practice\practice_solutions.py

```
############################################################
# stdlib_math-01.py
import math

# 피타고라스 정리를 사용해서 두점 사이의 거리를 구하기
# define the two points as tuples
point1 = (3, 4)
point2 = (6, 8)

# calculate the distance using the distance formula
distance = math.sqrt((point2[0] - point1[0])**2 + (point2[1] - point1[1])**2)

# print the distance
print("The distance between point 1 and point 2 is", distance)
############################################################


############################################################
# stdlib_math-02.py
import math

a = int(input("첫 번째 수 a를 입력하세요: "))
b = int(input("두 번째 수 b(a보다 같거나 큰)를 입력하세요: "))

for i in range(1, a+1):
    if a % i == 0 and b % i == 0:
        print(i)

print( '최대공약수: ', math.gcd(a, b) )
############################################################


############################################################
# stdlib_os-01.py
print("dummy가 현재 폴더에 존재하는가?")
print( os.path.exists("dummy") )
print()

print("dummy가 폴더인가?")
print( os.path.isdir("dummy") )
print()

print("foo.txt가 폴더인가?")
print( os.path.isdir("foo.txt") )
print()

print(".\\path\\path2\\myfile.txt 경로를 os에 맞게 만들어주세요.")
print( os.path.join('.', 'path', 'path2', 'myfile.txt'))
print()

print("현재 폴더를 파고들어가면서 파일을 탐색합니다.")
```

```
for root, dirs, files in os.walk('.'):
    print("현재폴더:", root)

    for dir in dirs:
        print('  폴더-', dir)

    for file in files:
        print('  파일-', file)
############################################################


############################################################
# stdlib_os-02.py
import os

def get_file_list(root='.', prefix=''):
    files = os.listdir(root)

    for f in files:
        if os.path.isfile( os.path.join(root, f) ):
            print('F', f"{prefix}{f}")
        else:
            print('D', f"{prefix}{f}")
            get_file_list(os.path.join(root, f), prefix+"    ")

# 현재 폴더 절대 경로
cur_path = os.getcwd()
print(cur_path)

get_file_list(cur_path)
# get_file_list()
############################################################


############################################################
# stdlib_random.py
import random


print("0~1사이에 숫자를 하나 선택합니다.")
print(random.random())
print()

print("1~10사이에 정수 하나를 선택합니다.")
print( random.randrange(1, 11) )
print()

print("foo, bar, egg 중 하나를 선택합니다.")
print( random.choice(('foo', 'bar', 'egg')) )
############################################################


############################################################
# stdlib_time.py
import time
```

```
print("for 문 실행을 1초씩 지연시킵니다.")
for i in range(5):
    print(i)
    time.sleep(1)
print()

print("실행시간을 측정합니다.")
start= time.time()
for i in range(5):
    print(i)
    time.sleep(1)
end = time.time()
print("코드실행시간: ", end-start)
############################################################


############################################################
# file_w.py
with open('file_w.txt', 'w', encoding='utf-8') as f:
    f.write('파이썬 재밌다.\n')
    f.write('진짜가?\n')
    f.write('아니 ㅠㅠ\n')
############################################################


############################################################
# file_r.py
print("파일을 모두 읽습니다.")
with open('file_w.txt', 'r', encoding='utf-8') as f:
    # f.seek(6) # 숫자는 1 byte
    print( f.read() ) # 숫자는 한 글자(character)

print("\n파일을 한줄씩 읽습니다.")
with open('file_w.txt', 'r', encoding='utf-8') as f:
    print( f.readline(), end='' )
    print( f.readline(), end='' )

print("\n파일을 for문을 통해 모두 읽습니다.")
with open('file_w.txt', 'r', encoding='utf-8') as f:
    for line in f:
        print(line, end='')
    print()

print("\n파일을 모두 읽어 리스트로 반환 받습니다.")
with open('file_w.txt', 'r', encoding='utf-8') as f:
    lines = f.readlines()
    for line in lines:
        print(line, end='')
    print()
############################################################


############################################################
# file_a.py
```

```
with open('file_w.txt', 'a', encoding='utf-8') as f:
    f.write('좀 더해봐!')
############################################################


############################################################
# extlib_pil-01.py
# https://pillow.readthedocs.io/en/stable/index.html
# https://realpython.com/image-processing-with-the-python-pillow-library/

from PIL import Image, ImageFilter  # 라이브러리를 임포트 한다.

original = Image.open("gogh.png") # 하드 드라이브에서 이미지를 읽어 들인다.
print(original.mode)
blurred = original.filter(ImageFilter.BLUR) # 이미지를 흐리게, 경계선 검출 CONTOUR
blurred.save("gogh_blurred.png")

gray_img = original.convert("L")
gray_img.save("gogh_gray.png")

# original.show() # 두 이미지를 디스플레이 한다.
# blurred.show()
############################################################


############################################################
# extlib_pil-02.py
from PIL import Image, ImageFilter  # 라이브러리를 임포트 한다.
import argparse

#########################################################################
# https://docs.python.org/ko/3/library/argparse.html
# 이 부분은 파이썬 파일을 실행할 때 옵션을 받을 수 있도록 하는 부분입니다.
# 현재 만들고 싶은 프로그램은 다음처럼 동작하게 하고 싶습니다.
#
# extlib_pil-02.py gogh.png -a b : gogh.png를 흐림화시켜 gogh_blur.png로 저장
# extlib_pil-02.py gogh.png -a g : gogh.png를 흑배화시켜 gogh_gray.png로 저장
#
# 이렇게 하기 위해서는 extlib_pil-02.py 뒤에 입력이미지 파일을 적을 수 있어야 하고
# 흐림화할지 흑백하 할지 결정할 수 있는 옵션 -ab, -ag 를 줄수 있어야 합니다.
# 꽤 복잡한 작업이지만 내부 라이브러리 argparser를 쓰면 비교적 간단하게
# 처리할 수 있습니다.
# 이정도로 이해하고 아래 코드는 그대로 두고 진행합니다.
parser = argparse.ArgumentParser(description='my first pillow app.')
parser.add_argument('image')
parser.add_argument('-a', '--action', choices=['b', 'g'],
                    default='b', help='b: blur, g: grayscale')
args = parser.parse_args()
#########################################################################


def blur(filename):
    # 여기 입력받은 이미지 파일을 블러 시켜 *_blur.png로 저장
    print('blur')
    original = Image.open(filename)
```

```
    blurred = original.filter(ImageFilter.BLUR)
    blurred.save(f"{filename[:-3]}_blurred.png")

def gray(filename):
    # 여기 입력받은 이미지 파일을 흑백화 시켜 *_gray.png로 저장
    print('gray')
    original = Image.open(filename)
    gray_img = original.convert("L")
    gray_img.save(f"{filename[:-3]}_gray.png")

if __name__ == '__main__':
    # argparser는 다음처럼 커맨트 라인에 옵션으로 준 값을 저장합니다.
    # 입력된 이미지 파일이름: args.image
    # 입력된 처리 옵션: args.action
    if args.action == 'b':
        blur(args.image)
    else:
        gray(args.image)
############################################################
```