# # Input\_while\_examples

# Section 1:

print('Section 1:\n\n')

# # Part (A)

print("\nPart A: \n")

# Take input from the user and store it in a variable.

# The prompt should ask for a name.

# Print 'Hello' and the input variable.

### name = input("Please enter your name: ")

# print (f"\nHello {name}!")

# # Part (B)

print("\nPart B: \n")

# Use the compound operator += to create a multiline prompt.

# The prompt should say hello on one line and ask for a name on the next line.

# Print 'Hello' and the input variable.

### prompt\_0 = "Hello, "

# prompt\_0 += "\nWhat is your name? "

# name = input(prompt\_0)

# print(f"\nHello, {name}!")

# # Part (C)

print("\nPart C: \n")

# Ask the user for a first name and a last name in separate input statements.

# Print 'Hello' and the full name.

### prompt\_first = "Please enter your first name "

# prompt\_first += "\nFirst:"

# first\_name = input(prompt\_first)

# prompt\_last = "Now your last "

# prompt\_last += "\nLast:"

# last\_name= input(prompt\_last)

# print(f'hello, {first\_name} {last\_name}')

# # Part (D)

print("\nPart D:  \n")

# Ask the user for an age and convert it to an integer.

# Print the age.

###age = input('How old are you? ')

# age = int(age)

# print(age)

# # Part (E)

print("\nPart E: \n")

# Ask the user for an age and determine if they are able to vote (18 or older).

# Print a message indicating if they can vote or not.

### age = input('How old are you? ')

# age = int(age)

# if age >= 18:

#     print("\nYou are able to vote!")

# else:

#     print("\nYou cannot vote until you are 18 and older.")

# # Part (F)

print("\nPart F: \n")

# Prompt the user for a number, indicate you will check to see if it is evenly divisible by 3.

# Use the modulus operator to determine if it is evenly divisible by 3.

# Print a message indicating if it is or not.

### number = input("Enter a number, and I'll let you know if it is divisible 3: ")

# number = int(number)

# if number % 3 == 0:

#     print(f"\nThe number {number} is divisable by 3.")

# else:

#     print(f"\nThe number {number} is not divisable by 3.")

# Section 2:

print('\n\nSection 2:\n\n')

# # Part (A)

print("\nPart A: \n")

# Use a while loop to print the numbers 1 through 5.

# Each number should be printed on its own line.

number = 1

while number <= 4:

    print(number)

    number += 1

print(number)

# # Part (B)

print("\nPart B: \n")

# Use a while loop to print the text a user enters.

# The loop should continue until the user enters 'quit'.

# Use a common string function to make the input case insensitive.

# Print a message indicating the loop is ending when the user enters 'quit'.

# ##prompt = "\nTell me something about yourself:"

# prompt += "\nEnter 'quit' to end the program. "

# message = ""

# while message.lower() != 'quit':

#     message = input(prompt)

#     if message.lower() != 'quit':

#         print(message)

# # Part (C)

print("\nPart C: \n")

# Use a while loop to see if a user enters a city from a list of cities.

# Make your own list of cities.

# The loop should continue until the user enters 'quit'.

# Use a common string function to make the input case insensitive.

# Print a message indicating if the city is in the list or not.

# Print a message indicating the loop is ending when the user enters 'quit'.

# Use the break keyword to end the loop when the user enters 'quit'.

# cities = ['boise', 'nampa', 'rupert', 'burley', 'paul', 'heyburn']

# prompt\_5 = '\nEnter a city to see if we have both been there: '

# prompt\_5 += '\n(Enter "quit" when you are finished.)'

# while True:

#     city = input(prompt\_5)

#     if city in cities:

#         print('That city is in the list.')

#     elif city.lower() == 'quit':

#         print('Exiting')

#         break

#     else:

#         print('This city is NOT in the list.')

# # Part (D)

print("\nPart D: \n")

# Use a while loop that continues until a user enters a string that is not in a list.

# Make your own list of strings.

# Use a common string function to make the input case insensitive.

# Print a message indicating if the string is in the list or not.

# Print a message indicating the loop is ending when the user enters a string that is not in the list.

# Use the continue keyword to skip to the next iteration when the string is in the list.

# things\_to\_buy = ['eggs', 'milk', 'bread', 'tomatos', 'limes', 'rice', 'spices']

# print(things\_to\_buy)

# simple\_shopping\_list = '\nPlease enter simple shopping item name in when collected from list:'

# simple\_shopping\_list += '\nOnce item enter is not on list program ends.\nEnter item: '

# while True:

#     bought = input(simple\_shopping\_list)

#     print('\nItem collected move to next item.')

#     if bought.lower() in things\_to\_buy:

#         print("Item in list...")

#         continue

#     else:

#         print('Done shopping!')

#         break

# # Part (E)

print("\nPart E: \n")

# Use a while loop that continues until a list is empty.

# Make your own list of user names.

# In each iteration of the loop, ask the user to keep or delete the item.

# If they choose to keep it, put it in a list called 'confirmed'.

# If they choose to delete it, do not put it in the 'confirmed' list.

# Use the pop() method to remove items from the original list.

# When the user name list is empty, print a message indicating the loop is ending.

inactive\_users = ['harry', 'dick', 'tom']

active\_users = []

message = "If you would you like to add user to active, please enter y/n"

message += "\nIf not user will remain in inactive users list\n"

print(message)

while inactive\_users:

    current\_user = inactive\_users.pop()

    keep = input((f"activating user?: {current\_user.title()} "))

    if keep.lower() == 'y':

        active\_users.append(current\_user)

        print("User now active!\n")

    elif keep.lower() == "n":

        print("User will remain inactive.\n")

    else:

        print("unknown input, skipping user.\n")

print("out of users!")

print("\nActive users:")

for active\_user in active\_users:

    print(active\_user.title())