

HomeWork2

- 기간 : 11월 15일부터 21일까지
- 파이썬 파일(.py)파일을 자신의 학번_hw2_1.py형식으로 저장하여 담당TA메일로 제출하기
- 각 프로그램 문장에 comment를 넣기 (프로그램 내에도 자신의 이름_학번을 기술함)
 - Comment는 첫줄에 #자신의 학번, 이름을 적으시고
 - 각 문장당 주석처리를 해주세요.

과제 1 : 문자표현(Data Representation)

- 제시된 ASCII코드값을 이용하여 숫자값을 문자값으로, 문자값은 숫자값으로 전환하는 프로그램을 작성하기.
- 함수 4개 구현

함수명	인자	반환값
def character(decoding_table,code):	decoding_table(사전) : 키는 코드번호, 값은 코드번호에 해당하는 한 글자 code(정수, 코드번호) : ASCII코드에 해당하는 코드(32-127)만 있다고 가정함	code 에 해당하는 글자 하나를 포함하는 문자열을 반환함
def ordinal(decoding_table,char):	decoding_table(사전): 키는 코드번호 , 값은 코드번호에 해당하는 한 글자 code(정수, 코드번호) : ASCII코드에 해당하는 코드(32-127)만 있다고 가정함	char에 해당하는 코드값의 정수를 반환함
def decode(decoding_table,codes):	decoding_table(사전) : 키는 코드번호, 값은 코드번호에 해당하는 한 글자 Codes(정수로 이루어진 리스트): ASCII코드에 해당하는 코드(32-127)만으로 이루어진 리스트	codes에 포함된 정수에 해당하는 글자들을 포함하는 문자열
def encode(decoding_table,string):	decoding_table(사전) : 키는 코드번호, 값은 코드번호에 해당하는 한 글자 Codes(정수로 이루어진 리스트): ASCII코드에 해당하는 코드(32-127)문자만 포함하는 문자열	String에 해당하는 각 글자의 코드(정수)를 포함하는 리스트

```

if __name__ == '__main__':
    ascii_table={
        32:' ', 33:'!', 34:'"', 35:'#',
        36:'$', 37:'%', 38:'&', 39:'"',
        40:'(', 41:')', 42:'*', 43:'+',
        44:',', 45:'-', 46:'.', 47:'/',
        48:'0', 49:'1', 50:'2', 51:'3',
        52:'4', 53:'5', 54:'6', 55:'7',
        56:'8', 57:'9', 58:':', 59:';',
        60:'<', 61:'=', 62:'>', 63:'?',
        64:'@', 65:'A', 66:'B', 67:'C',
        68:'D', 69:'E', 70:'F', 71:'G',
        72:'H', 73:'I', 74:'J', 75:'K',
        76:'L', 77:'M', 78:'N', 79:'O',
        80:'P', 81:'Q', 82:'R', 83:'S',
        84:'T', 85:'U', 86:'V', 87:'W',
        88:'X', 89:'Y', 90:'Z', 91:'[',
        92:'\\', 93:']', 94:'^', 95:'_',
        96:'`', 97:'a', 98:'b', 99:'c',
        100:'d', 101:'e', 102:'f', 103:'g',
        104:'h', 105:'i', 106:'j', 107:'k',
        108:'l', 109:'m', 110:'n', 111:'o',
        112:'p', 113:'q', 114:'r', 115:'s',
        116:'t', 117:'u', 118:'v', 119:'w',
        120:'x', 121:'y', 122:'z', 123:'{',
        124:'|', 125:'}', 126:'~'}

```

ASCII CODE 설정부분

호출부분

```

print(character(ascii_table,65))
print(ordinal(ascii_table,'G'))
print(decode(ascii_table,[72,101,108,108,111,44,32, 80,121,
                        116,104,111,110,33]))
print(encode(ascii_table,"Have a nice day ~ :"))

```

결과

```

A
71
Hello, Python!
[72, 97, 118, 101, 32, 97, 32, 110, 105, 99, 101, 32, 100, 97, 121, 32, 126, 32, 58, 41]

```

과제 2: 동아리 멤버

- 동아리 정보가 들어있는 사전형 자료member를 이용하여 학생이 이름이 인자로 주어지면, 해당 학생이 속한 동아리 이름을 집합으로 반환하며, 동아리 이름을 전달하면 해당 동아리들에 속한 학생들을 리스트로 반환하는 함수를 작성하기
- 함수 2가지

함수명	인자	반환값
<pre>def find_circles(data, student_name):</pre> <p>동아리 정보가 담긴 사전과 학생이름을 인자로 받아 학생이 속한 모든 동아리를 찾아 집합으로 이름을 반환하기</p>	<p>data: 사전으로부터 읽어 들인 동아리 정보</p> <p>student_name: 문자열로 조회하고자 하는 학생이름</p>	<p>- 동아리의 이름을 담은 집합</p> <p>- 존재하지 않으면 없음의 문자열 반환</p>
<pre>def find_members(data, *members):</pre> <p>동아리 이름을 전달하면, 해당 동아리들에 속한 학생들을 찾아 리스트로 반환하기</p>	<p>data: 사전을 통한 동아리 정보</p> <p>members: 가변형태의 tuple로 조회하고자 하는 동아리 이름들이 문자열로 저장되어 있는 튜플</p>	<p>- 학생들의 이름을 담은 리스트</p> <p>- 존재하지 않으면 No result</p>

과제 2

호출

```
if __name__ == '__main__':  
    members = {'Music club': ['Bob', 'Alice', 'Dan', 'Heidi', 'Victor'],  
               'Football club': ['Mallory', 'Victor', 'Bob', 'Dan', 'Frank', 'Peggy'],  
               'Tennis club': ['Alice', 'Bob', 'Carol', 'Victor', 'Frank']}  
    print(find_circles(members, 'Alice'))  
    print(find_circles(members, 'Bob'))  
    print(find_circles(members, 'Dennins'))  
    print(find_circles(members, 'Peggy'))  
    print(find_members(members, 'Football club', 'Tennis club'))  
    print(find_members(members, 'Music club', 'Football club', 'Tennis club'))
```

결과

```
{'Tennis club', 'Music club'}  
{'Football club', 'Tennis club', 'Music club'}  
There is no data for Dennins.  
{'Football club'}  
['Frank', 'Victor', 'Bob']  
['Victor', 'Bob']
```