that fosters fun by hiding clues in words or sentences in mystery novels. There are difficulties in generating and interpreting anagram because anagram has to be created by changing the order of spelling one by one. To solve this problem, we need a program that automatically makes anagram and recommends it to our customers.

2) Project goal

It aims to create a program that finds words corresponding to words input by users in English dictionaries and recommends them.

3) Differences from existing programs

Existing programs supported anagram of words mostly in English and not Korean. And it simply mixed up the spelling of words and didn't just extract the combination of meaningful words. We will support Korean Anagram and only recommend combinations of meaningful words, so there is a difference from existing programs.

2. Functional Requirement

1) Function to choose between English and Korean

- Select an anagram to generate between English and Korean

2) Function to receive and save words from users

- Receive words from the user and save them one letter each in the list

(1) Save English words in lowercase on the list

(2) Hangeul stores consonants and vowels separately

3) Save words in a word dictionary

The word dictionary is stored in the txt file. Save the words in the vocabulary dictionary one letter at a time in a two-dimensional list.

(1) Hangeul stores consonants and vowels separately

4) The ability to output corresponding words from a word dictionary

- Prints the saved words if they correspond to the word dictionary saved in 2)

(1) Check correspondence by calculating the number of characters between words

3. Progress

1) Implementation of features

(1) Function to choose between English and Korean

- Input and output

Input: 0 or 1 or else

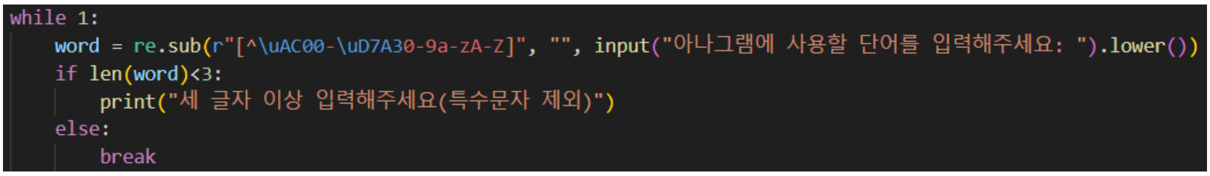
- Explanation

Determine the language by having the user choose between 0 and 1. Ask again if the input is not 0 or 1

- Apply what I learned

Loop(while), Condition,

- code screen shot



```
while 1:
    word = re.sub(r"[^\uAC00-\uD7A30-9a-zA-Z]", "", input("아나그램에 사용할 단어를 입력해주세요: ").lower())
    if len(word)<3:
        print("세 글자 이상 입력해주세요(특수문자 제외)")
    else:
        break
```

(2) Function to receive and save words from users

- Input and output

Input: user input

Output: organized string

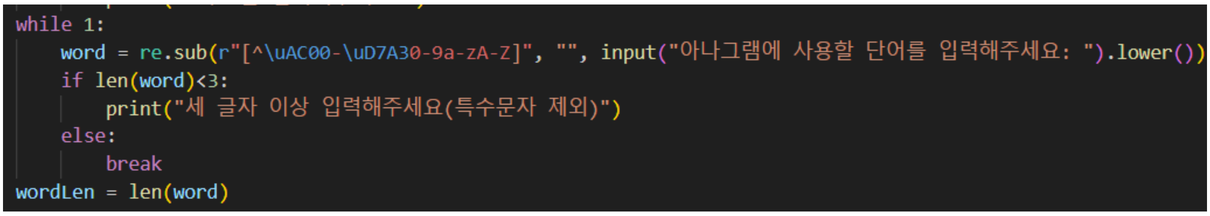
- Explanation

Receive a string from the user and remove special characters and spaces. Capital letters change to lowercase letters. If it's less than 2 letters, it's re-entered.

- Apply what I learned

Module(re), Loop(while), condition

- code screen shot



```
while 1:
    word = re.sub(r"[^\uAC00-\uD7A30-9a-zA-Z]", "", input("아나그램에 사용할 단어를 입력해주세요: ").lower())
    if len(word)<3:
        print("세 글자 이상 입력해주세요(특수문자 제외)")
    else:
        break
wordLen = len(word)
```

(3) Function to separate consonants and vowels in Korean

- Input and output

Input: previously saved variable "word"(kor)

Output: List of Korean consonants and vowels separated

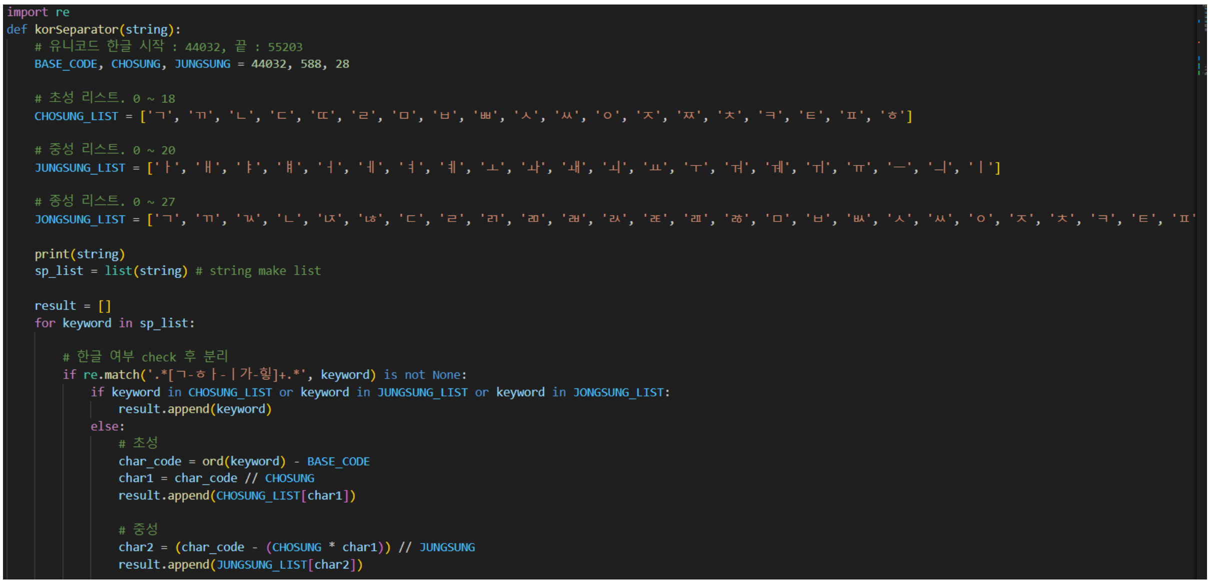
- Explanation

There are 588 Korean Unicode for each consonant. Therefore, the initial consonant can be obtained by subtracting 44032 (first letter) and dividing it by 588. In addition, vowels and consonants can be obtained by calculating the number of 588 identical initial vowels.

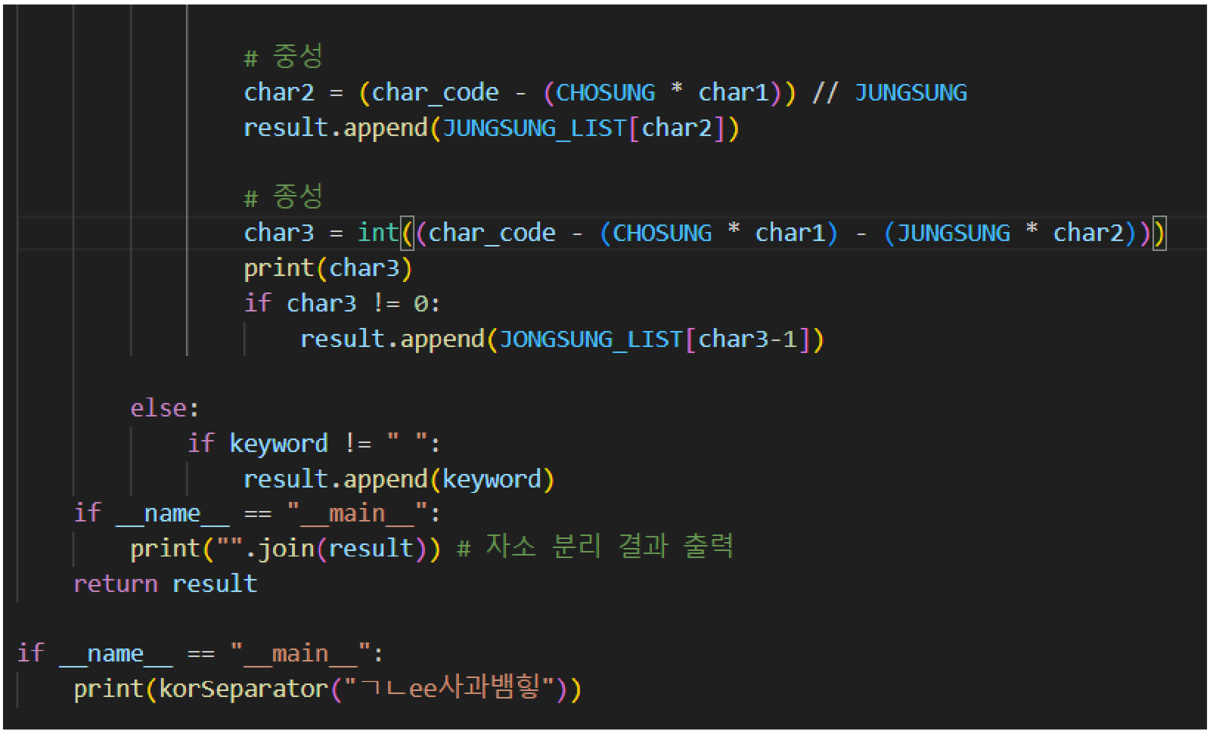
- Apply what I learned

List, module(re), function, condition

- code screen shot



```
import re
def korSeparator(string):
    # 유니코드 한글 시작 : 44032, 끝 : 55203
    BASE_CODE, CHOSUNG, JUNGSUNG = 44032, 588, 28

    # 초성 리스트. 0 ~ 18
    CHOSUNG_LIST = ['ㄱ', 'ㄲ', 'ㄴ', 'ㄷ', 'ㄸ', 'ㄹ', 'ㅁ', 'ㅂ', 'ㅃ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅉ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ', 'ㅎ']

    # 중성 리스트. 0 ~ 20
    JUNGSUNG_LIST = ['ㅏ', 'ㅐ', 'ㅑ', 'ㅒ', 'ㅓ', 'ㅔ', 'ㅕ', 'ㅖ', 'ㅗ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅛ', 'ㅜ', 'ㅝ', 'ㅞ', 'ㅟ', 'ㅠ', 'ㅡ', 'ㅢ', 'ㅣ']

    # 종성 리스트. 0 ~ 27
    JONGSUNG_LIST = ['ㄱ', 'ㄲ', 'ㄳ', 'ㄴ', 'ㄵ', 'ㄶ', 'ㄷ', 'ㄹ', 'ㄺ', 'ㄻ', 'ㄼ', 'ㄽ', 'ㄾ', 'ㄿ', 'ㅀ', 'ㅁ', 'ㅂ', 'ㅄ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ'

    print(string)
    sp_list = list(string) # string make list

    result = []
    for keyword in sp_list:

        # 한글 여부 check 후 분리
        if re.match('.*[ㄱ-ㅎㅏ-ㅣ가-힗]+.*', keyword) is not None:
            if keyword in CHOSUNG_LIST or keyword in JUNGSUNG_LIST or keyword in JONGSUNG_LIST:
                result.append(keyword)
            else:
                # 초성
                char_code = ord(keyword) - BASE_CODE
                char1 = char_code // CHOSUNG
                result.append(CHOSUNG_LIST[char1])

                # 중성
                char2 = (char_code - (CHOSUNG * char1)) // JUNGSUNG
                result.append(JUNGSUNG_LIST[char2])
```



```python
                # 중성
                char2 = (char_code - (CHOSUNG * char1)) // JUNGSUNG
                result.append(JUNGSUNG_LIST[char2])

                # 종성
                char3 = int((char_code - (CHOSUNG * char1) - (JUNGSUNG * char2)))
                print(char3)
                if char3 != 0:
                    result.append(JONGSUNG_LIST[char3-1])

        else:
            if keyword != " ":
                result.append(keyword)
    if __name__ == "__main__":
        print("".join(result)) # 자소 분리 결과 출력
    return result

if __name__ == "__main__":
    print(korSeparator("ㄱㄴee사과뱀힣"))
```

(4) Words with the same length are changed into a list form, sorted, and compared with the sorted list of words input by the user.

- Input and output

Input: user input, Words with the same length as the words entered by the user

Output: a list of words that match

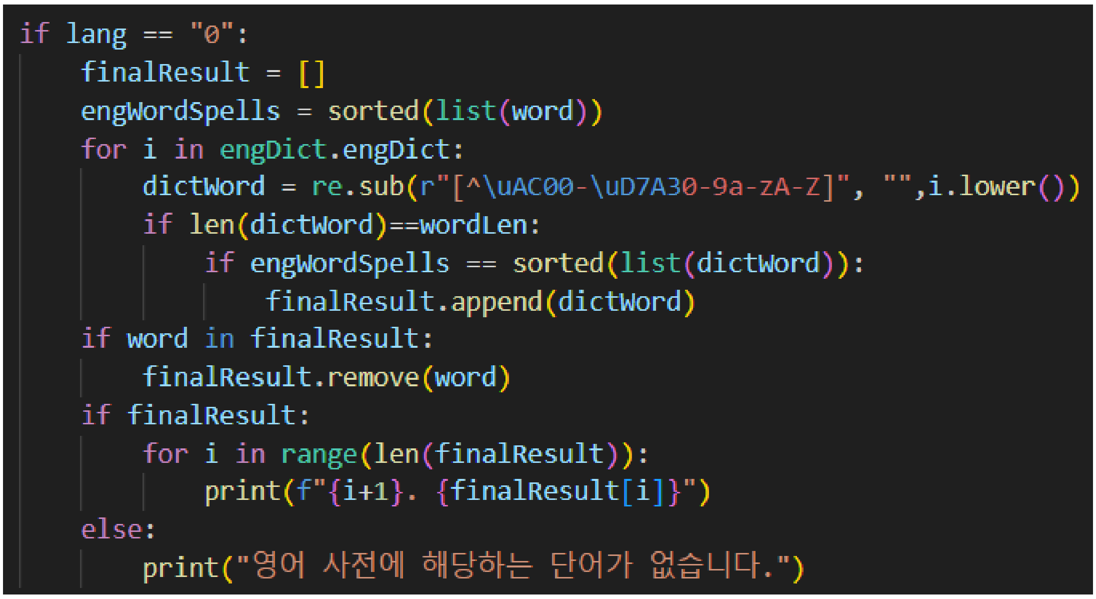
- Explanation

Words with the same length are changed into a list form, sorted, and compared with the sorted list of words input by the user.(In Korean, consonants and vowels are separated and compared)

- Apply what I learned

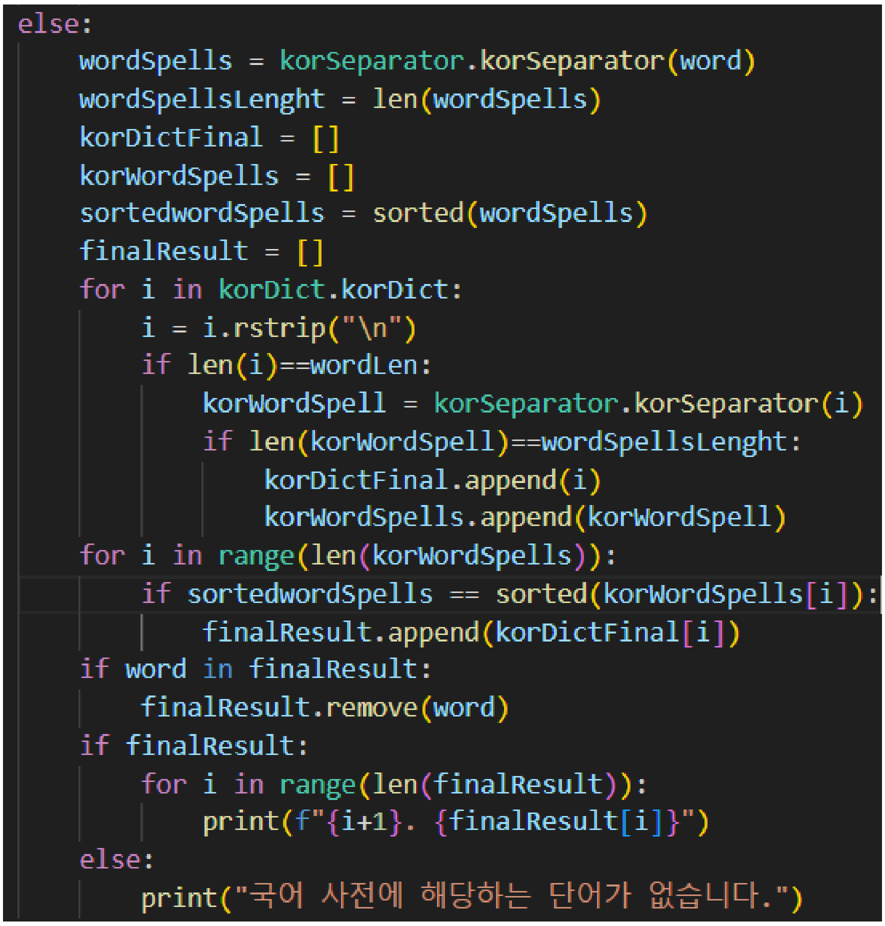
List, set, module(re), Loop, Condition

- code screen shot(eng)



```python
if lang == "0":
    finalResult = []
    engWordSpells = sorted(list(word))
    for i in engDict.engDict:
        dictWord = re.sub(r"[^\uAC00-\uD7A30-9a-zA-Z]", "",i.lower())
        if len(dictWord)==wordLen:
            if engWordSpells == sorted(list(dictWord)):
                finalResult.append(dictWord)
    if word in finalResult:
        finalResult.remove(word)
    if finalResult:
        for i in range(len(finalResult)):
            print(f"{i+1}. {finalResult[i]}")
    else:
        print("영어 사전에 해당하는 단어가 없습니다.")
```

- code screen shot(kor)



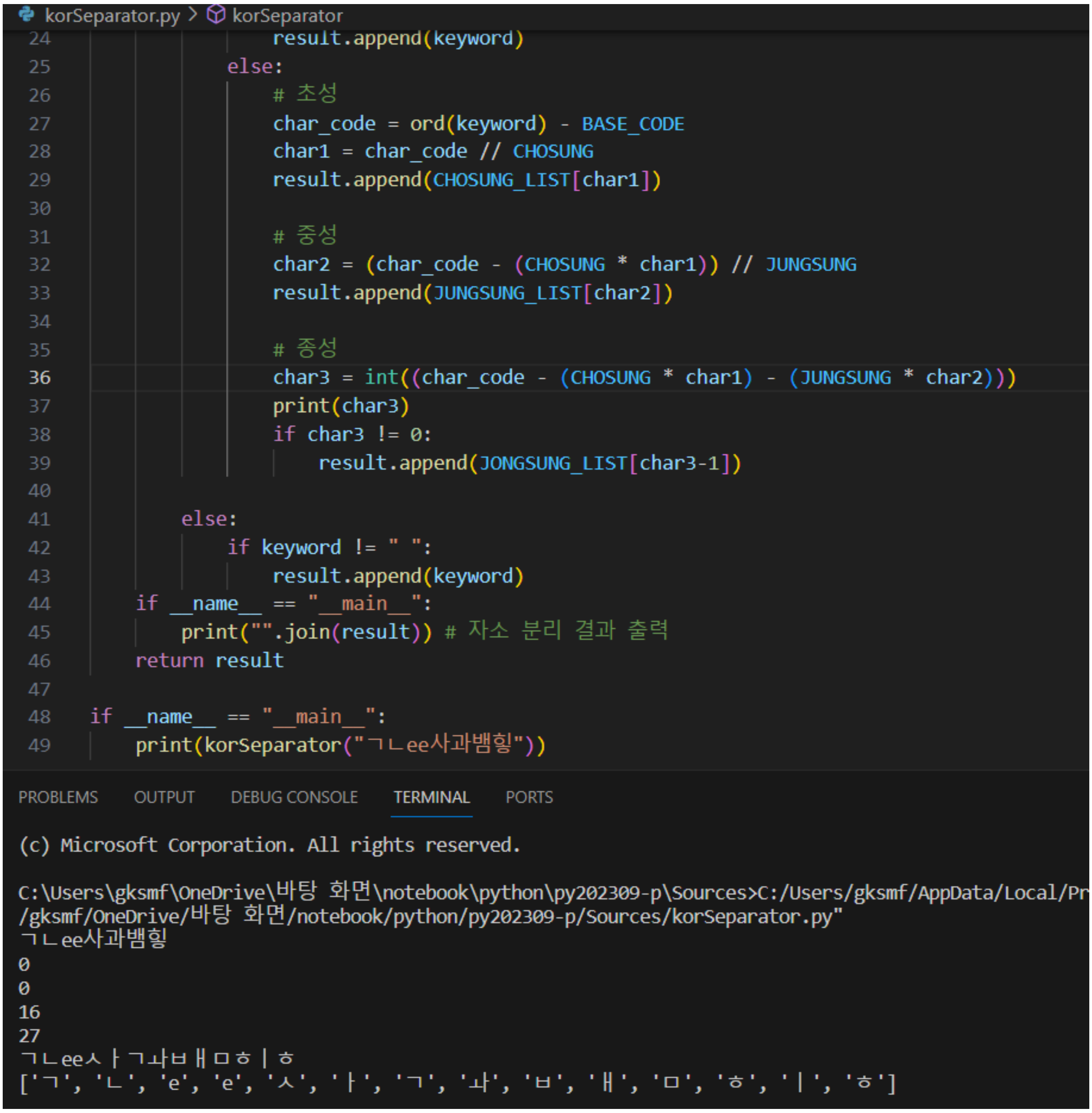
```python
else:
    wordSpells = korSeparator.korSeparator(word)
    wordSpellsLenght = len(wordSpells)
    korDictFinal = []
    korWordSpells = []
    sortedwordSpells = sorted(wordSpells)
    finalResult = []
    for i in korDict.korDict:
        i = i.rstrip("\n")
        if len(i)==wordLen:
            korWordSpell = korSeparator.korSeparator(i)
            if len(korWordSpell)==wordSpellsLenght:
                korDictFinal.append(i)
                korWordSpells.append(korWordSpell)
    for i in range(len(korWordSpells)):
        if sortedwordSpells == sorted(korWordSpells[i]):
            finalResult.append(korDictFinal[i])
    if word in finalResult:
        finalResult.remove(word)
    if finalResult:
        for i in range(len(finalResult)):
            print(f"{i+1}. {finalResult[i]}")
    else:
        print("국어 사전에 해당하는 단어가 없습니다.")
```

2) Test result

(1) Function to separate consonants and vowels in Korean

- explanation: There are 588 Korean Unicode for each consonant. Therefore, the initial consonant can be obtained by subtracting 44032 (first letter) and dividing it by 588. In addition, vowels and consonants can be obtained by calculating the number of 588 identical initial vowels.

- Test Results Screenshot



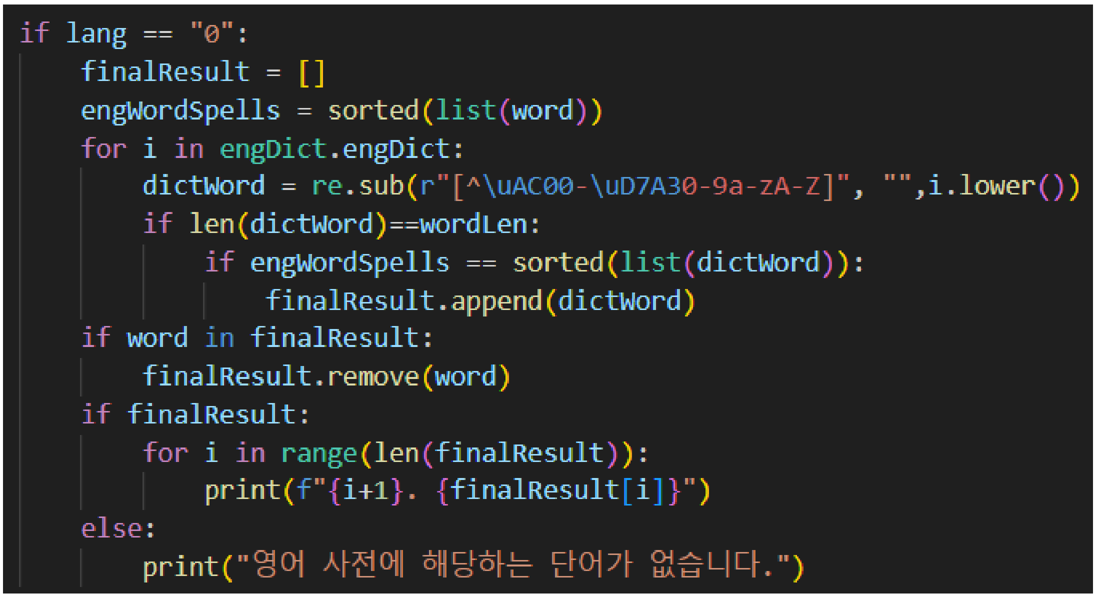
```
                result.append(keyword)
            else:
                # 초성
                char_code = ord(keyword) - BASE_CODE
                char1 = char_code // CHOSUNG
                result.append(CHOSUNG_LIST[char1])

                # 중성
                char2 = (char_code - (CHOSUNG * char1)) // JUNGSUNG
                result.append(JUNGSUNG_LIST[char2])

                # 종성
                char3 = int((char_code - (CHOSUNG * char1) - (JUNGSUNG * char2)))
                print(char3)
                if char3 != 0:
                    result.append(JONGSUNG_LIST[char3-1])

        else:
            if keyword != " ":
                result.append(keyword)
    if __name__ == "__main__":
        print("".join(result)) # 자소 분리 결과 출력
    return result

if __name__ == "__main__":
    print(korSeparator("ㄱㄴee사과뱀힣"))
```

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\gksmf\OneDrive\바탕 화면\notebook\python\py202309-p\Sources>C:/Users/gksmf/AppData/Local/Pr
/gksmf/OneDrive/바탕 화면/notebook/python/py202309-p/Sources/korSeparator.py"
ㄱㄴee사과뱀힣
0
0
16
27
ㄱㄴeeㅅㅏㄱㅘㅂㅐㅁㅎㅣㅎ
['ㄱ', 'ㄴ', 'e', 'e', 'ㅅ', 'ㅏ', 'ㄱ', 'ㅘ', 'ㅂ', 'ㅐ', 'ㅁ', 'ㅎ', 'ㅣ', 'ㅎ']
```

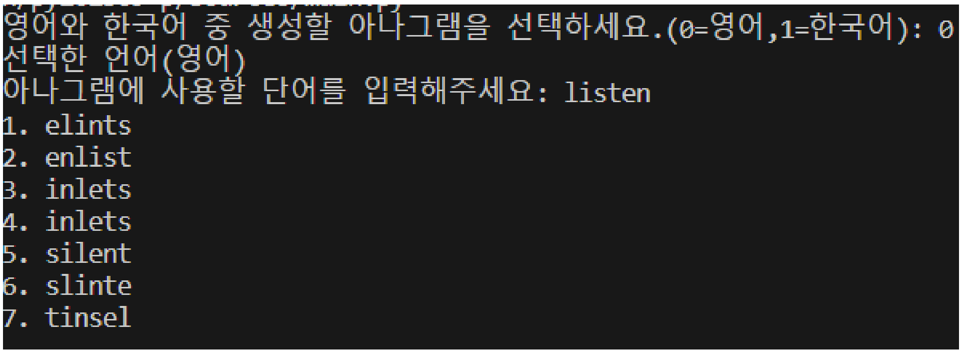
(2) Words with the same length are changed into a list form, sorted, and compared with the sorted list of words input by the user.

- explanation: Words with the same length are changed into a list form, sorted, and compared with the sorted list of words input by the user.(In Korean, consonants and vowels are separated and compared)

- Test Results Screenshot(eng)

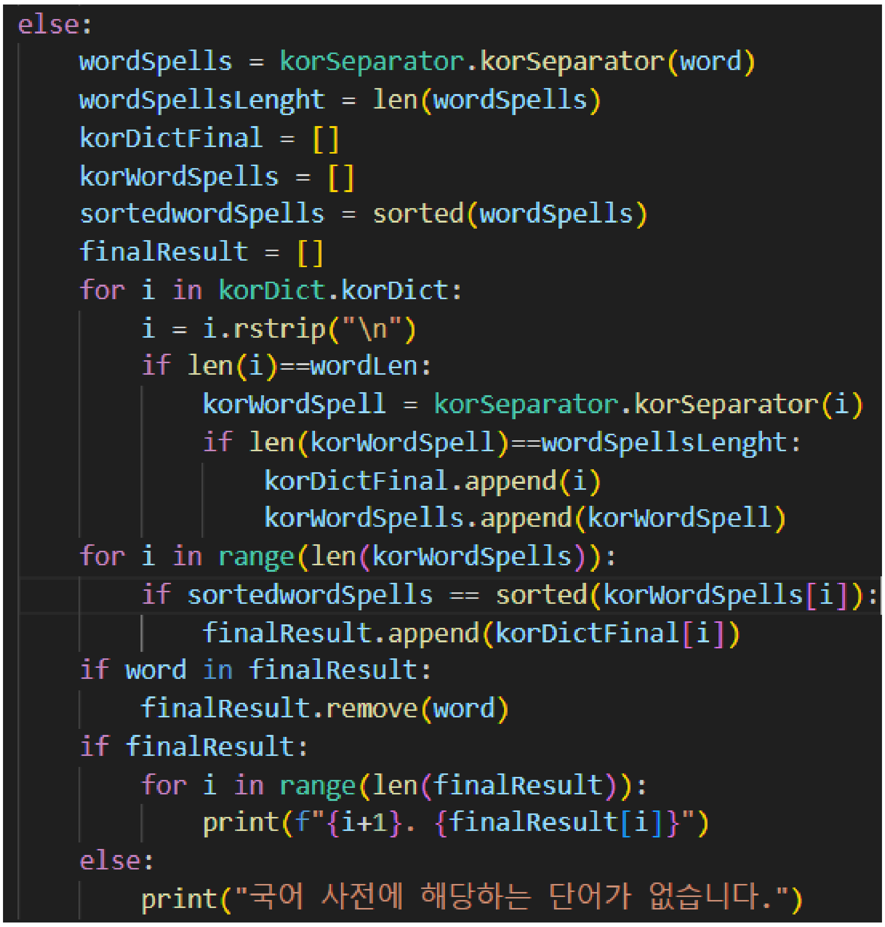


```
if lang == "0":
    finalResult = []
    engWordSpells = sorted(list(word))
    for i in engDict.engDict:
        dictWord = re.sub(r"[^\uAC00-\uD7A30-9a-zA-Z]", "",i.lower())
        if len(dictWord)==wordLen:
            if engWordSpells == sorted(list(dictWord)):
                finalResult.append(dictWord)
    if word in finalResult:
        finalResult.remove(word)
    if finalResult:
        for i in range(len(finalResult)):
            print(f"{i+1}. {finalResult[i]}")
    else:
        print("영어 사전에 해당하는 단어가 없습니다.")
```

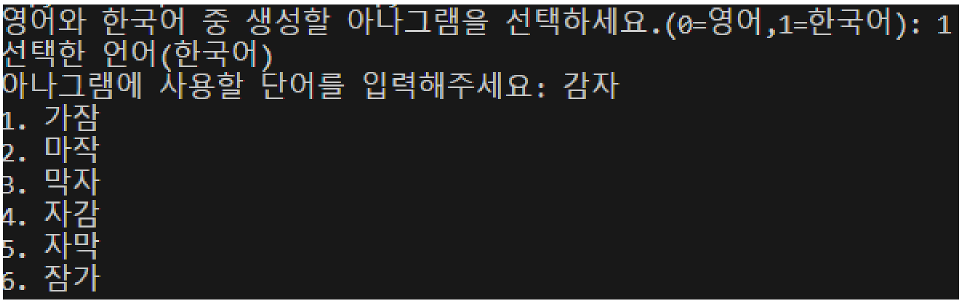


```
영어와 한국어 중 생성할 아나그램을 선택하세요.(0=영어,1=한국어): 0
선택한 언어(영어)
아나그램에 사용할 단어를 입력해주세요: listen
1. elints
2. enlist
3. inlets
4. inlets
5. silent
6. slinte
7. tinsel
```

- Test Results Screenshot(kor)



```
else:
    wordSpells = korSeparator.korSeparator(word)
    wordSpellsLenght = len(wordSpells)
    korDictFinal = []
    korWordSpells = []
    sortedwordSpells = sorted(wordSpells)
    finalResult = []
    for i in korDict.korDict:
        i = i.rstrip("\n")
        if len(i)==wordLen:
            korWordSpell = korSeparator.korSeparator(i)
            if len(korWordSpell)==wordSpellsLenght:
                korDictFinal.append(i)
                korWordSpells.append(korWordSpell)
    for i in range(len(korWordSpells)):
        if sortedwordSpells == sorted(korWordSpells[i]):
            finalResult.append(korDictFinal[i])
    if word in finalResult:
        finalResult.remove(word)
    if finalResult:
        for i in range(len(finalResult)):
            print(f"{i+1}. {finalResult[i]}")
    else:
        print("국어 사전에 해당하는 단어가 없습니다.")
```



```
영어와 한국어 중 생성할 아나그램을 선택하세요.(0=영어,1=한국어): 1
선택한 언어(한국어)
아나그램에 사용할 단어를 입력해주세요: 감자
1. 가잠
2. 마작
3. 막자
4. 자감
5. 자막
6. 잠가
```

4. Changes in Comparison to the Plan

1) Change the word search algorithm

-Before

Create a word permutation and compare it with a word dictionary.

-After

Compare the number of letters without creating a permutation.

-Reason

Overwhelmingly shorter return time

5. Schedule

- 진행 상황 표기

업무		11/3	11/10	11/17	11/23	12/10	12/17
제안서 작성		complete					
1) Function to choose between English and Korean	세부기능1		complete				
2) Function to receive and save words from users	세부기능 1,2			complete			
3) Save words in a word dictionary	세부기능1			complete			
4) The ability to output corresponding words from a word dictionary	세부기능1					complete	
	세부기능2					complete	