# 9주 과제

# 6번 문제

'''

arr = (1, 2, 3)

brr = list(arr)

temp = brr[2]

brr[2] = brr[1]

brr[1] = temp

print(brr)

'''

# 7번 문제

'''

num = [10, 9, 9, 6, 7, 6, 8, 9]

num = set(num)

print(num)

'''

# 8번 문제

'''

menu = { "돈가스":5000, "생선가스":5500, "우동":2500, "초밥 세트":9000 }

for i in menu :

print(i, menu[i])

'''

# 9번 문제

'''

compA = [ "회장1", "이사1", "영업1", "영업2", "생산1", "생산2" ]

compB = [ "회장1", "사외이사1", "영업1", "홍보1" ]

com = set(compA + compB)

print(com)

'''

# 10번 문제

'''

num1 = int(input("첫 번째 수 : "))

num2 = int(input("첫 번째 수 : "))

numa = []

numb = []

numc = set()

for i in range(1, num1+1) :

if num1 % i == 0 :

numa.append(i)

print("첫 번째 수의 약수는 {}".format(set(numa)))

for i in range(1, num2+1) :

if num2 % i == 0 :

numb.append(i)

print("첫 번째 수의 약수는 {}".format(set(numb)))

print("20와 10의 공약수는")

numc = set(numa) & set(numb)

print("{}".format(numc))

'''

# 11번 문제

'''

dic = {}

while True :

select = int(input("1. 회원가입, 2. 프로그램 종료 "))

if select == 2 :

print("-"\*20)

print("아이디 : 비밀번호")

print("-"\*20)

for i in dic.keys() : # 키 값으로만 접근

print(i, "\t", dic[i])

print("-"\*20)

break

if select == 1 :

\_id = input("아이디를 입력하세요. : ")

\_pw = int(input("비밀번호를 입력하세요. : "))

dic[\_id] = \_pw

'''

# 12번 문제

'''

exam = ( ["3+2",5,3], ["5/2의 몫",2,5], ["10-2",8,3], ["1-(10/4의 나머지)",-1,5], ["4\*2+(100)",108,3], ["3\*7\*1+2",23,5], ["12/3\*7",28,3] )

score = 0

O\_cnt = 0

X\_cnt = 0

for i in range(0, len(exam)) :

print("문제 : {}".format(exam[i][0]))

dap = int(input("정답을 입력하세요. "))

if dap == exam[i][1] :

score += exam[i][2]

O\_cnt += 1

else :

X\_cnt += 1

print("-"\*20)

print("정답 개수 : {}".format(O\_cnt))

print("오답 개수 : {}".format(X\_cnt))

print("총점 : {}".format(score))

print("-"\*20)

'''

# 10주 과제

# 실습 1 가위바위보 게임 while문

'''

import random as r # random 대신 r을 사용가능

w\_cnt = 0

l\_cnt = 0

b\_cnt = 0

while True :

user = int(input("가위 : 1, 바위 : 2, 보 : 3, 종료 : 4 "))

if user == 4 :

print("종료되었습니다.")

print("이긴 횟수 : {}, 진 횟수 : {}, 비긴 횟수 : {}".format(w\_cnt, l\_cnt, b\_cnt))

break

com = r.randint(1, 3) # randint() : 숫자 범위 모두 포함

print("user : {}".format(user, com))

if user == com :

print("비겼습니다.")

b\_cnt += 1

elif (user == 1 and com == 3) or (user == 2 and com == 1) or (user == 3 and com == 2) :

print("user 승!")

w\_cnt += 1

else :

print("com 승")

l\_cnt += 1

'''

# 실습 2 가위바위보 list, dict 사용

'''

import random as r

str = ['', '가위', '바위', '보']

rule = {1:3, 2:1, 3:2}

while True :

print('-'\*40)

user = int(input('가위 : 1, 바위 : 2, 보 : 3, 종료 : 4'))

if user == 4 :

print("종료되었습니다.")

break

com = r.randint(1,3)

print('user : {}, com : {}'.format(str[user], str[com]))

print('-'\*40)

if user == com :

print('비겼습니다.')

elif rule[user] == com : # 키값의 접근하여 키값이 같으면 이김

print('이겼습니다.')

else :

print('졌습니다.')

'''

# 실습 3 set 자료형으로 만드는 lotto 번호

'''

import random as r

lucky\_set = set()

while len(lucky\_set) < 6 :

n = r.randint(1,45)

lucky\_set.add(n)

lucky = list(sorted(lucky\_set))

print(lucky)

'''

# 실습 4 컴프리핸션으로 수정하기 공약수 구하기

'''

f\_num = int(input('첫 번째 수를 입력하세요! : '))

n\_num = int(input('두 번째 수를 입력하세요! : '))

f\_set = set(i for i in range(1, f\_num+1) if f\_num % i == 0)

n\_set = set(i for i in range(1, n\_num+1) if n\_num % i == 0)

print('첫 번째 수의 약수는 {}'.format(sorted(f\_set)))

print('두 번째 수의 약수는 {}'.format(sorted(n\_set)))

print('{}와 {}의 공약수는 {}'.format(f\_num, n\_num, sorted(f\_set & n\_set)))

'''

# 실습 5 number list 내부에 들어있는 숫자가 몇 번 등장하는지를 출력하는 프로그램

'''

import random

number = []

counter = {}

# 숫자 10개를 요소로 갖는 리스트 생성

number = [random.randint(1,9) for i in range(10)]

# 컴프리핸션으로 사용

print('number', number)

# 숫자 등장 횟수 계산 - 딕셔너리 사용

for i in number :

if i not in counter : # counter가 i에 있나 확인

counter[i] = 1 # 없으면 1로 변경

else :

counter[i] = counter[i]+1 # 있으면 +1 연산 수행

# 결과 출력

for i in sorted(counter) : # 수 정렬

print(i, " : ", counter[i]) # 출력

'''

# 실습 6 암호 해독 사전 만들기

'''

secret\_msg = ""

key\_list = []

value\_list = []

key\_list = [chr(i) for i in range(97,123)] # chr()은 아스키코드에 해당하는 문자로 변환

value\_list = list(set('abcdefghijklmnopqrstuvwxyz')) # 위치가 정렬이 되는 것이 아닌 랜덤으로 지정

secret\_code = dict(zip(key\_list, value\_list)) # zip()은 딕셔너리의 키와 값으로 만들어줌

for key\_, value\_ in secret\_code.items() :

print("{} -> {}".format(key\_, value\_), end=" , ")

# 암호화 하기

msg = list(input('암호화 할 단어를 입력하세요. >>> '))

for i in msg :

secret\_msg += secret\_code[i] # 입력한 문자 하나하나씩 키에 들어가서 값을 찾음

# secret\_msg에 넣는 이유는 복호화에 사용할려고 함.

print(secret\_code[i], end='') # 키의 값 출력

# 복호화 하기

print()

print('복호화 결과')

decode = dict(zip(value\_list, key\_list)) # 키와 값을 역으로 저장

for key\_, value\_ in secret\_code.items() : # items()가 key\_와 value\_에게 각각이 키와 벨류를 할당

print("{} -> {}".format(key\_, value\_), end=" , ")

print()

for i in secret\_msg : # 복호화에 사용할 secret\_msg로 반복

print(decode[i], end="")

'''

# 11주 과제

# 문제 1번 - 문자열의 중간마다 #을 끼워넣는 코드

'''

ss = 'Python'

for i in range(0, len(ss)) :

print(ss[i] + '#', end = '')

'''

# 문제 2번 - 문자열을 반대의 순서로 출력하는 프로그램

'''

inStr, outStr = 'Python', ""

strlen = len(inStr)

for i in range(0, strlen) :

outStr += inStr[strlen - i - 1] # - 1을 하는 이유는 안하면 index가 넘어감

print("내용을 거꾸로 출력 --> %s" % outStr)

'''

# 문제 8번 - 영문자랑 숫자가 섞여잇나

'''

pw = input('비밀번호를 입력해주세요(8자리 이상, 영문자+숫자로 조합) : ')

if len(pw) >= 8 :

if pw.isalpha() == False and pw.isdigit() == False : # isalpha()나 isdigit()이 참이 되면 알파벳이나 숫자로만 구성되어 있는거므로 거짓으로 함

print("비밀번호는 {}입니다.".format(pw))

else :

print('비밀번호는 영문자와 숫자의 조합입니다.')

else :

print('비밀번호는 8자리 이상입니다.')

'''

# 문제 9번 - 이름은 영문자, 학번은 숫자, 주소는 영문자인지 확인하고 주소는 첫 글자가 대문자로

'''

while True :

name, hak, add = input('Data 입력 (이름 학번 주소) : ').split(' ')

if name.isalpha() == True and hak.isdigit() == True and add.isalpha() == True :

print("['{}', '{}', '{}']".format(name, hak, add.capitalize()))

else :

print('다시 입력하세요')

'''

# 문제 10번 - 문자열 입력한 빈도수 세기

'''

Ap, ap, di, ha, gi = 0, 0, 0, 0, 0

ch = '' # 문자로 초기화

mun = '' # 문자로 초기화

mun = input("문자열을 입력하세요 : ") # mun으로 넣음

# A = 65, Z = 90

# a = 97, z = 122

# 0 = 48, 9 = 57

# ㄱ = 12593, 힣 = 54724

for ch in mun :

if ord(ch) >= ord('A') and ord(ch) <= ord('Z') : # ord()는 아스키코드로 변환

Ap += 1

elif ord(ch) >= ord('a') and ord(ch) <= ord('z') :

ap += 1

elif ord(ch) >= ord('0') and ord(ch) <= ord('9') :

di += 1

elif ord(ch) >= ord('ㄱ') and ord(ch) <= ord('힣') :

ha += 1

else :

gi += 1 # 공백까지

print('대문자 : {} 소문자 : {} 숫자 : {} 한글 : {} 기타 : {}'.format(Ap, ap, di, ha, gi))

'''

# 문제 11번 - 입력된 문자열에서 소문자의 발생 빈도를 세는 프로그램 작성

'''

countDic = {}

inStr = input('문자열을 입력하세요 > ')

for ch in inStr :

if 'A' <= ch and 'z' >= ch :

if ch in countDic : # ch 안에 countDic가 있으면 값에 1을 추가

countDic[ch] += 1

else :

countDic[ch] = 1 # 없으면 키와 값을 추가

print("원문" + "\*" \* 50, "\n" + inStr);

print("-" \* 20);

print("문자" + " " + "빈도수");

print("-" \* 20);

for i in sorted(countDic) :

print(i, '\t ', countDic[i]);

'''

# 단어 세기

'''

countDic = {}

inStr = input('여러 단어를 입력하세요 > ')

inList = inStr.split() # 단어이므로 split()으로 쪼개줘야 함

for ch in inList :

if ch in countDic : # ch안에 countDic가 있으면

countDic[ch] += 1 # 해당하는 키의 값을 1을 더해줌

else : # ch안에 countDic가 없으면

countDic[ch] = 1 # 키와 값을 추가

print("원문" + "\*" \* 50, "\n" + inStr);

print("-" \* 20);

print("문자" + " " + "빈도수");

print("-" \* 20);

for i in sorted(countDic) : # 정렬

print(i, '\t ', countDic[i]);

'''

# 시저 암호 만들기

'''

# 암호화

msg = input('암호화할 문장을 영어로 입력하세요. ')

key = int(input('암호화 키(1~26 숫자)를 입력하세요. '))

encode = '' # 문자로 초기화

for a in msg :

tmp = ord(a) + key # 각 알파벳을 아스키코드로 바꾼 후 key를 더함

a = chr(tmp) # 다시 문자로 변환

encode = encode + a # 문자를 더함

print("평 문 : ", msg)

print("암 호 키 : ", key)

print("암 호 문 : ", encode);

# 복호화

print("암호 복호화 하기")

decode = ''

for a in encode :

tmp = ord(a) - key # 각 알파벳을 아스키코드로 바꾼 후 key를 뺌

a = chr(tmp) # 다시 문자로 변환

decode = decode + a # 문자를 더함

print("복호문 : ", decode);

'''

# 12주 과제

# 문제 1번

'''

# 리스트 math에는 학생 여러 명의 수학 성적이 저장되어 있습니다.

# 리스트를 score\_ info( )라는 함수로 넘겨서,

# 학생 수, 최고 성적, 최저 성적, 평균을 반환하는 프로그램을 작성하세요.

# len(), max(), min(), sum() 함수 사용

def score\_info(a) :

return len(a), max(a), min(a), sum(a)/len(a)

math = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

a,b,c,d = score\_info(math)

print('학생수는 {}입니다.'.format(a))

print('최고점 : {}'.format(b))

print('최저점 : {}'.format(c))

print('평균 : {}'.format(d))

'''

# 문제 2번

'''

# 덧셈, 뺄셈, 곱셈, 나눗셈을 하는 계산기 함수를 작성하시오.

# 계산기 함수는 main으로부터 연산자와,

# 피연산자 두 개를 전달받아 연산 후 연산한 결과를 main 으로 return함

def g(a, b, c) :

if c == '+' :

return a + b

if c == '-' :

return a - b

if c == '\*' :

return a \* b

if c == '/' :

return a / b

y = input("연산자를 입력하세요(+,-,\*,/) : ")

num1 = int(input('첫 번째 수를 입력하세요 : '))

num2 = int(input('두 번째 수를 입력하세요 : '))

x = g(num1, num2, y)

print('{} {} {} = {}'.format(num1,y,num2,x))

'''

# 문제 3번

'''

# 리스트 score에는 학생들의 성적이 저장되어 있습니다.

# 모든 학생의 성적을 가산하는 프로그램을 작성하시오.

# 가산 점수는 1~10점중 선택하여 입력, 가산점과 score는 main에서 넘겨줌.

# score = [90, 88, 84, 77, 85, 97, 60, 66, 79, 93]

def sum(s, n) :

for i in range(len(s)) : # 배열을 넘겼으므로 len으로 길이를 알아야 함.

s[i] += n

return s

score = [90, 88, 84, 77, 85, 97, 60, 66, 79, 93]

num = int(input('가산 점수 입력 : '))

print('가산 전 성적 : {}'.format(score))

print('{}점씩 가산한 최종 성적 : {}'.format(num, sum(score, num)))

'''

# 문제 4번

'''

# 전달받은 매개변수의 개수와 관계없이 모든 매개변수를 곱한 결과를

# 반환하는 프로그램을 작성하시오.

def multiply(\*args) : # \*를 붙이면 가변이 됨

num = 1

for i in range(len(args)) : # 리스트로 접근

num \*= args[i]

return num

print('multiply(5,7,9) 매개변수가 3개인 함수를 호출한 결과 --> {}'.format(multiply(5,7,9)))

print('multiply(1,2,3,4) 매개변수가 4개인 함수를 호출한 결과 --> {}'.format(multiply(1,2,3,4)))

print('multiply(1,2,3,4,5,6,7,8,9,10) 매개변수가 10개인 함수를 호출한 결과 --> {}'.format(multiply(1,2,3,4,5,6,7,8,9,10)))

'''

# 13주 과제

'''

from tkinter import \* # tkinter는 파이썬에서 GUI 관련 모듈 제공해주는 표준 윈도 라이브러리

window = Tk()

# 부모 윈도우

# Tk()는 기본이 되는 윈도를 반환, 이를 루트 윈도 또는 베이스 윈도라고 함.

# 이 부분에서 화면을 구성하고 처리

entry = Entry(window)

# Entry안에 window를 만들고 변수에 대입

entry.pack()

# 배치관리자 중 하나

# pack()있어야 배치 가능

window.title("윈도창 연습")

# 창 이름

window.geometry("400x100")

# 가로 400 x 세로 100

window.resizable(width = FALSE, height = FALSE)

# 크기 변경 X

window.mainloop()

# 위 행에서 베이스 윈도를 window 변수에 넣고 window.mainloop() 함수 실행

# 위젯의 이름 첫 글자는 대문자

# .configure(옵션) : 제일 많이 사용

'''

# 도전횟수 추가 할 때 활용

'''

from tkinter import \*

def cnt() :

global count

count += 1

label2.config(text = count)

def in\_txt() :

msg = ent.get()

label3.config(text = msg)

count = 0

window = Tk()

photo = PhotoImage(file = "가위바위보/rock.png")

label = Label(window, image = photo)

label2 = Label(window, text = count)

label3 = Label(window, text = "")

btn = Button(window, text="눌러봐~", command = cnt)

btn1 = Button(window, text="in\_txt", command = in\_txt)

ent = Entry(window, font=("맑은고딕", 30))

ent.pack()

btn.pack()

label.pack()

label2.pack()

label3.pack()

btn1.pack()

window.mainloop()

'''

# 가위바위보 게임

'''

from tkinter import \*

import random as r

def pass\_s() :

choice("가위") # 선택한 그림에 해당하는 가위바위보 선택

# choice("가위")함수 호출

def pass\_r() :

choice("바위") # 선택한 그림에 해당하는 가위바위보 선택

# choice("바위")함수 호출

def pass\_p() :

choice("보") # 선택한 그림에 해당하는 가위바위보 선택

# choice("보")함수 호출

def choice(human) : # 이미지를 선택 후 이동

global count, w\_cnt, l\_cnt, b\_cnt

computer = r.choice(["가위", "바위", "보"]) # choice() 함수를 사용하여 computer 변수에 랜덤으로 들어감

if computer == "바위" : # computer가 바위일 경우

com\_image["image"] = rock\_image # 이미지를 바위로 바꿈

elif computer == "보" : # computer가 보일 경우

com\_image["image"] = paper\_image # 이미지를 보로 바꿈

else : # computer가 가위일 경우

com\_image["image"] = scissors\_image # 이미지를 가위로 바꿈

if victory[computer] == human : # 필승법을 토대로 ex) (victory[바위] == 보) == human(보)

result = "인간 승리!" # 필승법과 같을 경우 승리

w\_cnt += 1

elif computer == human : # computer 과 human이 같으면

result = "비겼습니다." # 비김

l\_cnt += 1

else : # 아닐 시

result = "컴퓨터 승리!" # 컴터 승

b\_cnt += 1

count += 1

output.config(text = "인간 : " + human + " 컴퓨터 : " + computer + " " + result) # output의 text를 바꿈

msg3.config(text = "이긴 횟수는 {}, 진 횟수는 {}, 비긴 횟수는 {}, 총 {}판을 하였습니다.".format(w\_cnt, l\_cnt, b\_cnt, count))

count = 0

w\_cnt = 0

l\_cnt = 0

b\_cnt = 0

victory = {"바위":"보", "가위":"바위", "보":"가위"} # 필승법 딕셔너리로 생성

win = Tk() # 베이스 윈도우

rock\_image = PhotoImage(file = "가위바위보/rock.png") # 이미지 불러오기

paper\_image = PhotoImage(file="가위바위보/paper.png") # 이미지 불러오기

scissors\_image = PhotoImage(file="가위바위보/scissors.png") # 이미지 불러오기

# 위젯 생성

msg3 = Label(win, text=count, font=("맑은 고딕", "16"))

msg1 = Label(win, text="선택하세요", font=("맑은 고딕", "16")) # label 생성 후 '선택하세요'의 스타일 지정

rock = Button(win, image=rock\_image, command=pass\_r) # button을 생성, 이미지로 바꾸고 눌렀을 시 pass\_r 함수로 이동

paper = Button(win, image=paper\_image, command=pass\_p) # button을 생성, 이미지로 바꾸고 눌렀을 시 pass\_p 함수로 이동

scissors = Button(win, image=scissors\_image, command=pass\_s) # button을 생성, 이미지로 바꾸고 눌렀을 시 pass\_s 함수로 이동

msg2 = Label(win, text="컴퓨터는 다음을 선택하였습니다.", font=("맑은 고딕", "16")) # label 생성 후 '컴퓨터는 다음을 선택하였습니다.'의 스타일 지정

com\_image = Label(win, image=rock\_image) # label 생성 후 기본 이미지를 바위로 지정

output = Label(win, text="", font=("맑은 고딕", "16")) # label 생성 후 ''의 스타일 지정

# 위젯 배치

# grid는 표

msg1.grid(row=0, column=0, columnspan = 3) # 0번재 행의 3개의 열을 합침

scissors.grid(row=1, column=0) # 1번째 행의 0번째 열

rock.grid(row=1, column=1) # 1번째 행의 1번째 열

paper.grid(row=1, column=2) # 1번째 행의 2번째 열

msg2.grid(row=2, column=0,columnspan = 3) # 2번째 행의 3개의 열을 합침

com\_image.grid(row=3, column=0,columnspan = 3) # 3번째 행의 3개의 열을 합침

output.grid(row=4, column=0,columnspan = 3) # 4번째 행의 3개의 열을 합침

msg3.grid(row=5, column = 0, columnspan = 3)

'''

# up-down 게임

'''

from tkinter import \*

import random as r

def inNum() :

global count # 전역변수 선언

global num1 # 전역변수 선언

num = int(ent\_inNum.get()) # ent\_inNum.get()은 Entry에 입력한 값 가져오기

# get()으로 가져올 경우 문자

if num == comNum : # 랜덤으로 지정한 수가 가져온 수와 같으면

lab\_M.config(text = "정답입니다.") # 출력

elif num > comNum : # 랜덤으로 지정한 수가 가져온 수보다 크면

lab\_M.config(text = "DOWN") # 출력

else : # 둘 다 해당이 안되면

lab\_M.config(text = "UP") # 출력

count += 1 # 0으로 초기화한 count에 버튼이 눌릴 떄 마다 1씩 더함

lab\_C.config(text = count) # lab\_C에 text를 count로 변경

num1 += ent\_inNum.get() + ", " # ent\_inNum.get()은 Entry에 입력한 값 가져온 후 ','를 더해 num1에 더함

lab\_Y.config(text = num1) # lab\_Y에 text를 num1에 저장되어 있는 문자들로 초기화

def reset() :

global comNum

comNum = r.randint(0, 50)

ent\_inNum.delete(0, END)

lab\_M.config(text = "1~100까지 숫자 중 선택!")

count = 0

lab\_C.config(text = count)

num1 = ""

lab\_Y.config(text = num1)

# main

msg = "1 ~ 100까지 숫자 중 선택!"

comNum = r.randint(0, 100) # 0 ~ 100 까지 랜덤 숫자 생성

count = 0 # count 변수 초기화

num1 = "" # num1 문자로 초기화

win = Tk() # 부모 윈도우

win.title("13주 과제 -- UP DOWN GAME")

lab\_T = Label(win, text="도전횟수 : ", font = ("맑은고딕", 15), height = 2) # '도전횟수 : ' 출력, 폰트 스타일 지정

lab\_C = Label(win, text=count) # count 출력

lab\_M = Label( win, text = msg, width = 20, font = ("맑은고딕", 12), height = 2) # msg 출력, label 가로 길이 조정, 폰트 스타일 지정

lab\_Y = Label(win, text = num1, bg = "yellow", width = 80, height = 2) # num1 출력, 배경색 지정, label 가로 세로 길이 조정

ent\_inNum = Entry( win, width = 45, font = ("맑은고딕", 18), bg = "yellow green") # Entry(입력공간) 색 지정

btn\_inNum = Button( win, width = 20, text = "숫자 입력", font = ("맑은고딕", 15), command = inNum) # '숫자 입력'을 누르면 inNum 함수로 이동

btn\_reSet = Button( win, width = 15, text = "다시 도전", font = ("맑은고딕", 15), command = reset) # '다시 도전'을 누르면 reset 함수로 이동

btn\_end = Button( win, width = 20, text = "게임 종료", font = ("맑은고딕", 15), command = quit) # '게임 종료'를 누르면 quit 함수로 이동

lab\_M.grid(row = 0, column = 0, columnspan = 2) # 0번째 행의 0번째 열부터 1번째 열까지 병합

lab\_T.grid(row = 0, column = 2, columnspan = 2) # 0번째 행의 2번째 열부터 3번째 열까지 병합

lab\_C.grid(row = 0, column = 3) # 0번째 행의 3번째 열

ent\_inNum.grid(row = 1, column = 0, columnspan = 4) # 1번째 행의 4번째 열까지 병합

btn\_inNum.grid(row = 2, column = 0, columnspan = 2) # 2번째 행의 0번째 열부터 1번재 열까지 병합

btn\_reSet.grid(row = 2, column = 2) # 2번째 행의 2번째 열

btn\_end.grid(row = 2, column = 3) # 2번째 행의 3번째 열

lab\_Y.grid(row = 3, columnspan = 4) # 3번째 행의 0번째 열부터 3번째 열까지 병합

win.mainloop()

'''

# 14주 과제 - 사진 앨범

'''

from tkinter import\*

# 전역 변수 선언 부분

fnameList = [] # 리스트로 선언

for i in range(1, 10) : # 1 ~ 9까지 반복

fnameList.append('pic' + str(i) + '.gif') # 1번재 이미지부터 9번째 이미지의 경로를 추가

photoList = [None]\*9

num = 0 # num을 0으로 초기화

# 함수 선언 부분

def clickNext() :

# 다음 페이지

global num # 전역 변수 선언

num += 1 # 클릭시 1 추가

if num > 8 : # num의 크기가 8보다 커지면

num = 0 # num을 0으로 초기화

photo = PhotoImage(file = 'gif/' + fnameList[num]) # 전역 변수로 선언된 fnameList를 더하여 경로 탐색

pLabel.config(image = photo) # pLabel의 image의 경로를 바꿈

pLabel.image = photo # .image = photo 는 현재 이미지를 출력함.

file\_name.config(text = '현재 이미지 : ' + fnameList[num]) # 무슨 경로인지를 출력

def clickPrev() :

# 다음 페이지

global num # 전역 변수 선언

num -= 1 # 클릭시 1 감소

if num < 0 : # num이 0보다 작아지면

num = 8 # num을 8로 바꿈

photo = PhotoImage(file = 'gif/' + fnameList[num]) # 전역 변수로 선언된 fnameList를 더하여 경로 탐색

pLabel.config(image = photo) # pLabel의 image의 경로를 바꿈

pLabel.image = photo # .image = photo 는 현재 이미지를 출력함.

file\_name.config(text = '현재 이미지 : ' + fnameList[num]) # 무슨 경로인지를 출력

# 메인 코드 부분

window = Tk() # 부모 윈도우

window.geometry("700x500") # Tk에서의 창의 크기를 지정 (가로 X 세로)

window.title('사진 앨범 보기') # 창 이름

btnPrev = Button(window, text = "<< 이전", command = clickPrev) # 버튼을 누르면 clickPrev()함수로 이동

btnNext = Button(window, text = "다음 >>", command = clickNext) # 버튼을 누르면 clickNext()함수로 이동

file\_name = Label(window, text = "") # 레이블 추가

photo = PhotoImage(file = "gif/" + fnameList[0]) # 첫 사진을 file의 경로에 해당하는 첫 번째 이미지를 출력

pLabel = Label(window, image = photo) # 레이블 추가

file\_name.place(x = 80, y = 10) # 좌표

btnPrev.place(x = 250, y = 10)

btnNext.place(x = 400, y = 10)

pLabel.place(x = 250, y = 100)

window.mainloop()

'''