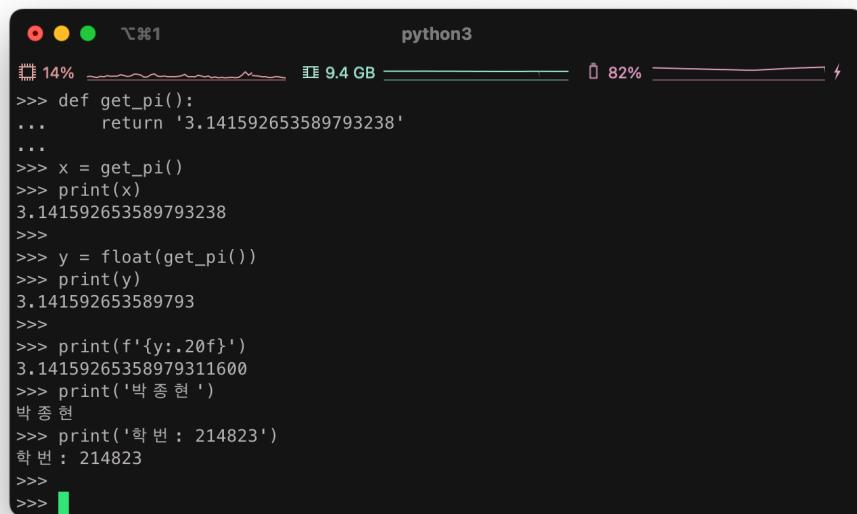


Homework 04

이름: 박종현

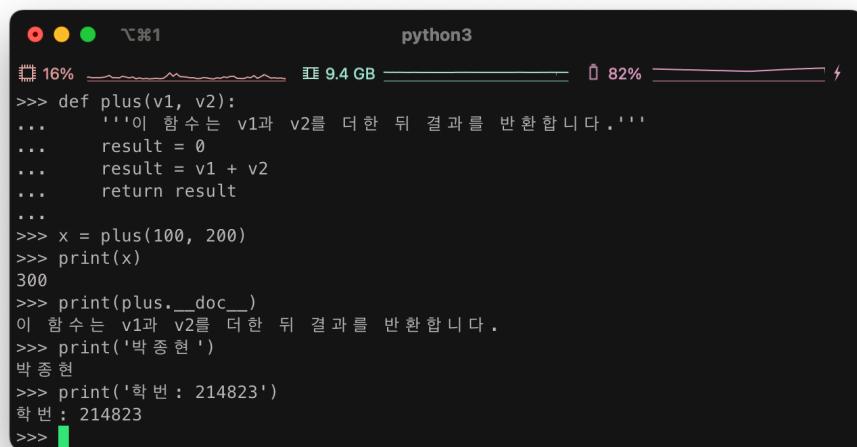
학번: 214823

[1] 7-1. 16~17 쪽



```
python3
14% 9.4 GB 82%
>>> def get_pi():
...     return '3.141592653589793238'
...
>>> x = get_pi()
>>> print(x)
3.141592653589793238
>>>
>>> y = float(get_pi())
>>> print(y)
3.141592653589793
>>>
>>> print(f'{y:.20f}')
3.14159265358979311600
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

[2] 7-1. 21~22 쪽



```
python3
16% 9.4 GB 82%
>>> def plus(v1, v2):
...     '''이 함수는 v1과 v2를 더한 뒤 결과를 반환합니다.'''
...     result = 0
...     result = v1 + v2
...     return result
...
>>> x = plus(100, 200)
>>> print(x)
300
>>> print(plus.__doc__)
이 함수는 v1과 v2를 더한 뒤 결과를 반환합니다.
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

[3] 7-2. 9~10 쪽

```
python3
15% 9.4 GB 83%
>>> def para_func(v1, v2, v3=0):
...     result = 0
...     result = v1 + v2 + v3
...     return result
...
>>> hap = 0
>>>
>>> hap = para_func(10, 20)
>>> print('매개 변수 2개 함수 호출 결과 ==> ', hap)
매개 변수 2개 함수 호출 결과 ==> 30
>>> hap = para_func(10, 20, 30)
>>> print('매개 변수 3개 함수 호출 결과 ==> ', hap)
매개 변수 3개 함수 호출 결과 ==> 60
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

```
python3
14% 9.5 GB 83%
>>> def para3_func(v1, v2, v3):
...     return v1 + v2 + v3
...
>>> print(para3_func(10, 20, 30))
60
>>>
>>> def para3_func(v1, v2=0, v3):
    File "<stdin>", line 1
        def para3_func(v1, v2=0, v3):
        ^
SyntaxError: parameter without a default follows parameter with a default
>>>     return v1 + v2 + v3
    File "<stdin>", line 1
        return v1 + v2 + v3
IndentationError: unexpected indent
>>>
>>> def para3_func(v1, v2, v3=0):
...     return v1 + v2 + v3
...
>>> print(para3_func(10, 20))
30
>>>
>>> def para3_func(v1, v2=0, v3=0):
...     return v1 + v2 + v3
...
>>> print(para3_func(10))
10
>>> print(para3_func(10, 20, 30))
60
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

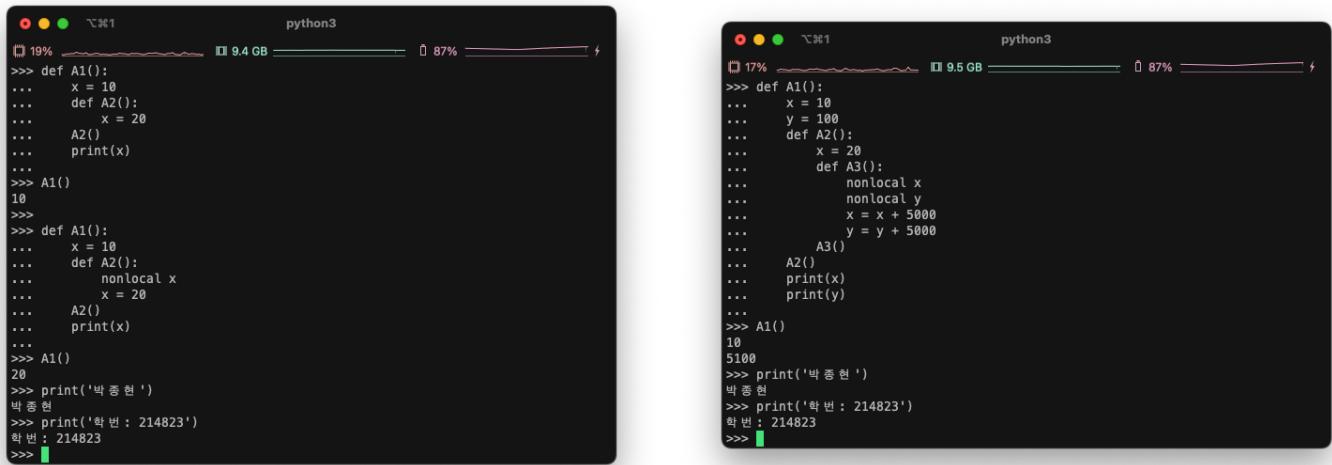
[4] 7-2. 16~17 쪽

```
python3
>>> def func3():
...     result1 = 100
...     result2 = 200
...     return result1, result2
...
>>> hap1, hap2 = 0, 0
>>>
>>> hap1, hap2 = func3()
>>> print('func3()에서 돌려준 값 ==> ', hap1, hap2)
func3()에서 돌려준 값 ==> 100 200
>>>
>>> hap_tup = func3()
>>> print('func3()에서 돌려준 값 ==>', hap_tup)
func3()에서 돌려준 값 ==> (100, 200)
>>> print('func3()에서 돌려준 값 ==>', hap_tup[0], hap_tup[1])
func3()에서 돌려준 값 ==> 100 200
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

[5] 7-2. 25,27~30 쪽

```
python3
>>> def func1():
...     a = 10
...     print('func1()에서 a의 값 ', a)
...
>>> def func2():
...     print('func2()에서 a의 값 ', a)
...
>>> a = 20
>>>
>>> func1()
func1()에서 a의 값 10
>>> func2()
func2()에서 a의 값 20
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

```
python3
>>> def func1():
...     global a
...     a = 10
...     print('func1()에서 a의 값 ', a)
...
>>> def func2():
...     print('func2()에서 a의 값 ', a)
...
>>> a = 20
>>>
>>> func1()
func1()에서 a의 값 10
>>> func2()
func2()에서 a의 값 10
>>>
>>> del(a)
>>> func2()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "<stdin>", line 2, in func2
NameError: name 'a' is not defined
>>>
>>> func1()
func1()에서 a의 값 10
>>> func2()
func2()에서 a의 값 10
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```



```

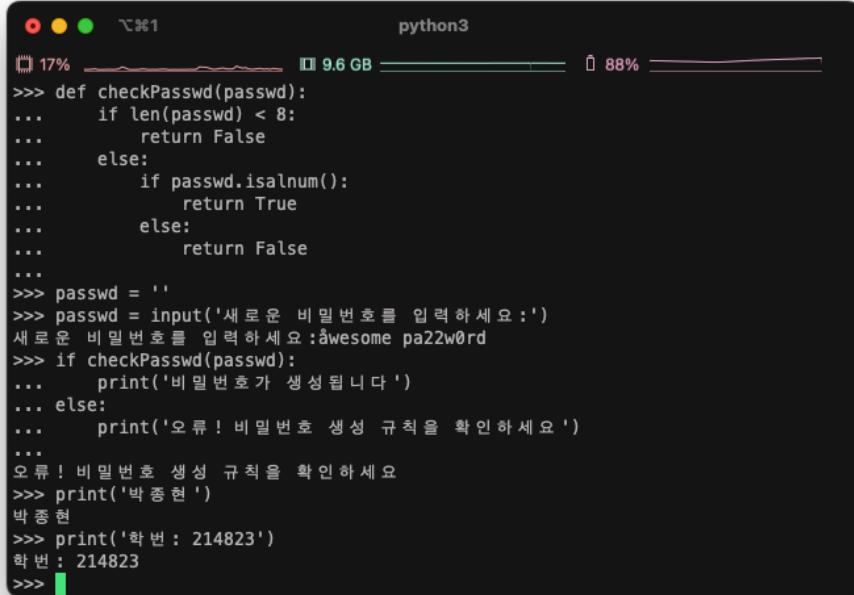
python3
19% 9.4 GB 87%
>>> def A1():
...     x = 10
...     def A2():
...         x = 20
...     A2()
...     print(x)
...
>>> A1()
10
>>> def A1():
...     x = 10
...     def A2():
...         nonlocal x
...         x = 20
...     A2()
...     print(x)
...
>>> A1()
20
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>> 
```



```

python3
17% 9.5 GB 87%
>>> def A1():
...     x = 10
...     y = 100
...     def A2():
...         x = 20
...         def A3():
...             nonlocal x
...             nonlocal y
...             x = x + 5000
...             y = y + 5000
...         A3()
...     A2()
...     print(x)
...     print(y)
...
>>> A1()
10
5100
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>> 
```

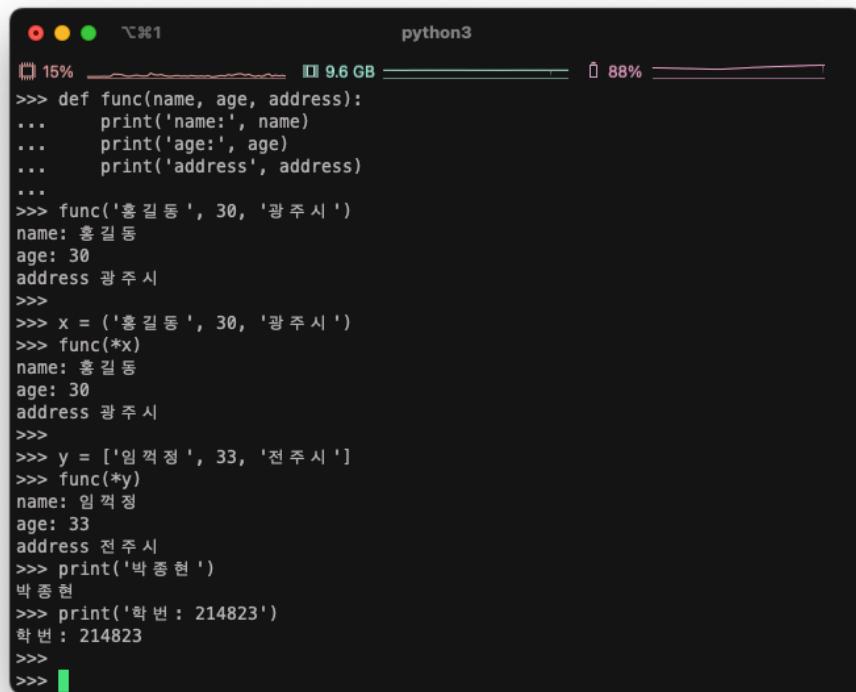
[6] 7-2. 31~32 쪽



```

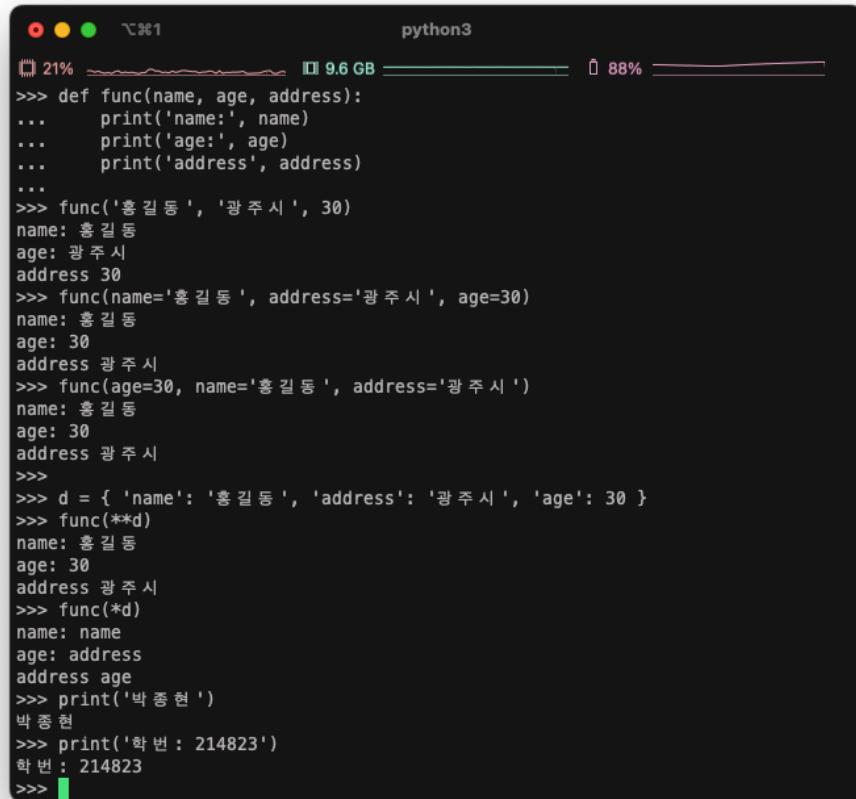
python3
17% 9.6 GB 88%
>>> def checkPasswd(passwd):
...     if len(passwd) < 8:
...         return False
...     else:
...         if passwd.isalnum():
...             return True
...         else:
...             return False
...
>>> passwd = ''
>>> passwd = input('새로운 비밀번호를 입력하세요 : ')
새로운 비밀번호를 입력하세요 : awesome pa22w0rd
>>> if checkPasswd(passwd):
...     print('비밀번호가 생성됩니다.')
... else:
...     print('오류! 비밀번호 생성 규칙을 확인하세요')
...
오류! 비밀번호 생성 규칙을 확인하세요
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>> 
```

[7] 7-3. 6쪽



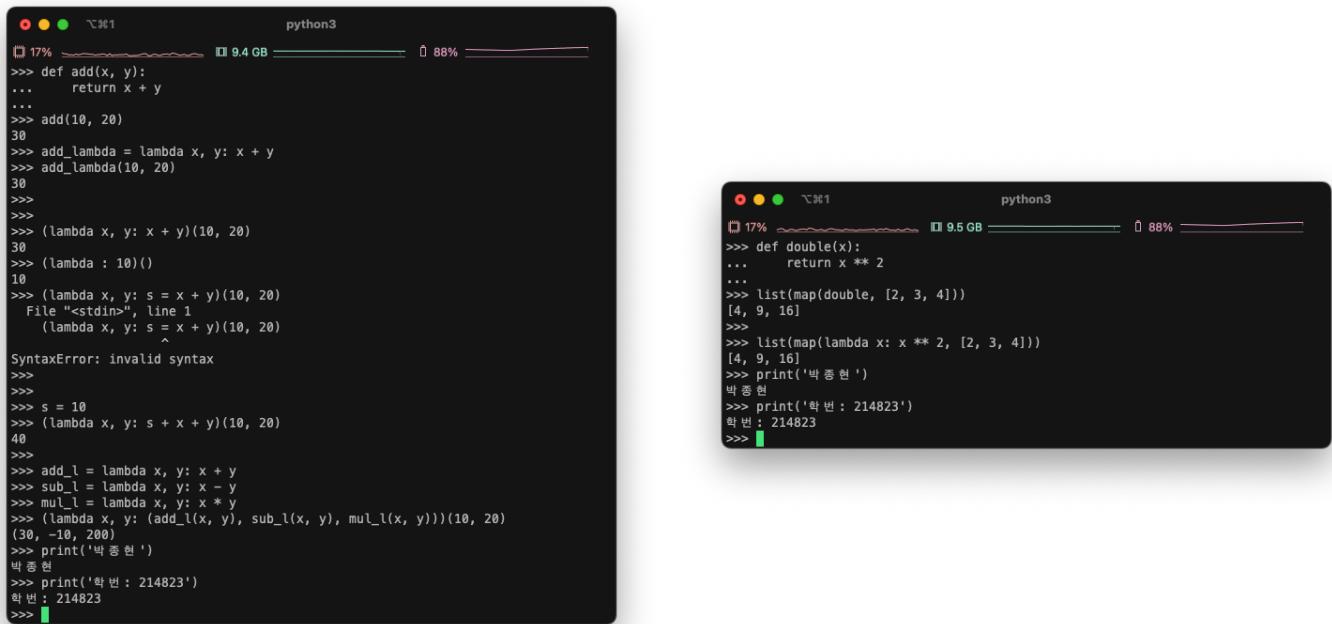
```
python3
15% 9.6 GB 88%
>>> def func(name, age, address):
...     print('name:', name)
...     print('age:', age)
...     print('address', address)
...
>>> func('홍길동', 30, '광주시')
name: 홍길동
age: 30
address 광주시
>>>
>>> x = ('홍길동', 30, '광주시')
>>> func(*x)
name: 홍길동
age: 30
address 광주시
>>>
>>> y = ['임꺽정', 33, '전주시']
>>> func(*y)
name: 임꺽정
age: 33
address 전주시
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
>>>
```

[8] 7-3. 12~13쪽



```
python3
21% 9.6 GB 88%
>>> def func(name, age, address):
...     print('name:', name)
...     print('age:', age)
...     print('address', address)
...
>>> func('홍길동', '광주시', 30)
name: 홍길동
age: 광주시
address 30
>>> func(name='홍길동', address='광주시', age=30)
name: 홍길동
age: 30
address 광주시
>>> func(age=30, name='홍길동', address='광주시')
name: 홍길동
age: 30
address 광주시
>>>
>>> d = { 'name': '홍길동', 'address': '광주시', 'age': 30 }
>>> func(**d)
name: 홍길동
age: 30
address 광주시
>>> func(*d)
name: name
age: address
address age
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

[9] 7-3. 19~22 쪽



The image shows two side-by-side terminal windows, both titled "python3".

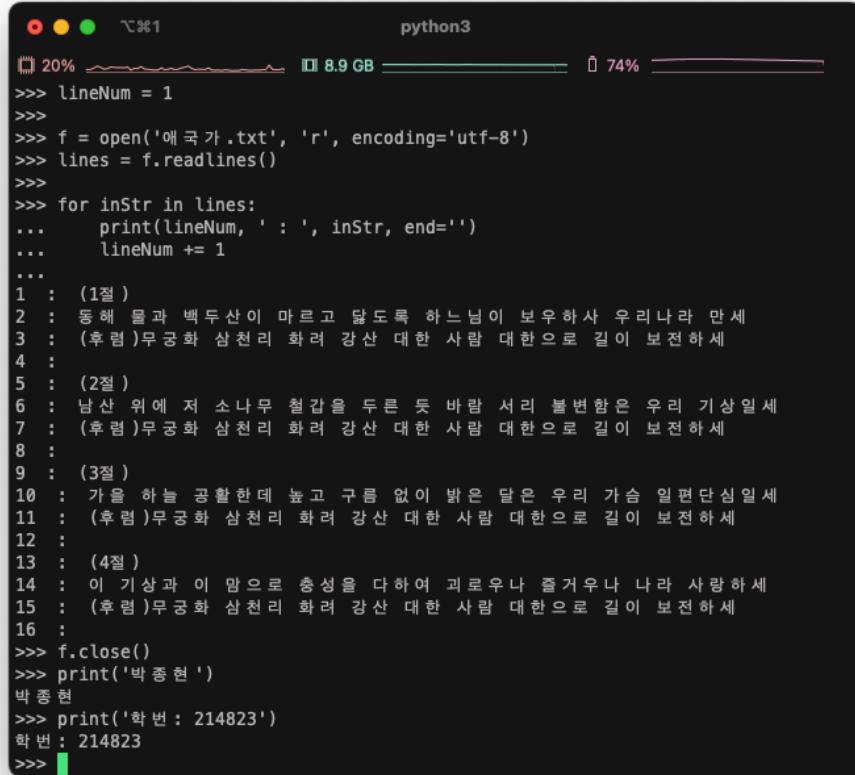
The left terminal window has a progress bar at 17% and a memory usage of 9.4 GB. It contains the following code:

```
>>> def add(x, y):
...     return x + y
...
>>> add(10, 20)
30
>>> add_lambda = lambda x, y: x + y
>>> add_lambda(10, 20)
30
>>>
>>> (lambda x, y: x + y)(10, 20)
30
>>> (lambda x: 10)()
10
>>> (lambda x, y: s = x + y)(10, 20)
File "<stdin>", line 1
  (lambda x, y: s = x + y)(10, 20)
          ^
SyntaxError: invalid syntax
>>>
>>>
>>> s = 10
>>> (lambda x, y: s + x + y)(10, 20)
40
>>>
>>> add_l = lambda x, y: x + y
>>> sub_l = lambda x, y: x - y
>>> mul_l = lambda x, y: x * y
>>> (lambda x, y: (add_l(x, y), sub_l(x, y), mul_l(x, y)))(10, 20)
(30, -10, 200)
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

The right terminal window has a progress bar at 17% and a memory usage of 9.5 GB. It contains the following code:

```
>>> def double(x):
...     return x ** 2
...
>>> list(map(double, [2, 3, 4]))
[4, 9, 16]
>>>
>>> list(map(lambda x: x ** 2, [2, 3, 4]))
[4, 9, 16]
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

[10] 9-1. 19 쪽



The terminal window is titled "python3" and has a progress bar at 20% and a memory usage of 8.9 GB. It contains the following code:

```
>>> lineNum = 1
>>>
>>> f = open('애국가.txt', 'r', encoding='utf-8')
>>> lines = f.readlines()
>>>
>>> for inStr in lines:
...     print(lineNum, ' : ', inStr, end='')
...     lineNum += 1
...
1 : (1절)
2 : 동해 물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세
3 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
4 :
5 : (2절)
6 : 남산 위에 저 소나무 철갑을 두른 듯 바람 서리 불변함은 우리 기상일세
7 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
8 :
9 : (3절)
10 : 가을 하늘 공활한데 높고 구름 없이 밝은 달은 우리 가슴 일편단심일세
11 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
12 :
13 : (4절)
14 : 이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라 사랑하세
15 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
16 :
>>> f.close()
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

[11] 9-1. 21 쪽

```
python3
>>> lineNumber = 1
>>>
>>> with open('애국가.txt', 'r', encoding='utf-8') as f:
...     lines = f.readlines()
...
>>> for inStr in lines:
...     print(lineNumber, ' : ', inStr, end='')
...     lineNumber += 1
...
1 : (1절 )
2 : 동 해 물과 백두산이 마르고 달도록 하느님이 보우하사 우리나라 만세
3 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
4 :
5 : (2절 )
6 : 남산 위에 저 소나무 철갑을 두른 듯 바람 서리 불변함은 우리 기상일세
7 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
8 :
9 : (3절 )
10 : 가을 하늘 공활한데 높고 구름 없이 밝은 달은 우리 가슴 일편단심일세
11 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
12 :
13 : (4절 )
14 : 이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라 사랑하세
15 : (후렴)무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세
16 :
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

[12] 9-1. 25~27 쪽

```
python3
16% 9.4 GB 88%
>>> with open('애국가.txt', 'rb') as f:
...     f.read(5)
...     f.readline()
...     f.readline(10)
...
b'(1\xec\xa0\x88'
b')\n'
b'\xeb\x8f\x99\xed\x95\xb4 \xeb\xac\xbc'
>>> with open('애국가.txt', 'r', encoding='utf-8') as f:
...     f.read(5)
...     f.readline()
...     f.readline(10)
...
'(1월)\n'
'동해 물과 백두산이 마르고 달도록 하느님이 보우하사 우리나라 만세 \n'
'(후렴)부궁 홍심천'
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>> 
```

```
python3
21% 9.5 GB 88%
>>> with open('에국가.txt', 'r', encoding='utf-8') as f:
...     lines = f.readlines()
...
>>> with open('에국가.txt', 'rb') as f:
...     lines_b = f.readlines()
...
>>> lines[0]
'(1월 )\n'
>>> lines_b[0]
b'(1xec\x a@x88)\n'
>>>
>>> lines[1]
'동해 물과 백두산이 마르고 달도 뜰 하느님이 보우하사 우리나라 만세\n'
>>> lines_b[1]
b'\xeab\x8f\x99\xed\x95\xb4 \xeab\xac\xbc\xea\xab\xb3\xbc \xeab\xb0\xb1\xeb\x91\x90\xec
\xb2\xb0\xec\x9d\xb4 \xeab\xa7\x88\xeb\xab\x5\xb4\xab\xb3\xab \xeab\xb1\x8f\x
84\xeb\xab\x1\x9d \xeab\x95\x98\xeb\x8a\x90\xeb\xb8\x98\xec\x9d\xb4 \xeab\xb3\xb4\xec
\x9a\xb0\xed\x95\x98\xec\x82\xac \xeab\x9a\xb0\xeb\xab\xac\xeb\x82\x98\xeb\x9d\xb
\xeb\xab\x7\x8c\xec\x84\xb8\xn'
>>> print('ㅂ종현')
ㅂ종현
>>> print('학번: 214823')
학번: 214823
>>> 
```

```
python3
Help on built-in function encode:
encode(encoding='utf-8', errors='strict') method of builtins.str instance
  Encode the string using the codec registered for encoding.

  encoding
    The encoding in which to encode the string.
  errors
    The error handling scheme to use for encoding errors.
    The default is 'strict' meaning that encoding errors raise a
    UnicodeEncodeError. Other possible values are 'ignore', 'replace' and
    'xmlcharrefreplace' as well as any other name registered with
    codecs.register_error that can handle UnicodeEncodeErrors.

(END)
```

```
python3
Help on built-in function decode:
decode(encoding='utf-8', errors='strict') method of builtins.bytes instance
  Decode the bytes using the codec registered for encoding.

  encoding
    The encoding with which to decode the bytes.
  errors
    The error handling scheme to use for the handling of decoding errors.
    The default is 'strict' meaning that decoding errors raise a
    UnicodeDecodeError. Other possible values are 'ignore' and 'replace'
    as well as any other name registered with codecs.register_error that
    can handle UnicodeDecodeErrors.

(END)
```

```
python3
17% python3
>>> lines[0].encode()
b'(xec\xad\x88)\n'
>>> lines_b[0].decode()
'(_ENCODING)\n'
>>>
>>> help(lines[0].encode)

>>> help(lines_b[0].decode)

>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

```
shapeler@ShapeLayer-MacBook:~/Desktop/ai-basics/task04
python 12.py
~/Desktop/ai-basics/task04
```

[13] 9-2. 10~12 쪽

```
python3
14% python3
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.write('오늘의 일정 :')
...     f.write('내일의 일정 :')
...
7
7
>>>
```

```
python3
16% python3
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.write('오늘의 일정 :')
...     f.write('내일의 일정 :')
...
8
8
>>>
```

```
python3
17% python3
>>> strList = ['어제:', '오늘:', '내일:']
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.writelines(strList)
...
>>>
```

```
python3
18% python3
>>> strList = ['어제:\n', '오늘:\n', '내일:\n']
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.writelines(strList)
...
>>>
```

```
python3
14% 9.6 GB 88%
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.write('오늘의 일정:')
...     f.write('내일의 일정:')

...
7
7
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.write('오늘의 일정:' + '\n')
...     f.write('내일의 일정:' + '\n')
...
8
8
>>> strList = ['어제:', '오늘:', '내일:']
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.writelines(strList)
...
...
>>> strList = ['어제:\n', '오늘:\n', '내일:\n']
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     f.writelines(strList)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

```
python3
31% 9.6 GB 87%
>>> with open('일정표.txt', 'a', encoding='utf-8') as f:
...     while True:
...         str = input('일정을 입력하세요 : ')
...         if str != '':
...             f.writelines(str + '\n')
...         else:
...             break
...
일정을 입력하세요 : 오늘은 공부
일정을 입력하세요 : 내일은 여행
일정을 입력하세요 : 모레는 시험
일정을 입력하세요 :
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

[14] 9-2. 15~19 쪽

```
python3
13% 9.4 GB 87%
>>> with open('에국가.txt', 'r', encoding='utf-8') as fi:
...     with open('에국가-1.txt', 'w') as fo:
...         for line in fi:
...             fo.write(line)
...
5
37
37
1
5
40
37
1
5
39
37
1
5
38
37
1
1
>>> with open('에국가.txt', 'r', encoding='utf-8') as fi:
...     with open('에국가-1.txt', 'w') as fo:
...         lines = fi.readlines()
...         fo.writelines(lines)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

```
python3
18% 9.4 GB 86%
python (Python) 9.4 GB ..basics/task04 (-zsh)
>>> with open('에국가.txt', 'r', encoding='utf-8') as fi:
...     with open('에국가-1.txt', 'w') as fo:
...         for line in fi:
...             fo.write(line)
...
5
37
37
1
5
40
37
1
5
39
37
1
1
>>> with open('에국가.txt', 'r', encoding='utf-8') as fi:
...     with open('에국가-1.txt', 'w') as fo:
...         lines = fi.readlines()
...         fo.writelines(lines)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

(1절)
동해 물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세
(후령)무궁화 삼천리 화려 강산 대한 사람대한으로 길이 보전 하세

(2절)
남산 위에 저 소나무 철갑을 두른 듯 바람서리 블번함은 우리 기상 일세
(후령)무궁화 삼천리 화려 강산 대한 사람대한으로 길이 보전 하세

(3절)
이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라 사랑하세
(후령)무궁화 삼천리 화려 강산 대한 사람대한으로 길이 보전 하세

```
python3
21% 9.5 GB 86%
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     while True:
...         str = input('일정을 입력하세요 : ')
...         if str != '':
...             f.writelines(str + '\n')
...         else:
...             break
...
일정을 입력하세요 : 오늘은 공부
일정을 입력하세요 : 내일은 여행
일정을 입력하세요 :
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

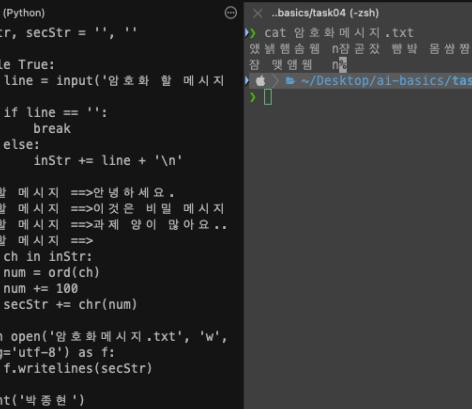
```
python3
17% 9.5 GB 86%
python (Python) 9.5 GB ..basics/task04 (-zsh)
>>> with open('일정표.txt', 'w', encoding='utf-8') as f:
...     while True:
...         str = input('일정을 입력하세요 : ')
...         if str != '':
...             f.writelines(str + '\n')
...         else:
...             break
...
일정을 입력하세요 : 오늘은 공부
일정을 입력하세요 : 내일은 여행
일정을 입력하세요 :
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

cat 일정표.txt
오늘은 공부
내일은 여행

```
python3
17% 9.6 GB 86%
>>> ord('남')
45212
>>> chr(45212)
'남'
>>>
>>> num = ord('남')
>>> chr(num + 100)
'њ'
>>>
>>> num = ord('њ')
>>> chr(num - 100)
'남'
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

```
 19% 9.6 GB 85%
>>> inStr, secStr = '', ''
>>>
>>> while True:
...     line = input('암호화 할 메시지 ==>')
...     if line == '':
...         break
...     else:
...         inStr += line + '\n'

암호화 할 메시지 ==>안녕하세요.
암호화 할 메시지 ==>이것은 비밀 메시지
암호화 할 메시지 ==>과제 영이 많아요..
암호화 할 메시지 ==>
>>> for ch in inStr:
...     num = ord(ch)
...     num += 100
...     secStr += chr(num)
...
>>> with open('암호화메시지.txt', 'w', encoding='utf-8') as f:
...     f.writelines(secStr)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>> 
```



```
python (Python)
>>> inStr, secStr = '', ''
>>>
>>> while True:
...     line = input('암호화 할 메시지 ')
...     if line == '':
...         break
...     else:
...         inStr += line + '\n'
...
암호화 할 메시지 ==>안녕하세요 .
암호화 할 메시지 ==>이것은 비밀 메시지
암호화 할 메시지 ==>과제 암이 많아요 ..
암호화 할 메시지 ==>
>>> for ch in inStr:
...     num = ord(ch)
...     num += 100
...     secStr += chr(num)
...
>>> with open('암호화 메시지.txt', 'w',
... encoding='utf-8') as f:
...     f.writelines(secStr)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

```
python3
16% 9.5 GB 85%
>>> secStr, orgStr = '', ''
>>>
>>> with open('암호화메시지.txt', 'r', encoding='utf-8') as f:
...     secStr = f.read()
...     for ch in secStr:
...         num = ord(ch)
...         num -= 100
...         orgStr += chr(num)
...     print('### 메시지 복호화 ###')
...     print(orgStr)
...
### 메시지 복호화 ###
안녕하세요.
이것은 비밀 메시지
과제 암호화입니다.
```

[15] 9-2. 21~24 쪽

```
 15%  ``Ctrl1  9:59 AM  9.9 GB  0  85%  
>>> def insertGrade(outfile):  
...     ...  
...     한 번, 이름, 성적을 입력 받아, 주어진 outfile에 쓰고 저장하는 함수  
...     ...  
...     while True:  
...         print('학번, 이름, 성적을 입력하세요.')  
...         sid = input('학번 : ')  
...         sname = input('이름 : ')  
...         sgrade = input('성적 : ')  
...         if sid != '':  
...             outStr = sid + ':' + sname + ':' + sgrade + '\n'  
...             outfile.write(outStr)  
...         else:  
...             break  
...  
>>> def calgradeAvg(infile, outfile):  
...     ...  
...     infile의 데이터를 읽고, 학생수와 성적평균을 구해서 outfile에 쓰는 함수  
...     ...  
...     numStudent = 0  
...     gradeSum = 0  
...     for line in infile:  
...         numStudent += 1  
...         data = line.split(':')  
...         gradeSum += float(data[2])  
...     outStr = ['학생수 : ', str(numStudent), '\n']  
...     outStr += ['평균성적 : ', str(gradeSum/numStudent)]  
...     outfile.writelines(outStr)  
...  
>>> print('파일명?')  
파일명? insertGrade('학번.txt')  
학번 : 214623  
>>> 
```

```
 16% █ 9.5 GB 84%
>>> with open('성적표.txt', 'a', encoding='utf-8') as f:
...     insertGradeDef()
...
학번, 이름, 성적을 입력하세요
학번 : 214823
이름 : 빅_종현
성적 : 4.5
학번, 이름, 성적을 입력하세요
학번 : 200123
이름 : 흥_일동
성적 : 3.5
학번, 이름, 성적을 입력하세요
학번 : 390231
이름 : 흥_민경
성적 : 3.1
학번, 이름, 성적을 입력하세요
학번 : 180154
이름 : 전_주지
성적 : 4.3
학번, 이름, 성적을 입력하세요
학번 :
이름 :
성적 :
>>> print('백종현')
백종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

```
python3
13% 9.4 GB 84%
python (Python) ..basics/task04 (~zsh)
>>> with open('성적표.txt', 'a', encoding='utf-8') as f:
...     insertGrade(f)
...
학번, 이름, 성적을 입력하세요
학번: 214823
이름: 박종현
성적: 4.5
학번, 이름, 성적을 입력하세요
학번: 200123
이름: 홍길동
성적: 3.5
학번, 이름, 성적을 입력하세요
학번: 190231
이름: 임꺽정
성적: 3.1
학번, 이름, 성적을 입력하세요
학번: 188154
이름: 전우치
성적: 4.3
학번, 이름, 성적을 입력하세요
학번:
이름:
성적:
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```



python3

```
python (Python) 9.6 GB .83% . basics/task04 (~zsh)
>>> with open('성적표.txt', 'r', encoding='utf-8') as f:
...     with open('성적평균.txt', 'w') as fo:
...         calGradeAvg(f, fo)
...
>>> print('박종현')
박종현
>>> print('학번: 214823')
학번: 214823
>>>
```

cat 성적평균.txt

학생 수: 4

평균 학점: 3.8499999999999996

~Desktop/ai-basics/task04

[16] 9-3. 10~13 쪽

[17] 9-3. 15 쪽

```
python3
20% 9.3 GB 82%
>>> import pickle
>>> data = {}
>>> data[1] = { 'no': 1, 'name': '홍길동', 'addr': '광주시', 'age': 30 }
>>> data[2] = { 'no': 2, 'name': '임꺽정', 'addr': '천주시', 'age': 34 }
>>> data
{1: {'no': 1, 'name': '홍길동', 'addr': '광주시', 'age': 30}, 2: {'no': 2, 'name': '임꺽정', 'addr': '천주시', 'age': 34}}
>>>
>>> with open('피클 데이터.p', 'wb') as f:
...     pickle.dump(data,f)
...
>>> with open('피클 데이터.p', 'rb') as f:
...     data_r = pickle.load(f)
...
>>> data_r
{1: {'no': 1, 'name': '홍길동', 'addr': '광주시', 'age': 30}, 2: {'no': 2, 'name': '임꺽정', 'addr': '천주시', 'age': 34}}
>>>
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
>>>
```

[18] 9-3. 17~19 쪽