

## ✓ AI기반 챗봇 및 OCR 개발 전문가 과정

교과목명 : m2\_programming

- 평가일 : 24.05.14
- 성명 :
- 점수 :

※ 문제당 2점으로 채점 60점 + 자유 프로그램 40점

### ✓ 1. numbers 리스트에서 15, 20, 25를 출력하세요.(for문과 if 조건 사용 필수)

```
numbers = [5, 15, 6, 20, 7, 25]
```

```
numbers= [5, 15, 6, 20, 7, 25]
print([i for i in numbers if i >= 15])
```

```
↻ [15, 20, 25]
```

### ✓ 2. 아래 list\_input에 대해서 주어진 사용자 함수를 일괄적으로 적용할 수 있는 2개의 내장함수를 활용하여 [1, 4, 9, 16, 25], [1, 2]를 출력하는 코드를 작성하세요.

```
# 사용자 함수를 선언합니다.
def power(item):
    return item * item
def under_3(item):
    return item < 3
```

```
# 변수를 선언합니다.
list_input = [1, 2, 3, 4, 5]
print([power(i) for i in list_input])
print([i for i in list_input if under_3(i)])
```

```
↻ [1, 4, 9, 16, 25]
   [1, 2]
```

### ✓ 3. 정수를 입력하면 짝수 홀수 여부를 반환해주는 프로그램을 작성하세요.(정수를 입력하지 않는 경우에 대해서 예외처리 수행)

```
while True:
    try:
        num=int(input("정수를 입력하세요: "))
        if num % 2 == 0:
            print("짝수입니다.\n")
        else:
            print("홀수입니다.\n")
        break
    except ValueError:
        print("올바른 값(정수)을 입력해주세요\n")
```

```
↻ 정수를 입력하세요: 2
   짝수입니다.

   정수를 입력하세요: asdfasdf
   올바른 값(정수)을 입력해주세요

   정수를 입력하세요: 1
   홀수입니다.
```

### ✓ 4. a의 b제곱에서 a x b를 뺀 수를 a, b가 아래와 같이 주어졌을 때 람다 함수를 이용해서 구하세요.

```
a = 2, b = 5
```

```
a= 5, b = 2
```

```
myEquation = lambda x, y: x**y-x*y
print(myEquation(2,5))
print(myEquation(5,2))
```

 22  
15


- ✓ 5. 정수를 입력하면 천 단위로 콤마로 구분하여 출력하세요.(정수가 아닌 경우 '정수가 아닙니다.'를 출력)

예시

정수를 입력하세요: 123456789

123,456,789

```
while True:
    try:
        num=int(input("정수를 입력하세요: "))
        print(f"{num:,}\n")
        #print(format(num, ','))
        break
    except ValueError:
        print("정수가 아닙니다.\n")
```

 정수를 입력하세요: adfasdf  
정수가 아닙니다.

정수를 입력하세요: 1.2  
정수가 아닙니다.

정수를 입력하세요: 123456789  
123,456,789

- ✓ 6. `example_list = ["요소A", "요소B", "요소C"]`를 파이썬 내장함수를 이용하여 아래와 같이 출력하세요.


0번째 요소는 요소A입니다.

1번째 요소는 요소B입니다.

2번째 요소는 요소C입니다.

```
example_list = ["요소A", "요소B", "요소C"]
```

```
[print(f"{index}번째 요소는 {value}입니다.") for index, value in enumerate(example_list)]
```

 0번째 요소는 요소A입니다.  
1번째 요소는 요소B입니다.  
2번째 요소는 요소C입니다.  
[None, None, None]

- ✓ 7. 97을 40으로 나눈 몫과 나머지를 내장함수를 이용하여 구하세요.(튜플로 출력 필수)

```
myEquation = lambda x, y: (x//y, x%y)
print(myEquation(97,40))
```

 (2, 17)

8. `list_number = [52, 273, 32, 72, 100]`로 변수 선언하고 정수를 입력하면 그 정수 인덱스와 리스트에서 해당하는 값을 출력하는 프로그램을 작성하세요. 단, 모든 예외처리를 수행하며 특히 `ValueError`와 `IndexError`는 별도 구분해서 예외처리 한 후 '예외 처리 완료'를 마지막으로 출력하세요.

- 4 입력시  
정수 입력> 4  
4번째 요소: 100 예외 처리 완료
- 문자 입력시  
정수 입력> 가  
정수를 입력해 주세요! 예외 처리 완료
- 5 입력시  
정수 입력> 5  
리스트의 인덱스를 벗어났어요! 예외 처리 완료

```
list_number = [52, 273, 32, 72, 100]
while True:
    try:
        input_num=int(input("정수입력 > "))
        index=input_num
        value=list_number[index]
        print(f"{index}번째 요소:{value}",end=" ")
        break
    except IndexError as e:
        #print(f'An error occurred: {e}')
        print(f'리스트의 인덱스를 벗어났어요!',end=" ")
    except ValueError as e:
        print(f'정수를 입력해주세요!',end=" ")
    finally:
        print("\n예외 처리완료\n")
```

↻ 정수입력 > asdfadsf  
정수를 입력해주세요! 예외 처리완료

정수입력 > 1.3  
정수를 입력해주세요! 예외 처리완료

정수입력 > 10  
리스트의 인덱스를 벗어났어요! 예외 처리완료

정수입력 > 4  
4번째 요소:100 예외 처리완료

✓ 9. 사용자로부터 서로 다른 숫자 두 개를 입력받고 첫 번째 와 두 번째중 큰 숫자를 구하는 프로그램을 수행하세요.

```
while True:
    a=input("숫자를 입력해주세요: ")
    b=input("숫자를 입력해주세요: ")
    if a > b: print("첫 번째 숫자가 더 큼니다.\n"); break
    elif b > a: print("두 번째 숫자가 더 큼니다.\n"); break
    else: print("두 수가 동일합니다. 다시 입력해주세요\n")
```

↻ 숫자를 입력해주세요: 1  
숫자를 입력해주세요: 1  
두 수가 동일합니다. 다시 입력해주세요

숫자를 입력해주세요: 2  
숫자를 입력해주세요: 5  
두 번째 숫자가 더 큼니다.

✓ 10. 1부터 100까지 아래와 같은 방식으로 2개의 수를 곱해서 가장 큰 수를 구하세요.

1 \* 99 ... 99 \* 1

```
maximum=0
for i in range(1,100):
    curNum=i*(100-i)
    if curNum > maximum: maximum=curNum
print(maximum)
```

↻ 2500

✓ 11. 현재 시간을 년-월-일 시:분:초로 출력하세요.

```
import datetime
import pytz
tz=pytz.timezone('Asia/Seoul');
now=datetime.datetime.now(tz)
now_time1 = now.strftime('%Y-%m-%d %H:%M:%S')
now_time2 = now.strftime('%Y년 %m월 %d일 %H시:%M분:%S초')
print(now_time2)
print(now_time1)
```

↻ 2024년 05월 14일 15시:15분:55초  
2024-05-14 15:15:55

✓ 12. 1에서 백만까지 더하는데 걸리는 프로그램 실행 시간을 밀리초(ms) 단위로 구하세요.(1000 밀리초(ms) = 1초)

```
import time
start_t=time.time()

sum1=0
for i in range(1,1000001):
    sum1 += i

end_t=time.time()
print(f"총 합계: {sum1}, 소요시간: {(end_t-start_t)*1000} mS")
```

➦ 총 합계: 500000500000, 소요시간: 138.9625072479248 mS

✓ 13. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c']에서 9와 'a'를 삭제 후 리스트로 출력하세요

```
List=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c']
List.remove(9)
List.remove('a')
print(List)
```

➦ [0, 1, 2, 3, 4, 5, 6, 7, 8, 'b', 'c']

✓ 14. 500 ~ 1000사이의 정수 중 5와 7의 공배수의 합을 구하세요

```
result=[]
for i in range(500,1001):
    if i %5 ==0 and i%7 == 0:
        result.append(i)
```

```
print(sum(result))
```

➦ 10535

✓ 15. []을 채워서 아래와 같이 출력하세요.

[[1, 4, 7], [2, 5, 8], [3, 6, 9]]

```
numbers = [1,2,3,4,5,6,7,8,9]
output = [[ ], [ ], [ ]]

for number in numbers:
    output[(number+2)%3].append(number)
print(output)
```

➦ [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

✓ 16. numbers 내부에 들어 있는 숫자가 몇 번 등장하는지를 출력하는 코드를 작성하세요.

numbers = [3,6,3,5,8,2,9,7,4,3,6,3,5,8,2,9,7,4] 답안 예시:  
{3: 4, 6: 2, 5: 2, 8: 2, 2: 2, 9: 2, 7: 2, 4: 2}

```
numbers = [3,6,3,5,8,2,9,7,4,3,6,3,5,8,2,9,7,4]
dict={}
for i in numbers:
    if i in dict:
        dict[i] += 1
    else: dict[i]=1
print(dict)
```

➦ {3: 4, 6: 2, 5: 2, 8: 2, 2: 2, 9: 2, 7: 2, 4: 2}

✓ 17. 클래스 Country를 완성하여 아래와 같이 출력하세요.

```
class Country:
    def __init__(self, name, pop, cap):
        self.name = name
        self.pop = pop
        self.cap = cap

    def desc(self):
        print(f"{self.name}의 인구는 {self.pop}천만명이며, 수도는 {self.cap}입니다.")
```

```
class Korea(Country):

    def desc1(self, gnp):
        self.gnp = gnp
        print('1인당 gnp는 USD %d만입니다.' % self.gnp)
```

```
obj = Korea('한국', 5, '서울')
obj.desc()
obj.desc1(3)
```

↩ 한국의 인구는 5천만명이며, 수도는 서울입니다.  
1인당 gnp는 USD 3만입니다.

- ✓ 18. 사칙연산을 수행하는 클래스를 작성하여 객체를 만들고 2개의 정수 10, 5를 입력한 후 연산 결과를 포맷을 이용하여 출력하세요.

```
class FundArith:
    a=int(input('사칙연산을 위한 첫번째 수 입력:> '))
    b=int(input('사칙연산을 위한 두번째 수 입력:> '))
    def Cal(self,operator):
        if self.a == 0:
            self.a=int(input('올바른 수를 입력하세요 (첫번째 수):> '))
        if self.b == 0:
            self.b=int(input('올바른 수를 입력하세요 (첫번째 수):> '))
        arith_eq="self.a"+operator+"self.b"
        print("{} {} {} = {}".format(self.a,operator,self.b,eval(arith_eq)))
```

```
obj=FundArith()
obj.Cal("+")
obj.Cal("-")
obj.Cal("*")
obj.Cal("/")
```

↩ 사칙연산을 위한 첫번째 수 입력:> 10  
사칙연산을 위한 두번째 수 입력:> 5  
10 + 5 = 15  
10 - 5 = 5  
10 \* 5 = 50  
10 / 5 = 2.0

- ✓ 19. 올해 경과된 날짜수를 계산하여 출력하세요.

답안 예시:

오늘은 2024-01-01 이후 127일째 되는 날입니다.

```
from datetime import datetime

DateTime=datetime.now()
#print(DateTime,type(DateTime))
TimeOut=DateTime.strftime('%Y-%m-%d %H:%M:%S')
PassDay=DateTime.strftime('%j')

print(f"오늘은 2024-01-01이후 {PassDay}일째 되는 날입니다.")
```

↩ 오늘은 2024-01-01이후 135일째 되는 날입니다.

- ✓ 20. 문자열 리스트 states를 아래 [출력]와 같이 전처리하여 출력하세요

```
states = [' Alabama ', 'Georgia!', 'Georgia', 'georgia', 'Fl0rIda', 'south carolina##', 'West virginia?']
```

[출력]  
['Alabama',  
'Georgia',  
'Georgia',  
'Georgia',

```
'Florida',
'South Carolina',
'West Virginia']

import re

states = [' Alabama ', 'Georgia!', 'Georgia', 'georgia', 'FlOrIda', 'south carolina##', 'West virginia?']
result=[]
for word in states:
    word=re.sub(r'[?.,!##\nW'::]', '', word).title()
    result.append(word)

print(result)

[' Alabama ', 'Georgia', 'Georgia', 'Georgia', 'Florida', 'South Carolina', 'West Virginia']
```

자유 프로그램 수행 결과를 실행하세요

```
### 1. 시작 인터페이스
### PachinkoByMin-Jin-Lee
### 청천벽력ByShogo-Kawashima
### TwilightByStephenieMeyer
### 편지ByNatsume-Soseki
### TheNameOfTheRoseByUmber toEco

# ## 외부 라이브러리인 googletrans 설치 여부 확인
# !pip list | grep googletrans
# ## 외부 라이브러리 googletrans가 install되어 있지 않으면, Package-Installer-of-python로부터 아래 구글 번역 라이브러리 install
#!pip install googletrans==4.0.0rc1
# ## install된 구글 번역 라이브러리 정보 double-Check
# !pip show googletrans

# ## 외부 라이브러리인 Google Text to Speech API 설치 여부 확인
# !pip list | grep gTTS
# ## 외부 라이브러리 googletrans가 install되어 있지 않으면, Package-Installer-of-python로부터 아래 구글 번역 라이브러리 install
#!pip install gTTS
# ## install된 구글 번역 라이브러리 정보 double-Check
# !pip show gTTS

# ## 외부 라이브러리인 Play Sound 설치 여부 확인
# !pip list | grep playsound
# ## 외부 라이브러리 Play Sound가 되어 있지 않으면, Package-Installer-of-python로부터 아래 구글 번역 라이브러리 install
#!pip install playsound
# ## install된 Play Sound 라이브러리 정보 double-Check
# !pip show playsound

deletingEntry=['the', 'when', 'two', 'twenty-eight', 'something', 'beside', 'want', 'eldest', 'of', 'near', 'next', 'mother', 'both', 'hands', 'eyes',
'1911', 'legs', 'too', 'this', 'his', 'help', 'but', 'from', 'couldnt', 'wrong', 'girls', 'after', 'with', 'sons', 'washed', 'walk', 'lost',
'at', 'just', 'that', 'her', 'having', 'old', 'middle', 'read', 'same', 'cold', 'one', 'did', 'foot:', 'korea', 'long', 'moved', 'kitch
'well', 'late', 'who', 'run', 'sleeping', 'an', 'korean', 'hoonies', 'aging', 'make', 'over', 'yeongdo', 'as', 'three', 'golden', 'made'
'lived', 'on', 'koreahistory', 'work', 'daughter', 'larger', 'full', 'being', 'would', 'mouth', 'him', 'not', 'were', 'she', 'lodgerst
'increase', 'quicklyboth', 'along', 'couple', 'girl', 'no', 'worked', 'whod', 'than', 'him', 'let', 'he', 'habit', 'outdoor', 'keep', 'c
'needed', 'talk', 'strong', 'ricein', 'evening', 'japanese', 'how', 'their', 'to', 'could', 'know', 'eventually', 'paper', 'in', 'hot', '
'countries', 'yet', 'survived', 'turn', 'man', 'avoided', 'its', 'benefit', 'wallat', 'seen', 'hundred', 'known', 'finished', 'called',
'sums', 'wooden', 'pointless', 'resembled', 'twice', 'number', 'inky', 'peasants', 'things', 'guessed', 'quickly', 'has', 'rent', 'gave'
'snug', 'low', 'child', 'fathom', 'vegetable', 'decided', 'rivals', 'learned', 'someone', 'dining', 'say', 'busan', 'asked', 'steady', '
'thinking', 'because', 'carry', 'out', 'gone', 'birth', 'so', 'tables', 'what', 'i', 'and', 'raised', 'growing', 'stay', 'day', 'morning
'elderly', 'years', 'hoonie', 'fisherman', 'us', 'himself', 'slept', 'take', 'do', 'by', 'they', 'possible', 'nature', 'it', 'five-mile
'any', 'doors', 'yeongdo-a', 'only', 'hardy', 'was', 'five', 'youngest', 'kitchen', 'bed', 'such', 'or', 'large', 'parents', 'lodgers', 'hec

import googletrans
from gtts import gTTS
#import playsound as ps
import os
import re
import pickle
import random
curpath='/content/drive/MyDrive/00_KITA_2404/M2_Program'
os.chdir(curpath)

str1="Welcome to Our Translation/Voiceenabled Program"
str2="WtWt용도 및 목적: 외국어 원문 Reading 학습을 위한 번역 및 Listening 학습을 위한 원어 음성 파일 생성 "
str3="WtWt기능:"
str4="WtWt(i) 사용자 입력 TEXT (1. sample-TEXTs, 2. File-Input, 3. Text-Input Window)로 부터 원하는 언어로 번역"
str5="WtWt(ii) 번역된 언어를 음성 파일(MP3)로 변환하여 Listening 반복 학습 기능 제공"
str6="WtWt(iii) 사용자 입력 외국어 원문으로부터 빈출 단어를 추출하여 사용자에게 제공하여 학습 효율 개선"
langs=['한국어', '영어', '일본어', '중국어-간체', '중국어-번체', '러시아어', '프랑스어', '독일어', '이탈리아어', '스페인', '베트남어', '태국', '인도네시아어', '체코어
langCodes=['ko', 'en', 'ja', 'zh-CN', 'zh-TW', 'ru', 'fr', 'de', 'it', 'es', 'vi', 'th', 'id', 'cs', 'da', 'pl']
```

```
langcodes = ['ko', 'en', 'ja', 'zh-cn', 'zh-tw', 'ru', 'it', 'de', 'fr', 'es', 'vi', 'th', 'tr', 'cs', 'da', 'et']
```

```
class TransSpeech:

    def InitDisp(self):
        print("#"*200,end='');
        for i in range(2): print("\n##"+ " "*195,"##",end='')
        print(f'\n##{str1:^195} ##',end='')
        for i in range(2): print("\n##"+ " "*195,"##",end='')
        print(f'\n##{str2:<152} ##',end='')
        for i in range(2): print("\n##"+ " "*195,"##",end='')
        print(f'\n##{str3:<181} ##',end='')
        print(f'\n##{str4:<167} ##',end='')
        print(f'\n##{str5:<160} ##',end='')
        print(f'\n##{str6:<145} ##',end='')
        for i in range(2): print("\n##"+ " "*195,"##",end='')
        i=1
        for lang in langs:
            if i%8==1: print("\n## ",end="")
            print(f'{i:<2}:Wt{lang:<15}',end="")
            if i%8==0: print(" "*49+"##",end="")
            i += 1
        for i in range(4): print("\n##"+ " "*195,"##",end='')
        print('\n'+ "#"*200);

    def ProvidedLangs(self):
        i=1: print("\n"+ "#"*20,"지원 언어")
        for lang in langs:
            print(f'{i:>2}:Wt{lang:<15}',end="")
            if i%8==0: print();
            i += 1

    def TextLoad(self):
        TEXT_DB={1: '/content/drive/MyDrive/00_KITA_2404/M2_Program/TEXTs/Pachinko_ByMinJinLee.txt', 2: '/content/drive/MyDrive/00_KITA_2404/M2_Program/TEXTs/
        print("\n"+ "#"*60)
        print("1. 기본 제공 Sample:(영문소설)파친코\n")
        print("2. 기본 제공 Sample:(일문소설)청천벽력\n")
        print("3. 기본 제공 Sample:(영문소설)Twilight\n")
        print("4. Loading Text File(Entered The File-Path:ex:/content/drive/MyDrive/00_KITA_2404/M2_Program/TEXTs/Letter_ByNatsumeSoseki.txt)\n")
        print("5. Directly Type the TEXT string in the following input window\n")
        selectedInput = int(input(f"***** TEXT를 불러올 방법을 선택하세요 (1~5) >>> "))
        if 1 <= selectedInput <= 3:
            with open(TEXT_DB[selectedInput], 'r') as f:
                textString = f.read()
        elif selectedInput == 4:
            textFile = input(f"***** 읽어들이 TEXT 파일의 path를 입력하세요 >>> ")
            with open(textFile, 'r') as f:
                textString = f.read()
        elif selectedInput == 5:
            textString = input(f"***** 읽어들이 문자열을 직접 입력하세요 >>> ")
        return textString

    def TransString(self, textString):
        transObj = googletrans.Translator()
        self.ProvidedLangs()
        inText = int(input(f"\n***** 입력된 문자열의 언어를 선택하세요 (1-16 선택, 자동으로 인식하고자 할 경우 17) >>> "))
        if 1 <= inText <= 16:
            inLang=langCodes[inText-1]
        elif inText == 17:
            inLang='auto'
        else: print("\n**** 올바른 입력 언어를 선택하세요!")
        outText = int(input(f"\n***** 출력하고자 하는 언어를 선택하세요 (1-16 선택) >>> "))
        if 1 <= outText <= 16:
            outLang=langCodes[outText-1]
        else: print("\n**** 올바른 출력 언어를 선택하세요!")
        translatedString = transObj.translate(textString, dest=outLang, src=inLang)
        transOut=translatedString.text
        detectedInLan=translatedString.src
        inLang=detectedInLan
        print("\n"*2,transOut)
        saveTextYN = input(f"\n***** 번역된 TEXT를 저장할까요 (Y/N)? >>> ").lower()
        if saveTextYN == "y":
            saveText = input(f"\n***** 저장할 파일의 경로와 파일명을 입력하세요(/content/drive/MyDrive/00_KITA_2404/M2_Program/translatedText.txt) >>> ")
            with open(saveText,'wb') as f:
                pickle.dump(transOut,f)
            print(f"생성된 번역 TEXT가 {saveText}에 저장되었습니다.")
            #print(data)
        return textString, transOut, inLang, outLang

    def TextToSpeech(self, textString, inLang):
        tts = gTTS(text=textString, lang=inLang)
        saveAudioYN = input(f"\n***** 생성된 오디오 파일을 저장할까요 (Y/N)? >>> ").lower()
```

```

if saveAudioYN == "y":
    saveAudio = input(f"Wn***** 저장할 오디오 파일의 경로/파일명을 입력하세요 (/content/drive/MyDrive/00_KITA_2404/M2_Program/translatedAudio.mp3) >>> ")
    tts.save(saveAudio)
    print(f"생성된 오디오 파일이 {saveAudio}에 저장되었습니다.")
#ps.playsound(outMP3)

def VocaBook(self, textString, inLang, outLang):
    words_list=re.sub(r'[.,!#WnW'::]', ' ', textString).lower()
    words_list = words_list.split()
    words_set=set(words_list); words_db={}
    for W in words_set:
        words_db[W]=words_list.count(W)
    words_db2=sorted(words_db.items(), key=lambda x: -x[1])    ### .item으로 부르니까 애는 list안에 tuple형태로 out된다.
    i=1
    nomVocas = int(input(f"Wn***** 생성할 단어 개수를 입력하세요 >>> "))
    saveVocas = input(f"Wn***** 저장할 단어장의 경로/파일명을 입력하세요 (/content/drive/MyDrive/00_KITA_2404/M2_Program/VocaSummary.txt) >>> ")
    transObj2 = googletrans.Translator()
    print("WnWn", "#"*58, "Wn", "#"*24, "단 어 장", "#"*24, "Wn", "#"*58)
    f_voca = open(saveVocas, 'w')
    vocaDB={}
    for ckey in words_db2:
        if ckey[0] not in deletingEntry:
            if i <= nomVocas:
                translatedVoca=transObj2.translate(ckey[0], dest=outLang, src=inLang).text
                f_voca.write(ckey[0]+"Wt"*2+translatedVoca+"Wn")
                print(f" {ckey[0]:<10}WtWt{translatedVoca:<10}")
                vocaDB[ckey[0]]=translatedVoca
            i += 1
    f_voca.close()
    return vocaDB

def Quiz(self, vocaDB):
    listKeys=list(vocaDB.keys())
    correct_no=0; wrong_no=0;
    i=1
    for voca in vocaDB:
        listKeys.remove(voca)
        sampling4=random.sample(listKeys,3)
        sampling4.append(voca); random.shuffle(sampling4)
        answer=sampling4.index(voca)+1
        print("Wn ", "#"*25, "Wn ", "#"*7, f"Quiz({i})번", "#"*7, "Wn ", "#"*25, "Wn ", "="*80)
        reply=int(input(f" {voca}: (1) {vocaDB[sampling4[0]]} (2) {vocaDB[sampling4[1]]} (3) {vocaDB[sampling4[2]]} (4) {vocaDB[sampling4[3]]}"))
        print(" ", "="*80)
        listKeys.append(voca)
        if reply == answer:
            print(f" 정답입니다.Wn"); correct_no += 1
        else: print(f" 오답입니다.Wn"); wrong_no += 1
        i += 1
    print(f" !!!! 수고하셨습니다. {correct_no+wrong_no}문항 중 {correct_no}문항 맞추셨습니다 !!!!!!!!!.Wn")

obj1=TransSpeech()
obj1.InitDisp()
textString=obj1.TextLoad()
textString, transOut, inLang, outLang=obj1.TransString(textString)
genAudioYN = input(f"Wn***** 음성 파일을 생성할까요 (Y/N)? >>> ").lower()
if genAudioYN == "y":
    obj1.TextToSpeech(textString, inLang)
vocaDB=obj1.VocaBook(textString, inLang, outLang)
obj1.Quiz(vocaDB)

```





desire	욕구
tiles	타일
sensible	현명한
multitudes	다수
annexed	합병
forbearance	인내
pillaged	약탈
shoulders	어깨
wine	와인
shy	수줍은
mortared	모르타르

```
#####
##### Quiz(1)번 #####
#####
```

```
=====
matchmaker:  (1) 결혼 중매인  (2) 학교  (3) 시장  (4) 어깨 >> 1
=====
```

정답입니다.

```
#####
##### Quiz(2)번 #####
#####
```

```
=====
enough:  (1) 욕구  (2) 충분한  (3) 학교  (4) 시장 >> 2
=====
```

정답입니다.

```
#####
##### Quiz(3)번 #####
#####
```

```
=====
speak:  (1) 수줍은  (2) 말하다  (3) 시장  (4) 가족 >> 2
=====
```

```
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive