파이썬프로그래밍 과제08 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[Mini Project 4]

마지막으로 표준 입력으로 받은 종합 소득 금액으로 세금을 계산해 출력하는 코드를 작성하자. 과세 표준 구간의 세율과 누진 공제액이 쌓인 항목으로 리스트를 구성한다.

[소스코드]

'08-04-incometax.py'

'Mini Project 4 : 종합 소득 세금 출력'

''' 마지막으로 표준 입력으로 받은 종합 소득 금액으로 세금을 계산해 출력하는 코드를 작성하자.

과세 표준 구간의 세율과 누진 공제액이 쌓인 항목으로 리스트를 구성한다. '''

' 리스트와 튜플, 산술연산, if 조건, 표준 입력과 출력 활용 '

''' 세율과 누진 공제액 쌓인 항목의 리스트

taxrate = [(6, 0), (15, 1\_1080\_000), (24, 5\_220\_000), (35, 14\_900\_000),

(38, 19\_400\_000), (40, 25\_400\_000), (42, 35\_400\_000)]

종합 소득 과세 표준 구간에 따른 기준 종합 소득

base = (0, 12\_000000, 46\_000000, 88\_000000, 150\_000000, 300\_000000, 500\_000000) '''

# 세율과 누진 공제액 쌓인 항목의 리스트

taxrate = [(6, 0), (15, 1\_1080\_000), (24, 5\_220\_000), (35, 14\_900\_000),

(38, 19\_400\_000), (40, 25\_400\_000), (42, 35\_400\_000)]

# 종합 소득 과세 표준 구간에 따른 기준 종합 소득

base = (0, 12\_000000, 46\_000000, 88\_000000, 150\_000000, 300\_000000, 500\_000000)

def getsection(income):

'''

종합 소득 금액 income에 따른 과세 표준 구간 번호를 반환

인지 income: 종합 소득 금액

'''

if base[6] < income:

section = 6

elif base[5] < income:

section = 5

elif base[4] < income:

section = 4

elif base[3] < income:

section = 3

elif base[2] < income:

section = 2

elif base[1] < income:

section = 1

else:

section = 0

return section

income = int(input('종합 소득 금액 입력 >> '))

section = getsection(income)

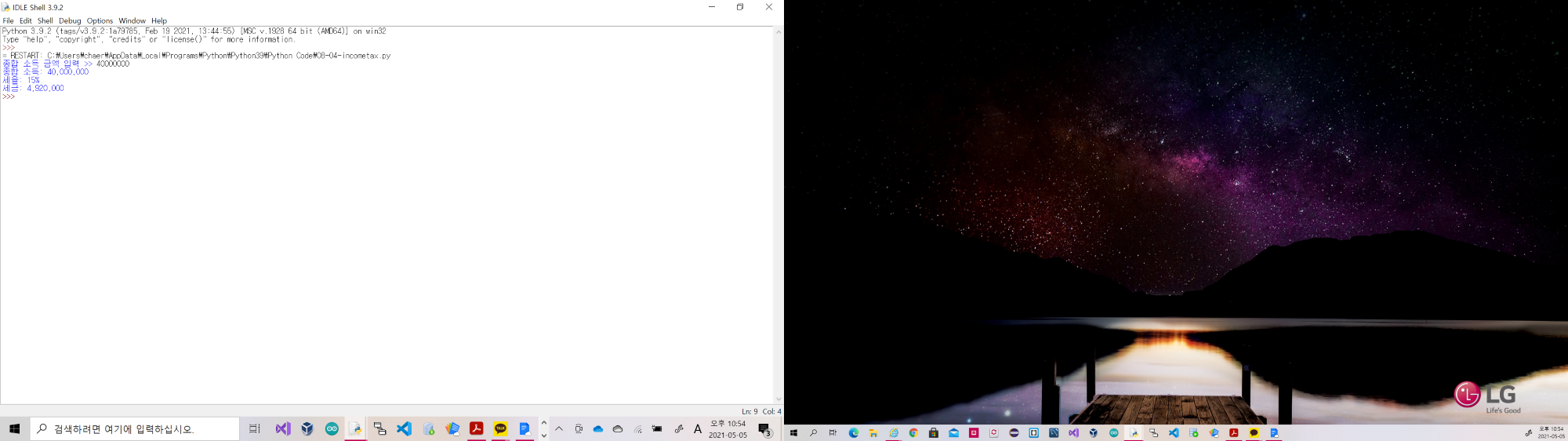
tax = income \* taxrate[section][0] / 100 - taxrate[section][1]

print("종합 소득: {:,.0f}".format(income))

print("세율: {}%".format(taxrate[section][0]))

print("세금: {:,.0f}".format(tax))

[실행결과]



[Mini Project 6]

0에서 5까지의 자릿수로 구성되는 정수를 디지털 정수 모양으로 출력

[소스코드]

'08-06-printdigitalint.py'

'Mini Project 6 : 0에서 5까지의 자릿수로 구성되는 정수를 디지털 정수 모양으로 출력'

'중첩리스트, while과 for in 반복 구문 활용'

LCD = [[ "|||||",

"| |",

"| |",

"| |",

"|||||" ],

[ " | ",

" || ",

" | ",

" | ",

" ||||" ],

[ "|||||",

" |",

"|||||",

"| ",

"|||||" ],

[ "|||||",

" |",

"|||||",

" |",

"|||||" ],

[ "| |",

"| |",

"|||||",

" |",

" |" ],

[ "|||||",

"| ",

"|||||",

" |",

"|||||" ]]

ROW = 5 # 디지털 숫자의 높이

def printLCD(digit):

"""

'234'의 문자열 숫자를 디지털 숫자로 출력하는 함수

인자

digit: 출력하려는 정수 형태의 문자열

"""

# 한 행에 출력하려는 여러 자릿수 각 행을 출력

for row in range(ROW):

# 출력하려는 자릿수만큼 디지털 행을 출력

for i in range(len(digit)):

# '32'라면 각각 3에 대해 row행, 2에 대해 row행 순으로

num = int(digit[i])

# 각 행을 한 줄에 계속 출력

print(LCD[num][row], end = ' ')

# 다음 줄로 이동

print()

# 문자열의 문자가 '012345'가 아니면 False 반환

def checkdgt(s):

tag = True

for c in s:

if c not in '012345':

tag = False

break

return tag

# 표준 입력

snum = '7'

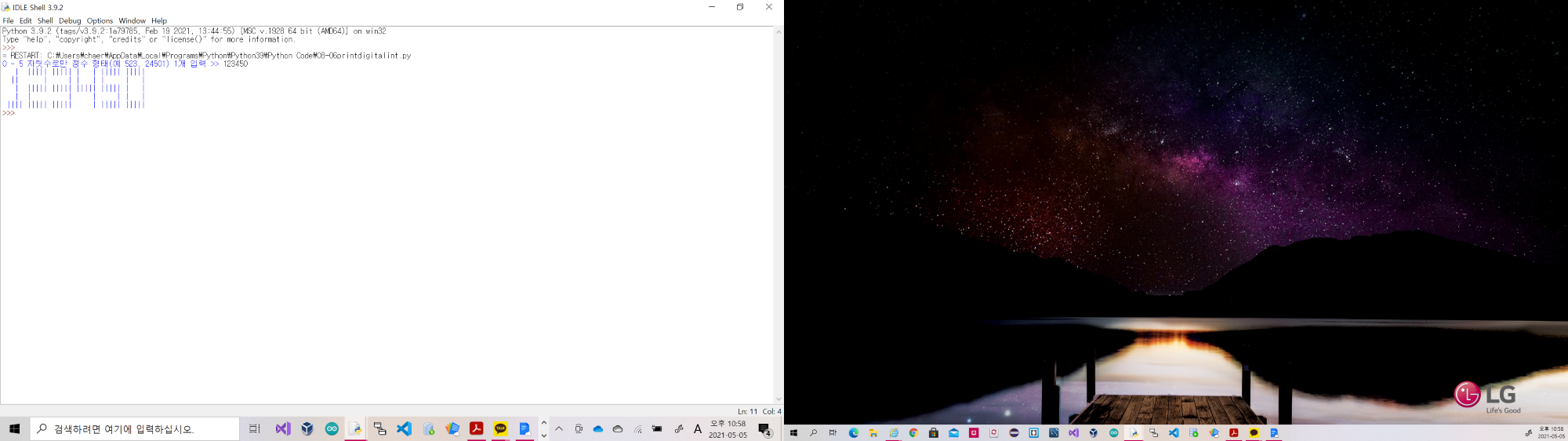
while not checkdgt(snum):

snum = input("0 ~ 5 자릿수로만 정수 형태(예 523, 24501) 1개 입력 >> ")

# 출력

printLCD(snum)

[실행결과]



[Mini Project 7]

조건문 if, 리스트와 딕셔너리, 모듈 random의 choice() 함수, while문을 활용하여 가위바위보 게임을 구현

[소스코드]

'08-07-rockgame.py'

'Mini Project 7 : 가위바위보 게임'

'조건문 if, 리스트와 딕셔너리, 모듈 random의 choice() 함수, while문 활용'

''' 다음 딕셔너리 madewin은 가위바위보에서 승패가 나눠지는 세 가지 경우를 항목으로 구성했다.

구성된 항목 앞의 키에는 승자, 값에는 패자의 손 모양을 기술한다.

# 가위바위보 게임 판정에 사용할 딕셔너리

madewin = {'가위': '보', # 가위는 보를 이김

'바위': '가위', # 바위는 가위를 이김

'보': '바위'} # 보는 바위를 이김 '''

# 가위바위보 중 하나를 임의로 선택하기 함수

from random import choice

# 가위바위보 게임 판정에 사용할 딕셔너리

madewin = {'가위': '보', # 가위는 보를 이김

'바위': '가위', # 바위는 가위를 이김

'보': '바위'} # 보는 바위를 이김

# rock scissors paper

game = ['가위# 바위# 보 게임 ', '가위', '바위', '보']

msg = ['비겼습니다', '축하합니다! 당신이 이겼습니다.']

msg.append('아쉽지만, 컴퓨터가 이겼습니다.')

# 시작 제목 출력

print((game[0] + '시작 ').center(55, '='), '\n')

# 게임 시작

while True:

try:

s = ' 0(종료), 1(가위), 2(바위), 3(보[자기]) 중 하나 선택 >> '

num = int(input(s))

except:

num = 1

if num == 0:

break;

if not (0 <= num <= 3): # 범위의 값이 아니면 다시 시작

print('\t입력이 잘못됐습니다. 다시 하세요!')

continue;

# 컴퓨터에게 하나 선택

player = game[num]

com = choice(game[1:]) # 가위바위보는 1부터 시작

# 게임 판정

if player == com:

winner = 0 # 비김 출력

# 참여자인 당신이 이기는 조건

elif madewin[player] == com:

winner = 1 # 당신 승리

# 컴퓨터가 이기는 조건

else:

winner = 2 # 컴퓨터 승리

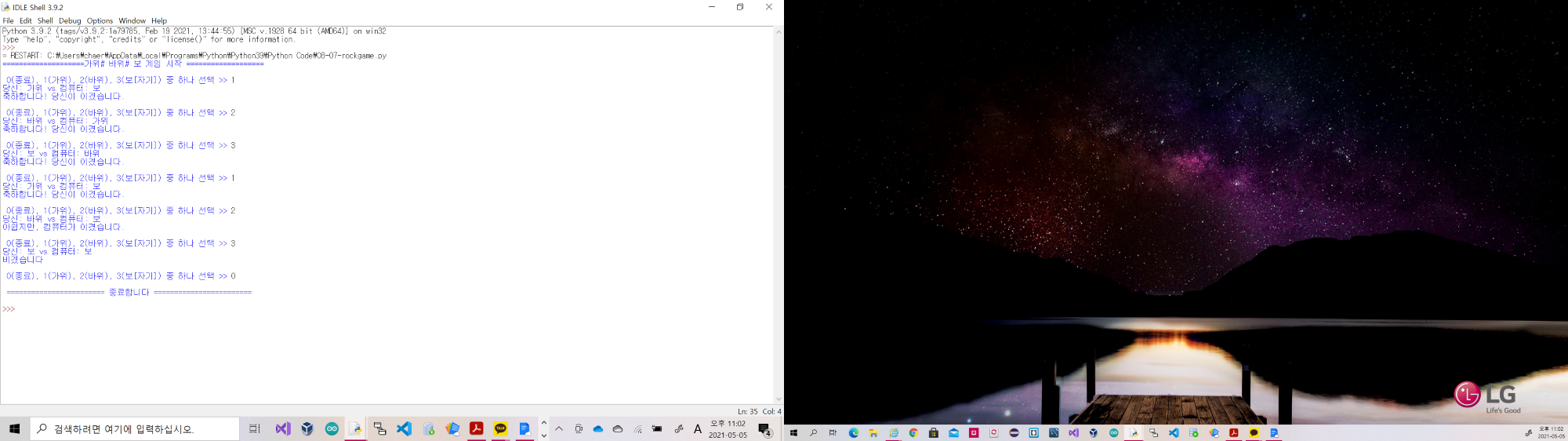
print('당신:', player, 'vs', '컴퓨터:', com)

print(msg[winner] + '\n')

# 종료 출력

print('\n', ' 종료합니다 '.center(55, '='), '\n')

[실행결과]



[Mini Project 8]

우리나라의 로또 당첨 순위를 알아보고 모든 등수와 등수외의 복권 번호를 의도적으로 만들어 출력해 보는 프로그램을 작성해 보자

[소스코드]

'08-08-lottogame.py'

'Mini Project 8 : 로또 복권 당첨 모의 실험'

'우리나라의 로또 당첨 순위를 알아보고 모든 등수와 등수외의 복권 번호를 의도적으로 만들어 출력해 보는 프로그램을 작성해 보자'

'딕셔너리와 튜플, 리스트, 조건, while과 for 반복, 모듈 random의 choice(), randint() 함수 활용'

''' 딕셔너리의 키(key)로 번호 6개 중 맞춘 개수로 정한다. 즉 키가 6은 1등으로 값(value)은

('1등'. 1000000000)처럼 등수와 상금의 튜플로 만든다. 다만, 5개를 맞히면 다시 키 1과 0으로,

2등과 3등으로 나눈다.실제 로또에서 3등 이상의 상금은 매회 달라지므로 다음처럼 정액 값으로 저장한다.

# 로또 복권 순위 판정에 사용할 딕셔너리

rank = { 6: ('1등', 1000000000), # 1등 10억 원, 맞춘 번호 6개

5: { 1: ('2등', 50000000), # 2등 5천만 원, 맞춘 번호 5개 + 보너스

0: ('3등', 1000000)}, # 3등 1백만 원, 맞춘 번호 5개

4: ('4등', 50000), # 4등 5만 원, 맞춘 번호 4개

3: ('5등', 5000) # 5등 5천원, 맞춘 번호 3개

} '''

from random import sample

from random import randint

# 로또 복권 순위 판정에 사용할 딕셔너리

rank = { 6: ('1등', 1000000000), # 1등 10억 원, 맞춘 번호 6개

5: { 1: ('2등', 50000000), # 2등 5천만 원, 맞춘 번호 5개 + 보너스

0: ('3등', 1000000)}, # 3등 1백만 원, 맞춘 번호 5개

4: ('4등', 50000), # 4등 5만 원, 맞춘 번호 4개

3: ('5등', 5000) # 5등 5천원, 맞춘 번호 3개

}

def makenum(same):

nums = winnum.copy()

# 다르게 만들기 위해 제거 6-same개 제거

for i in range(6 - same):

nums.pop()

# 당첨 번호가 아닌 2개를 넣어 모두 6개 만들기

while len(nums) != 6:

n = randint(1, 45)

if n not in winnum:

nums.add(n)

return nums

def getwinner(lotto):

''' 6개의 로또 번호에서 각각 당첨 번호 개수와 수를 출력 '''

global bonus

for i, item in enumerate(lotto):

print('%c' % (ord('A') + i), end = ' ')

win = winnum.intersection(item) # 당첨 번호 구하기

if win:

wincnt = len(win)

print('당첨 번호 개수 %d, ' % wincnt, end = '')

printnums(win)

if 3 <= wincnt:

grade = rank[wincnt]

if 5 == wincnt:

if bonus in item:

print('\t 보너스 번호 %d도 맞춤!!!' % bonus)

grade = rank[wincnt][1] # 2등 지정

else:

grade = rank[wincnt][0] # 3등 지정

print('\t%s %s원' % (grade[0], grade[1]))

else:

print('\t 2개 이하 맞춰, 꽝!')

else:

print('모두 꽝!!!')

def printlotto(lotto):

''' 로또 복권 표와 같이 출력 '''

for i, item in enumerate(lotto):

print('%c 자동 ' % (ord('A') + i), end = ' ')

printnums(item)

print()

def printnums(nums):

''' 시퀀스인 수를 정렬해 출력 '''

for num in sorted(nums):

print('%02d' % num, end = ' ')

print()

# 7개 선정

winnum = set(sample(list(range(1, 46)), 7))

# 1개를 빼 보너스 번호로 사용

bonus = winnum.pop()

print('당첨 번호: ', end = ' ')

printnums(winnum) # 당첨 번호 출력

print('보너스 번호: ', bonus)

print()

lottos = []

# 1등 번호(6개 모두 같은 셋)를 만들어 로또 리스트에 추가

rank1num = winnum.copy()

lottos.append(rank1num)

# 2등 번호(5개 같고, 1개는 보너스 번호)를 만들어 로또 리스트에 추가

rank2num = winnum.copy()

rank2num.pop(); rank2num.add(bonus)

lottos.append(rank2num)

# 3등 번호(5개 같고, 1개는 보너스 번호와 다르게)를 만들어 로또 리스트에 추가

rank3num = winnum.copy()

rank3num.pop();

# 보너스 번호가 아니고 기존의 번호가 아니면 1개 추가

while True:

num = randint(1, 45)

if num != bonus and num not in rank3num:

rank3num.add(num)

break

lottos.append(rank3num)

# 4개부터 0개 맞는 번호 자동 생성

for i in range(5):

lottos.append(makenum(4 - i))

# 로또 번호 출력

printlotto(lottos)

# 로또 번호와 당첨 번호 비교해 순위 결정 출력

getwinner(lottos)

[실행결과]

