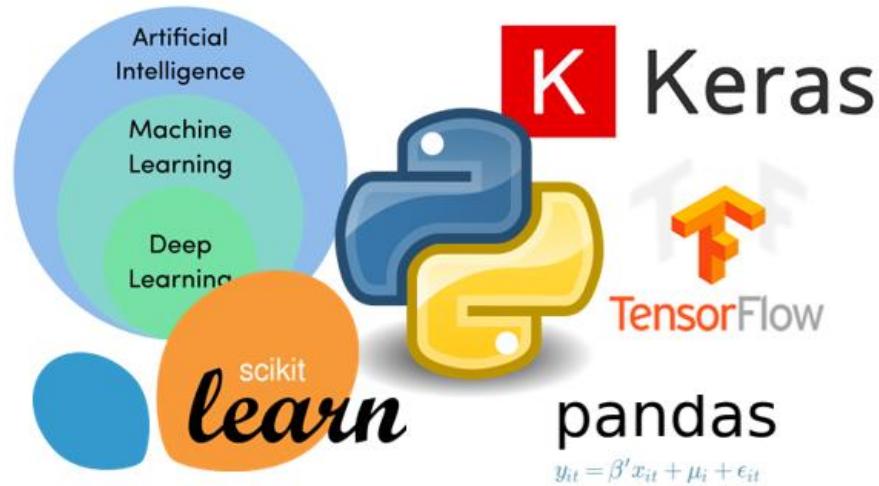


GUIDE

Exercises of Advanced Python Part 1

(Answers and explanations of how to solve the exercises)



Mentor:

- ✓ Tran Duy Thanh
- ✓ Email: thanhhtd@uel.edu.vn
- ✓ Phone: 0987773061
- ✓ Blog for self-study: <https://duythanhcse.wordpress.com>
- ✓ Facebook for solving coding problem:
<https://www.facebook.com/groups/communityuni>

-2021-

1. AUTHOR



Tran Duy Thanh was received Bachelor of Computer Science from the Industrial University of Ho Chi Minh City, and Bachelor of Business Administration from the University of Economics in Ho Chi Minh City, Vietnam.

Also, he received his M.S. degree in computer science from University of Information Technology, Vietnam National University Ho Chi Minh City (VNU-HCM), Vietnam.

He has become Full Lecturer in Faculty of Information Systems, University of Economics and Law, Vietnam National University Ho Chi Minh City (VNU)-HCM, Vietnam.

He is becoming Ph.D. Candidate of Department of Data Informatics, (National) Korea Maritime and Ocean University, Busan 49112, Republic of Korea (South Korea).

His research interests are social network analysis, big data, AI and robotics.

He received Best Paper Award the 15th International Conference on Multimedia Information Technology and Applications (MITA 2019)

He is co-author books: Basic mobile application development (ISBN: 978-604-73-5672-0), Advanced mobile application development (ISBN: 978-604-73-5673-7), Basic programming techniques (ISBN: 978-604-73-7187-7), Advanced programming techniques (ISBN: 978-604-73-7493-9).

2. BUY ME A COFFEE



Please **buy me a coffee** if you feel this document helps you



Account number: **0101146302**

Name: **Tran Duy Thanh**

Bank name: **Dong A Bank, Go Vap branch.**

3. ACKNOWLEDGMENTS

- ❖ Sincere thanks to Professors at Department of Data Science, (National) Korea Maritime and Ocean University, Busan 49112, Republic of Korea (South Korea). Exercises, documents and teaching methods and problem approach I have consulted from Professors in KMOU via Advanced Computer Engineering course, Introduction to Big Data Optimization course... I did all the exercises myself and gave detailed instructions on how to program, and applied the problem approach from the Professors, helping to increase the programming thinking for learners.
- ❖ Also, thank you to my colleagues in the Data Science Lab 407 who have supported me a lot in the past few days.

4. WHO READS?

- ❖ All students have basic knowledge of Python. Or have taken this Python course, <https://unica.vn/?aff=11929>(search Tran Duy Thanh) or click the link <https://unica.vn/lam-chu-python-trong-4-tuan?aff=11929>
- ❖ Teachers want to access new teaching methods or use them as reference materials when teaching
- ❖ Students who want to improve their knowledge of Python programming and new programming thinking methods
- ❖ The company wants to train employees.
- ❖ And those who are interested in Python.

TABLE OF CONTENTS

1.	AUTHOR.....	2
2.	BUY ME A COFFEE	3
3.	ACKNOWLEDGMENTS	4
4.	WHO READS?.....	5
5.	Exercise 1 - Automatic coffee ordering machine	20
5.1.	Learning Objectives	20
5.1.1.	Basic learning objectives	20
5.1.2.	In-depth learning objectives	20
5.1.3.	Problem situation	20
5.2.	Problem analysis	20
5.2.1.	The role of the computer.....	20
5.2.2.	User	20
5.2.3.	Input	20
5.2.4.	Output	20
5.2.5.	Data.....	21
5.3.	Full algorithm skeleton	21
5.4.	Algorithm efficiency	22
5.5.	Programming.....	22
5.5.1.	Variables	22
5.5.2.	input().....	22
5.5.3.	Programming-the whole program.....	22
5.6.	Testing and debugging	23
6.	Exercise 2 - Automatic coffee ordering machine (improve)	25
6.1.	Problem situation	25
6.2.	Programming-the whole program	25
6.3.	Testing and debugging	26
7.	Exercise 3 - Number guessing game machine.....	29
7.1.	Learning Objectives	29
7.1.1.	Basic learning objectives	29

7.1.2.	In-depth learning objectives	29
7.1.3.	Problem situation	29
7.2.	3.1. Problem analysis	29
7.2.1.	Input	29
7.2.2.	Output	29
7.2.3.	Problem decomposition	29
7.2.4.	Game rules	29
7.2.5.	Data	30
7.3.	Full algorithm skeleton	30
7.4.	Programming.....	31
7.4.1.	Variables	31
7.4.2.	input().....	31
7.4.3.	Programming-the whole program.....	31
7.5.	Testing and debugging	32
8.	Exercise 4 - Multiplication table game machine	35
8.1.	Learning Objectives	35
8.1.1.	Basic learning objectives	35
8.1.2.	In-depth learning objectives	35
8.1.3.	Problem situation	35
8.2.	Problem analysis	35
8.2.1.	Input	35
8.2.2.	Output	35
8.2.3.	Problem decomposition	35
8.2.4.	Data	35
8.2.5.	Game Rules.....	36
8.3.	Full algorithm skeleton	36
8.4.	Programming.....	37
8.4.1.	Variables	37
8.4.2.	input().....	37
8.4.3.	Programming-the whole program.....	37

8.5. Testing and debugging	38
9. Exercise 5 - Multiplication table game machine (improve)	39
9.1. Problem situation	39
9.2. Programming-the whole program.....	39
9.3. Testing and debugging	40
10. Exercise 6 - Book Search Machine	42
10.1. Learning Objectives.....	42
10.1.1. Basic learning objectives	42
10.1.2. In-depth learning objectives	42
10.1.3. Problem situation	42
10.2. Problem analysis	42
10.2.1. The role of the computer.....	42
10.2.2. User	42
10.2.3. Input	42
10.2.4. Output	42
10.2.5. Data	43
10.3. Full algorithm skeleton.....	43
10.4. Algorithm efficiency.....	44
10.5. Programming	44
10.5.1. Variables	44
10.5.2. Data Structure	44
10.5.3. Programming-the whole program.....	45
10.6. Testing and debugging.....	46
11. Exercise 7 - Book Search Machine (improve)	49
11.1. Problem situation	49
11.2. Programming-the whole program.....	49
11.3. Testing and debugging.....	51
12. Exercise 8 - Set time reminder	55
12.1. Learning Objectives.....	55
12.1.1. Basic learning objectives	55

12.1.2. In-depth learning objectives	55
12.1.3. Problem situation	55
12.2. Problem analysis	55
12.2.1. The role of the computer.....	55
12.2.2. User	55
12.2.3. Input	55
12.2.4. Output	55
12.2.5. Data	56
12.3. Full algorithm skeleton.....	56
12.4. Programming	57
12.4.1. Variables	57
12.4.2. Some utilities library.....	57
12.4.3. Programming-the whole program.....	57
12.5. Testing and debugging.....	58
13. Exercise 9 - Set time reminder (improve)	59
13.1. Problem situation	59
13.2. Programming-the whole program.....	59
13.3. Testing and debugging.....	60
14. Exercise 10 - Simple calculator.....	61
14.1. Learning Objectives.....	61
14.1.1. Basic learning objectives	61
14.1.2. In-depth learning objectives	61
14.1.3. Problem situation	61
14.2. Problem analysis	61
14.2.1. The role of the computer.....	61
14.2.2. User	61
14.2.3. Input	62
14.2.4. Output	62
14.2.5. Data.....	62
14.3. Algorithm Efficiency	62

14.4. Full algorithm skeleton.....	63
14.5. Programming	63
14.5.1. Variables	63
14.5.2. Some explanations	64
14.5.3. Programming-the whole program.....	64
14.6. Testing and debugging.....	65
15. Exercise 11 - Simple calculator (improve).....	67
15.1. Problem situation	67
15.2. Programming-the whole program.....	67
15.3. Testing and debugging.....	68
16. Exercise 12 - Pattern Quiz Game Machine	71
16.1. Learning Objectives.....	71
16.1.1. Basic learning objectives	71
16.1.2. In-depth learning objectives	71
16.1.3. Problem situation	71
16.2. Problem analysis.....	71
16.2.1. Input	71
16.2.2. Output	71
16.2.3. Problem decomposition	71
16.2.4. Data	71
16.3. Full algorithm skeleton.....	72
16.4. Programming	73
16.4.1. Variables	73
16.4.2. Some explanations	73
16.4.3. Programming-the whole program.....	74
16.5. Testing and debugging.....	74
17. Exercise 13 - Pattern Quiz Game Machine (improve)	76
17.1. Problem situation	76
17.2. Programming-the whole program.....	76
17.3. Testing and debugging.....	77

18. Exercise 14 - TOEIC Score Analyzer	80
18.1. Learning Objectives.....	80
18.1.1. Basic learning objectives	80
18.1.2. In-depth learning objectives	80
18.1.3. Problem situation	80
18.2. Problem analysis.....	80
18.2.1. Input	80
18.2.2. Output	80
18.2.3. Problem decomposition	80
18.2.4. Data	80
18.3. Full algorithm skeleton.....	81
18.4. Programming	82
18.4.1. Variables	82
18.4.2. Some explanations	82
18.4.3. Programming-the whole program.....	83
18.5. Testing and debugging.....	84
19. Exercise 15 - TOEIC Score Analyzer (improve)	85
19.1. Problem situation	85
19.2. Programming-the whole program.....	85
19.3. Testing and debugging.....	88
20. Exercise 16 - Time difference calculator	90
20.1. Learning Objectives.....	90
20.1.1. Basic learning objectives	90
20.1.2. In-depth learning objectives	90
20.1.3. Problem situation	90
20.2. Problem analysis.....	90
20.2.1. The role of the computer.....	90
20.2.2. User	90
20.2.3. Input	90
20.2.4. Output	90

20.2.5. Data	90
20.3. Full algorithm skeleton.....	91
20.4. Programming	92
20.4.1. Variables	92
20.4.2. Some explanations	92
20.4.3. Programming-the whole program.....	93
20.5. Testing and debugging.....	94
21. Exercise 17 - Time difference calculator (improve)	95
21.1. Problem situation	95
21.2. Programming-the whole program.....	95
21.3. Testing and debugging.....	98
22. Exercise 18 - Achievement processor	103
22.1. Learning Objectives.....	103
22.1.1. Basic learning objectives	103
22.1.2. In-depth learning objectives	103
22.1.3. Problem situation	103
22.2. Problem analysis	103
22.2.1. Input	103
22.2.2. Output	103
22.2.3. Problem decomposition	103
22.2.4. Data.....	103
22.3. Full algorithm skeleton.....	104
22.4. Programming	105
22.4.1. Variables	105
22.4.2. Some explanations	105
22.4.3. Programming-the whole program.....	106
22.5. Testing and debugging.....	107
23. Exercise 19 - Achievement processor (improve)	108
23.1. Problem situation	108
23.2. Programming-the whole program.....	108

23.3. Testing and debugging.....	110
24. Exercise 20 - Online shopping cart	114
24.1. Learning Objectives.....	114
24.1.1. Basic learning objectives	114
24.1.2. In-depth learning objectives	114
24.1.3. Problem situation	114
24.2. Problem analysis	114
24.2.1. The role of the computer.....	114
24.2.2. User	114
24.2.3. Input	114
24.2.4. Output	114
24.2.5. Data	114
24.3. Full algorithm skeleton.....	115
24.4. Programming	116
24.4.1. Variables	116
24.4.2. Some explanations	116
24.4.3. Programming-the whole program.....	117
24.5. Testing and debugging.....	118
25. Exercise 21 - Online shopping cart (improve)	119
25.1. Problem situation	119
25.2. Programming-the whole program.....	119
25.3. Testing and debugging.....	121
26. Exercise 22 - Word memorization helper	124
26.1. Learning Objectives.....	124
26.1.1. Basic learning objectives	124
26.1.2. In-depth learning objectives	124
26.1.3. Problem situation	124
26.2. Problem analysis	124
26.2.1. The role of the computer.....	124
26.2.2. User	125

26.2.3.	Input	125
26.2.4.	Output	125
26.2.5.	Data	125
26.3.	Algorithm Efficiency	125
26.4.	Full algorithm skeleton.....	126
26.5.	Programming	127
26.5.1.	Variables	127
26.5.2.	Some explanations	127
26.5.3.	Programming-the whole program.....	129
26.6.	Testing and debugging.....	130
27.	Exercise 23 - Word memorization helper (improve)	132
27.1.	Problem situation	132
27.2.	Programming-the whole program.....	132
27.3.	Testing and debugging.....	137
28.	Exercise 24 – Rock Paper Scissors Game	144
28.1.	Learning Objectives.....	144
28.1.1.	Basic learning objectives	144
28.1.2.	In-depth learning objectives	144
28.1.3.	Problem situation	144
28.2.	Problem analysis	144
28.2.1.	The role of the computer.....	144
28.2.2.	User	144
28.2.3.	Input	144
28.2.4.	Output	144
28.2.5.	Data	145
28.3.	Full algorithm skeleton.....	145
28.4.	Programming	146
28.4.1.	Variables	146
28.4.2.	Some explanations	146
28.4.3.	Programming-the whole program.....	147

28.5. Testing and debugging.....	149
29. Exercise 25 - Rock Paper Scissors (improve)	150
29.1. Problem situation	150
29.2. Programming-the whole program.....	150
29.3. Testing and debugging.....	153
30. Exercise 26 – Memory test game machine.....	157
30.1. Learning Objectives.....	157
30.1.1. Basic learning objectives	157
30.1.2. In-depth learning objectives	157
30.1.3. Problem situation	157
30.2. Problem analysis	157
30.2.1. The role of the computer.....	157
30.2.2. User	157
30.2.3. Input	157
30.2.4. Output	157
30.2.5. Data	157
30.3. Full algorithm skeleton.....	158
30.4. Programming	159
30.4.1. Variables	159
30.4.2. Some explanations	159
30.4.3. Programming-the whole program.....	160
30.5. Testing and debugging.....	161
31. Exercise 27 - Memory test game machine (improve)	162
31.1. Problem situation	162
31.2. Programming-the whole program.....	162
31.3. Testing and debugging.....	165
32. Exercise 28 – GUI calculator	169
32.1. Learning Objectives.....	169
32.1.1. Basic learning objectives	169
32.1.2. In-depth learning objectives	169

32.1.3. Problem situation	169
32.2. Problem analysis	169
32.2.1. The role of the computer.....	169
32.2.2. User	169
32.2.3. Input	169
32.2.4. Output	170
32.2.5. Data.....	170
32.3. Full algorithm skeleton.....	170
32.4. Programming	171
32.4.1. Variables	171
32.4.2. Some explanations	171
32.4.3. Programming-the whole program.....	172
32.5. Testing and debugging.....	173
33. Exercise 29 - GUI calculator (improve)	175
33.1. Problem situation	175
33.2. Programming-the whole program.....	175
33.3. Testing and debugging.....	176
34. Exercise 30 – Quiz Game	180
34.1. Learning Objectives.....	180
34.1.1. Basic learning objectives	180
34.1.2. In-depth learning objectives	180
34.1.3. Problem situation	180
34.2. Problem analysis	180
34.2.1. The role of the computer.....	180
34.2.2. User	180
34.2.3. Input	180
34.2.4. Output	181
34.2.5. Data.....	181
34.3. Full algorithm skeleton.....	181
34.4. Programming	182

34.4.1. Variables	182
34.4.2. Some explanations	182
34.4.3. Programming-the whole program.....	185
34.5. Testing and debugging.....	186
35. Exercise 31 - Quiz Game (improve)	188
35.1. Problem situation	188
35.2. Programming-the whole program.....	188
35.3. Testing and debugging.....	192
36. Exercise 32 – Visualize fine dust data	196
36.1. Learning Objectives.....	196
36.1.1. Basic learning objectives	196
36.1.2. In-depth learning objectives	196
36.1.3. Problem situation	196
36.2. Problem analysis	196
36.2.1. Input	196
36.2.2. Output	196
36.2.3. Problem decomposition	197
36.2.4. Thinking of users	197
36.2.5. Data.....	197
36.3. Full algorithm skeleton.....	197
36.4. Programming	198
36.4.1. Variables	198
36.4.2. Some explanations	198
36.4.3. Programming-the whole program.....	199
36.5. Testing and debugging.....	200
37. Exercise 33 - Visualize fine dust data (improve)	201
37.1. Problem situation	201
37.2. Programming-the whole program.....	201
37.3. Testing and debugging.....	203
38. Exercise 34 – Monte Carlo simulation	206

38.1.	Learning Objectives.....	206
38.1.1.	Basic learning objectives	206
38.1.2.	In-depth learning objectives	206
38.1.3.	Problem situation	206
38.2.	Problem analysis.....	206
38.2.1.	Input	206
38.2.2.	Output	206
38.2.3.	Problem decomposition	207
38.2.4.	Thinking of users	207
38.2.5.	Data	207
38.3.	Full algorithm skeleton.....	207
38.4.	Programming	208
38.4.1.	Variables	208
38.4.2.	Some explanations	208
38.4.3.	Programming-the whole program.....	211
38.5.	Testing and debugging.....	211
39.	Exercise 35 - Monte Carlo simulation (improve).....	213
39.1.	Problem situation	213
39.2.	Programming-the whole program.....	213
39.3.	Testing and debugging.....	216
40.	Exercise 36 – Numeric Recognizer – Machine learning.....	220
40.1.	Learning Objectives.....	220
40.1.1.	Basic learning objectives	220
40.1.2.	In-depth learning objectives	220
40.1.3.	Problem situation	220
40.2.	Problem analysis.....	220
40.2.1.	Input	220
40.2.2.	Output	220
40.2.3.	Problem decomposition	220
40.2.4.	Thinking of users	220

40.2.5.	Data	221
40.3.	Full algorithm skeleton	221
40.4.	Programming	222
40.4.1.	Variables	222
40.4.2.	Some explanations	222
40.4.3.	Programming-the whole program	223
40.5.	Testing and debugging	224
41.	Exercise 37 - Numeric Recognizer – Deep learning	225
41.1.	Problem situation	225
41.2.	Programming-the whole program	225
41.3.	Testing and debugging	228

5. Exercise 1 - Automatic coffee ordering machine

5.1. Learning Objectives

5.1.1. Basic learning objectives

- You can use the Input() function appropriately
- You can use the Print() function appropriately
- You can use the shorthand substitution operator +=

5.1.2. In-depth learning objectives

- Arithmetic can be performed using lists

5.1.3. Problem situation

I ordered 2 Americano cups, 2 cafe lattes, and 1 cappuccino in front of the cash coffee machine with my friends. Americano costs 2,500 won per cup, cafe latte costs 3,000 won, and cappuccino costs 3,000 won. After deciding the type of coffee and the number of cups, he invested 15,000 won in cash and received a change of 1,000 won. I wonder how the automatic coffee ordered calculates the total amount of coffee and returns the change.

5.2. Problem analysis

5.2.1. The role of the computer

To make an automatic coffee ordered with a computer, you can think of the following items

- Receive orders for the type and number of cups of coffee from users
- Calculate the total amount owed using the selected coffee information
- When receiving money from the user, calculate the change and print the result

5.2.2. User

- How users can place orders
- How to present a coffee list and amount
- How to finally present the results of the treatment

5.2.3. Input

- Number of cups for each coffee type
- The amount the user will pay

5.2.4. Output

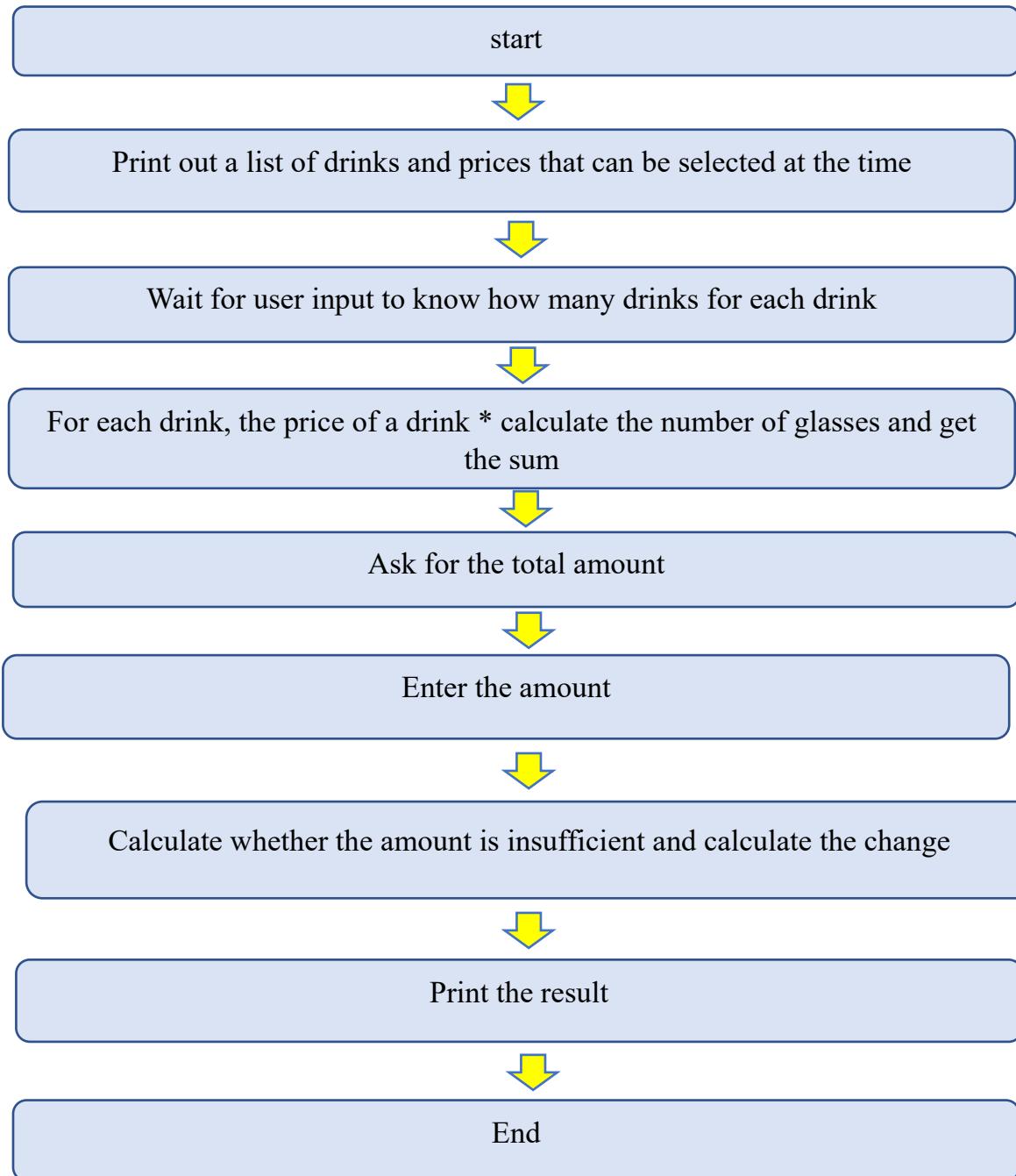
- Type of coffee and amount of cup
- Total amount ordered

- Change

5.2.5. Data

- Integer constant representing the price of each cup of coffee
- Integer variable representing the number of each cup
- Integer variable representing the total amount

5.3. Full algorithm skeleton



5.4. Algorithm efficiency

- For the efficiency of time and space, the basic algorithm for the three coffee menus is created without using repetitive statements in problem solving.
- Let's consider what the efficiency of the algorithm will be affected by using loops.

5.5. Programming

5.5.1. Variables

- noA: Number of glasses for ordered Americano
- noL: Number of cups per ordered caffe latte
- noC: Number of cups for the cappuccino ordered
- Sum: Total amount owed
- Money: Amount paid by the user

5.5.2. input()

Receives user input using, and since the received input value is a string, it is converted into an integer number by applying the int() function to perform an integer type operation

```
noA= int(input("Number of Americano (cups):"))
noL= int(input("Number of cafe lattes (cups):"))
noC= int(input("Number of cappuccinos (cups):"))
```

abbreviated substitution operator `+=`. Examples: `a = a + b` \Leftrightarrow `a += b`

Calculation of the total order amount

```
Sum = 0
Sum+=noA*2500      #sum=sum+(noA*2500)
Sum+=noL*3000      #sum=sum+(noL*3000)
Sum+=noC*3000      #sum=sum+(noC*3000)
```

5.5.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem:
https://www.facebook.com/groups/communityuni
print("Please tell me the drink you ordered.")
noA=int(input("Number of Americano (cups):"))
noL=int(input("Number of cafe lattes (cups):"))
noC=int(input("Number of cappuccinos (cups):"))

sum = 0
sum += noA*2500
sum += noL*3000
```

```
sum += noC*3000  
  
print("The total amount is:",sum," money ")  
  
money=int(input("Enter the amount to be paid >>"))  
if money <sum:  
    print("Not enough money.")  
else:  
    print("Change is",money-sum," money ")
```

5.6. Testing and debugging

Test case 1 (your money is greater than the bill).

Please tell me the drink you ordered.

Number of Americano (cups):2

Number of cafe lattes (cups):2

Number of cappuccinos (cups):1

The total amount is: 14000 money

Enter the amount to be paid >>24000

Change is 10000 money

Test case 2 (your money is less than the bill).

Please tell me the drink you ordered.

Number of Americano (cups):3

Number of cafe lattes (cups):4

Number of cappuccinos (cups):2

The total amount is: 25500 money

Enter the amount to be paid >>23540

Not enough money.

Test case 3 (your money is equal the bill).

Please tell me the drink you ordered.

Number of Americano (cups):1

Number of cafe lattes (cups):2

Number of cappuccinos (cups):3

The total amount is: 17500 money

Enter the amount to be paid >>17500

Change is 0 money

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise1.py

6. Exercise 2 - Automatic coffee ordering machine (improve)

6.1. Problem situation

Improve the Exercise 1. Let's solve the problem by using two lists and loops, assuming that more drinks and prices are stored in two lists as follows.

```
blist=['Americano','Café Latte','Cappuccino','Oranges','Colacola','Grapefruit']
```

```
plist=[2500,3000,3000,4000,1500,4000]
```

With this exercise, you should create a menu for food/drink, user can choose the food and quantity. Program will show the confirmation order list and calculate the total of money that user have to pay.

6.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem:
https://www.facebook.com/groups/communityuni
blist=['Americano', 'Café Latte', 'Cappuccino', 'Oranges
      ', 'Colacola', 'Grapefruit']
plist=[2500, 3000, 3000, 4000, 1500, 4000]
blistUserChoose=[] #the index of food name that user selected
plistUserChoose=[] #the quantity of food name that user selected
#print the menu for food/drink
print("No.", "Food Name\tPrice")
for i in range(len(blist)):
    print(i+1, '.', blist[i], '\t', plist[i])
print("Please tell me the drink you ordered.")
while True:#the loop for choosing menu food/drink from menu
    noFoodName=0
    # User has to choose from 1->n
    while noFoodName<=0 or noFoodName > len(blist):
        strnoFoodName=input("Please choose the food name(enter number) [{]-[}]: ".format(1, len(blist)))
        # If the value is a digit, we convert to int
        if strnoFoodName.isdigit():
            noFoodName=int(strnoFoodName)
        else:
            noFoodName=0 #reset 0 if the user input
    blistUserChoose.append(noFoodName)
    noQuantity=0
    while noQuantity<=0:#quantity must >0
        strnoQuantity = input("Please enter quantity[>0]: ")
        if strnoQuantity.isdigit():
            noQuantity=int(strnoQuantity)
        else:
            noQuantity=0
```

```

plistUserChoose.append(noQuantity)
# confirm continue or not
question=input("Do you want to choose another food? (yes/no) :")
# if no, finish choosing the food/drink
if question=='no':
    break
print("The list of ordered food/drink:")
sum = 0
print("Food Name\tPrice\tQuantity\tMoney")
#the loop to print all food/drink that user has chosen
for i in range(len(blistUserChoose)):
    # get the food name from user's choosing
    foodName=blist[blistUserChoose[i]-1]
    # get quantity that user bought
    quantity=plistUserChoose[i]
    # get the unit price for food/drink
    unitPrice=plist[blistUserChoose[i]-1]
    # calculate money for each food/drink
    money=quantity*unitPrice
    # sum every food/drink
    sum=sum+money
    # print detail information for each item
    print(foodName, "\t", unitPrice, "\t", quantity, "\t\t\t", money)
print("\tThe total amount is:\t", sum, " money ")
money=0
while True:#loop for user to pay money
    strmoney=input("Enter the amount to be paid >>")
    if strmoney.isdigit()==False:
        print(strmoney, " is not valid, it should be a digit numer ")
    else:
        money=int(strmoney)
        if money<0:
            print(strmoney, " is not valid number, it should be positive")
        else:
            if money < sum:
                print("Not enough money., you have to pay ",sum," money ")
            else:
                print("Paid is successfully!")
                if money>sum:
                    print("Change is", money - sum, " money")
                    break
print("Thank you so much! see you again!")

```

6.3. Testing and debugging

Here are test cases for the programming:

1-When start the program, it will show the menu as below:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books
No. Food Name    Price
1 . Americano   2500
2 . Café Latte  3000
3 . Cappuccino  3000
4 . Oranges     4000
5 . Colacola    1500
6 . Grapefruit  4000
Please tell me the drink you ordered.
Please choose the food name(enter number)[1-6]:
```

Program will ask user choose food/drink by enter the No [1-6], the number is automatic changing when the list changes.

2-User can add the item by enter 1,2,3,4,5 or 6. If the value is not in the range, the program will ask user enter value again

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books
No. Food Name    Price
1 . Americano   2500
2 . Café Latte  3000
3 . Cappuccino  3000
4 . Oranges     4000
5 . Colacola    1500
6 . Grapefruit  4000
Please tell me the drink you ordered.
Please choose the food name(enter number)[1-6]:1
Please enter quantity[>0]:-2
Please enter quantity[>0]:2
Do you want to choose another food?(yes/no):yes
Please choose the food name(enter number)[1-6]:3
Please enter quantity[>0]:1
Do you want to choose another food?(yes/no):yes
Please choose the food name(enter number)[1-6]:5
Please enter quantity[>0]:3
Do you want to choose another food?(yes/no):no
The list of ordered food/drink:
Food Name    Price    Quantity    Money
Americano    2500    2           5000
Cappuccino   3000    1           3000
Colacola     1500    3           4500
The total amount is:      12500   money
Enter the amount to be paid >>
```

User can choose many food/drink, If user enters a not valid value, program will ask the user enter value again. And when user finish choosing food/drink. The confirmation will

be shown with 4 columns: Food Nam, Price, Quantity and money; And the total money is displayed.

3-User enter the money: It must \geq the total money. There are 3 cases in here:

-money paid <money → show error, not enough money

-money paid =money → show paid successfully

-money paid >money → show paid successfully and the change for user.

```
The list of ordered food/drink:  
Food Name    Price    Quantity    Money  
Americano     2500      2          5000  
Cappuccino   3000      1          3000  
Colacola     1500      3          4500  
The total amount is: 12500 money  
Enter the amount to be paid >>10000  
Not enough money., you have to pay 12500 money  
Enter the amount to be paid >>13000  
Paid is successfully!  
Change is 500 money  
Thank you so much! see you again!
```

The picture above shows the input money that user has to pay. When user enters 10.000, it is not enough money, the program will ask user input again. After that user enters 13000, the program will show successfully and the change (500) for user.

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise2.py

7. Exercise 3 - Number guessing game machine

7.1. Learning Objectives

7.1.1. Basic learning objectives

- You can use the **break** statement appropriately
- You can use the **random** module
- You can generate a random integer using the randint() function.

7.1.2. In-depth learning objectives

- Understand the principles and differences between sequential search and binary search
- Can write binary search algorithms and programs

7.1.3. Problem situation

I want to develop a game program for guessing numbers. The game proceeds as follows. The computer generates a random number from 1 to 30 and asks the player to guess it. The player wins if he hits it within the specified number of times, otherwise he loses. If the player's guessed number is the same as the number he has, it is said to be correct, and if it is incorrect, it tells whether it is greater or less than the number he has. Decrease or increase the number of times the computer or player can unilaterally win.

7.2.3.1. Problem analysis

7.2.1. Input

- Enter the guessed number
- Enter 'YES' or 'NO' to continue the game

7.2.2. Output

- Whether the number is correct
- Whether the guessed number is large or small

7.2.3. Problem decomposition

- Generate random numbers as random numbers
- Telling if the guessed number is large or small
- Print whether it is correct
- Increase or decrease the number of limits

7.2.4. Game rules

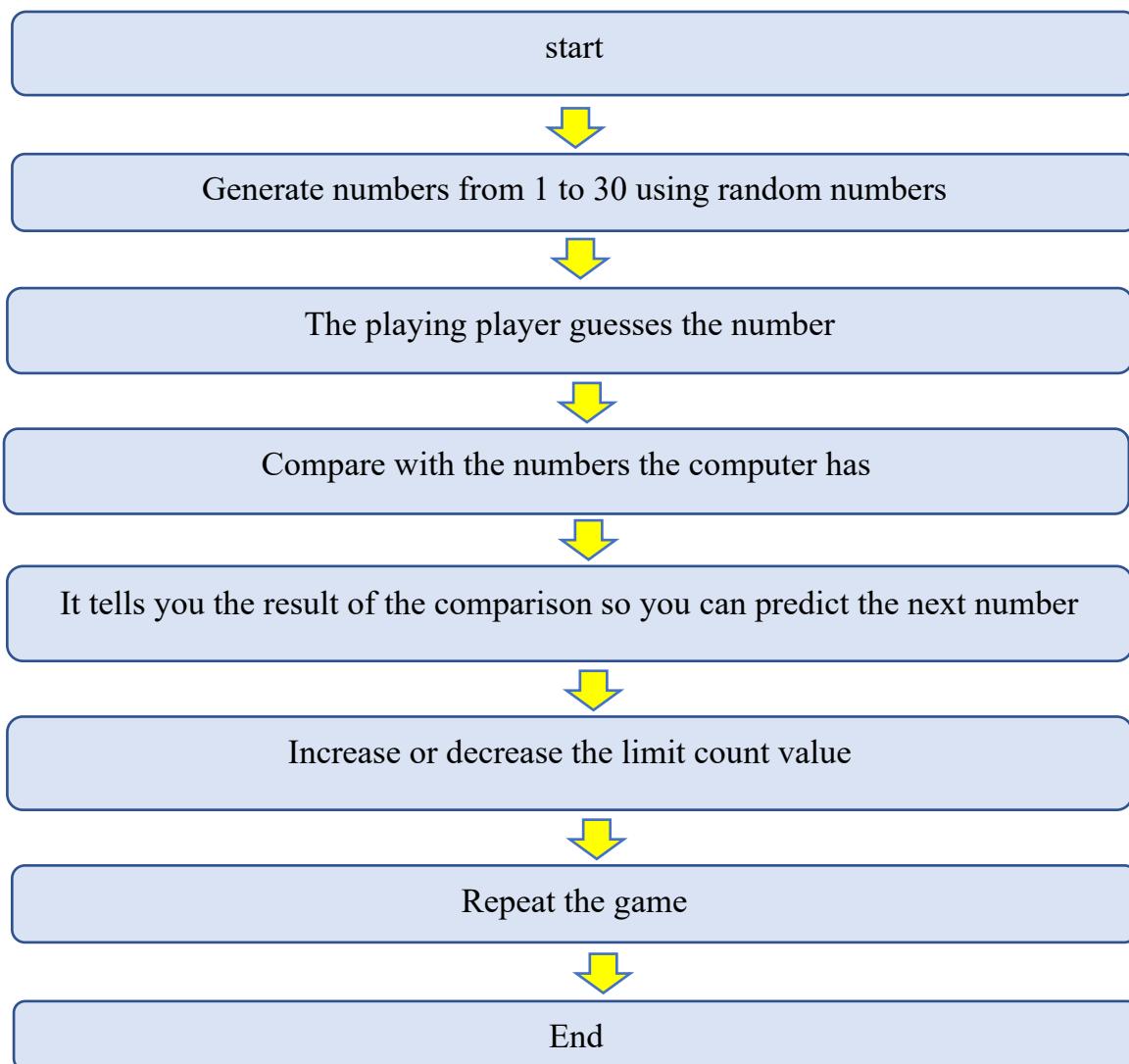
- Computer generates a random number between 1 and 30

- Player guesses a computer-generated number
- If the player's guess is wrong, the computer will tell you if it's greater or less than the generated number.
- When the game starts for the first time, the number of times limit is 5
- If it is hit within the limit, the limit is decreased by 1, and if it is incorrect, it is increased by 1.

7.2.5. Data

- An integer from 1 to 30
- Number of iterations
- String whether the game continues

7.3. Full algorithm skeleton



7.4. Programming

7.4.1. Variables

- palyerName: The name of the person playing the game
- guessNumber: Numbers generated as random numbers
- limit: Limit number of times
- count: Limit number of times
- playAgain: Whether the game continues

7.4.2. input()

The player's input-input() function

Generate random numbers-randint() function

```
#Necessary to use random numbers
import random

print("Let's play a number guessing game with me\n")
print("What's your name?\n")
#Enter player name
playerName=input()
count=1
guessNumber=-1
#Reset to limit number of 5
limit=5
#Setting the game repeat state
playAgain='YES'

ansNumber=random.randint(1,30)
```

7.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem:
https://www.facebook.com/groups/communityuni

#Necessary to use random numbers
import random

print("Let's play a number guessing game with me\n")
print("What's your name?\n")
#Enter player name
playerName=input()
count=1
guessNumber=-1
```

```

#Reset to limit number of 5
limit=5
#Setting the game repeat state
playAgain='YES'

ansNumber=random.randint(1,30)
print("Nice to meet you, "+ playerName +", I have a number between 1 and 30.
Guess it\n");

while playAgain=='YES':
    print(limit,"You only have to guess 5 times")
    while count<=limit and guessNumber!=ansNumber:
        guessNumber=int(input("Please enter the guessed number->"))
        if guessNumber==ansNumber:
            break#If you get the correct answer, exit the inner while
statement.
        elif guessNumber<ansNumber:
            print("The guessed number is less than the number with the
computer.")
        else:
            print("The guessed number is greater than the number with the
computer.")
        count=count+1

    if guessNumber == ansNumber:
        print(count, "I got it right once!! Congratulations\n")
        if limit > 1:
            limit = limit - 1
    else:
        print("The number of computers is ", ansNumber)
        limit = limit + 1

    playAgain=input("Shall we play the game again?(YES or NO)\n")
    count=1
    guessNumber=-1

```

7.5. Testing and debugging

Test case 1.

Use binary algorithms to guess the game. Examples:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

-time 1: I guess 15

The guessed number is greater than the number with the computer. So, the computer number is on the left 15.

So, the time 2, I guess 7:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

-time 2: I guess 7

The guessed number is less than the number with the computer. So, the computer number is on the right 7.

So the time 3, I guess 11:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

-time 3: I guess 11

The guessed number is greater than the number with the computer. So, the computer number is on the left 11

So, the time 4, I guess 9:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

-time 4: I guess 9

Please enter the guessed number->9.

4 I got it right once!! Congratulations

```
n: Exercise3 x
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Exercise_Par1/Exercise3.py
Let's play a number guessing game with me

What's your name?

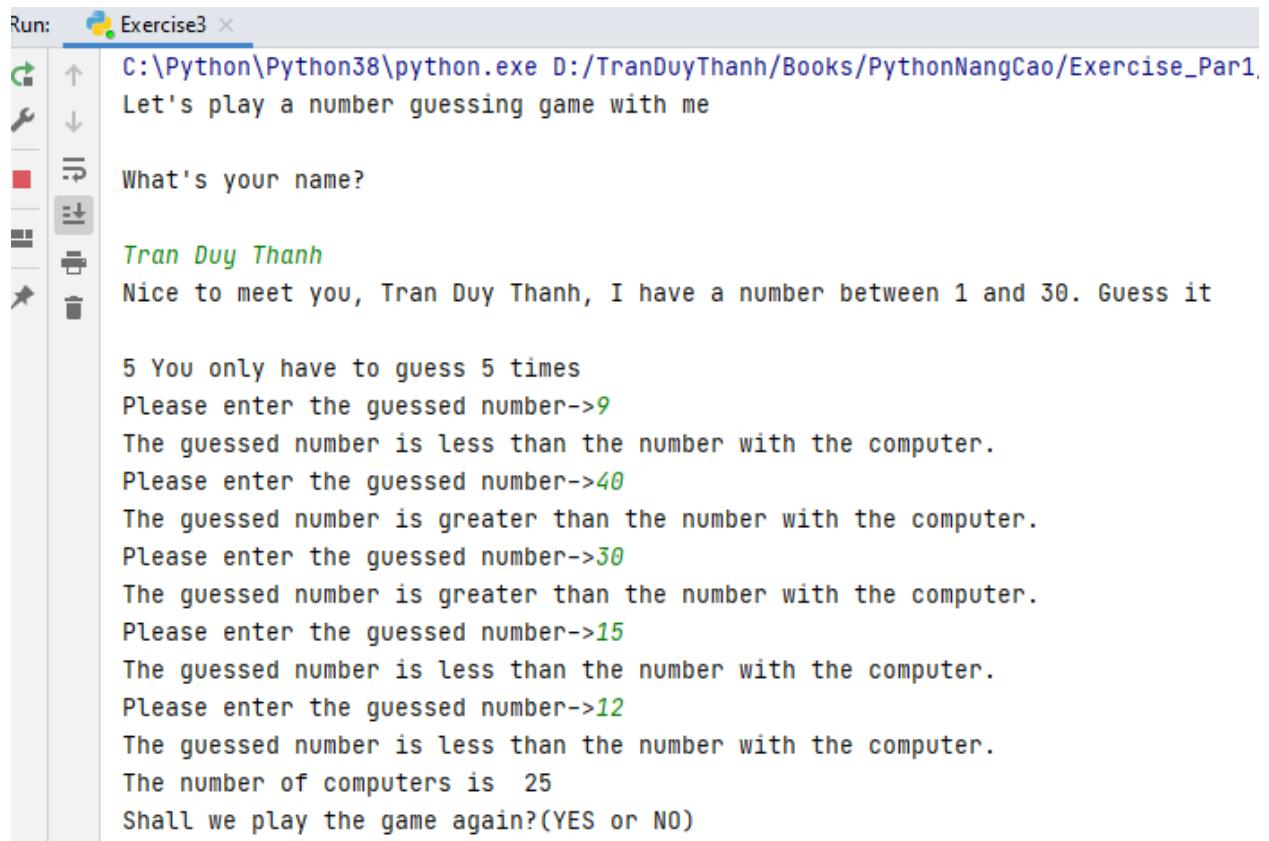
Tran Duy Thanh
Nice to meet you, Tran Duy Thanh, I have a number between 1 and 30. Guess it

5 You only have to guess 5 times
Please enter the guessed number->15
The guessed number is greater than the number with the computer.
Please enter the guessed number->7
The guessed number is less than the number with the computer.
Please enter the guessed number->11
The guessed number is greater than the number with the computer.
Please enter the guessed number->9
4 I got it right once!! Congratulations

Shall we play the game again?(YES or NO)
```

Test case 2.

Not follow the binary algorithms



The screenshot shows a Python terminal window titled "Exercise3". The command run is "C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Exercise_Par1". The terminal displays the following interaction:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Exercise_Par1
Let's play a number guessing game with me

What's your name?

Tran Duy Thanh
Nice to meet you, Tran Duy Thanh, I have a number between 1 and 30. Guess it

5 You only have to guess 5 times
Please enter the guessed number->9
The guessed number is less than the number with the computer.
Please enter the guessed number->40
The guessed number is greater than the number with the computer.
Please enter the guessed number->30
The guessed number is greater than the number with the computer.
Please enter the guessed number->15
The guessed number is less than the number with the computer.
Please enter the guessed number->12
The guessed number is less than the number with the computer.
The number of computers is 25
Shall we play the game again?(YES or NO)
```

You guess number with 5 times and not right.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise3.py

8. Exercise 4 - Multiplication table game machine

8.1. Learning Objectives

8.1.1. Basic learning objectives

- You can use the continue() statement appropriately
- You can measure time using the time() function.
- A desired random number can be generated using a loop and the Randint() function.

8.1.2. In-depth learning objectives

- Duplicated random numbers can be removed using a list

8.1.3. Problem situation

I want to play a multiplication table game that is easy in everyday life with a computer. In the game method, the computer creates a random multiplication table and asks the player to guess it within a set time. If the player hits it within the specified time, it is counted as the correct number, and if the calculation is incorrect or exceeds the time, it is regarded as invalid. Also, for the difficulty of multiplication tables, the 2nd and 5th stages are excluded.

8.2. Problem analysis

8.2.1. Input

- Number of multiplication tables
- Multiplication table problem answer

8.2.2. Output

- Multiplication table problem
- Correct and wrong
- Response time, etc

8.2.3. Problem decomposition

- Creating arbitrary multiplication tables except for the 2nd and 5th columns
- Measure response time
- Check whether the multiplication table is correct and print it out

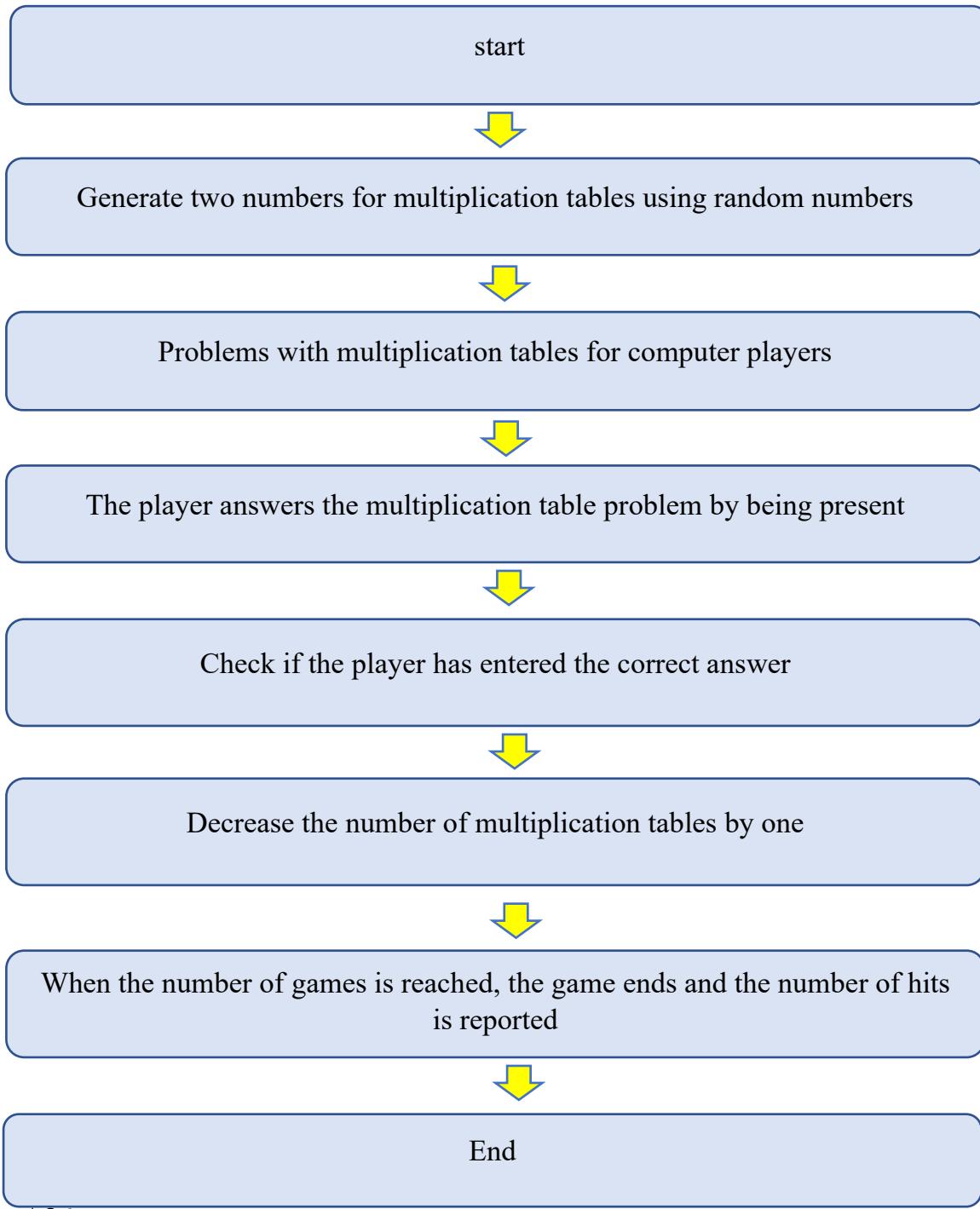
8.2.4. Data

- An integer from 1 to 10
- Multiplication table result value integer
- Integer variable representing the total amount
- Measured time

8.2.5. Game Rules

- Computer solves any multiplication table problem except for the 2nd and 5th columns
- Players solve problems within a set amount of time
- If the timeout or calculation is incorrect, the problem is regarded as incorrect
- Repeat the game up to a set number of games

8.3. Full algorithm skeleton



8.4. Programming

8.4.1. Variables

- correctAns: Correct number
- wrongAns: Wrong count
- Count: Number of games

8.4.2. input()

import random and time modules for random number and time measurements

randint() to generate random multiplication tables

When the 5th stage comes out, a random number is generated.

```
import random #Import random module
import time#Import time-related modules

correctAns=0 #Right count
wrongAns=0#Wrong count

count=int(input("How many times should I do?"))
while count!=0:
    #Generate numbers from 3rd to 9th
    a=random.randint(3,9)
    b=random.randint(3,9)
    #If it is 5th stage, random number is generated again
    if a==5 or b==5:
        continue
```

8.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

import random #Import random module
import time#Import time-related modules

correctAns=0 #Right count
wrongAns=0#Wrong count

count=int(input("How many times should I do?"))
while count!=0:
    #Generate numbers from 3rd to 9th
    a=random.randint(3,9)
    b=random.randint(3,9)
```

```
#If it is 5th stage, random number is generated again
if a==5 or b==5:
    continue

count=count-1
print("%d X %d?" %(a,b))
startTime=time.time() #Measure reaction time
product=int(input())
endTime=time.time()
print("I answered in seconds %1.f"%(endTime-startTime))

# Check if the multiplication is correct
if product==a*b:
    correctAns=correctAns+1
    print("Right!\n")
else:
    wrongAns=wrongAns+1
    print("Try again!\n")

print("total response %d, correct %d times"
% (correctAns+wrongAns,correctAns))
```

8.5. Testing and debugging

Test case with 5 cases below.

```
Exercise4 x
C:\Python\Python38\python.exe D:/TranDuyThanh
How many times should I do?5
6 X 4?
24
I answered in seconds 5
Right!

8 X 8?
64
I answered in seconds 6
Try again!

7 X 6?
42
I answered in seconds 109
Right!

3 X 4?
12
I answered in seconds 7
Right!

3 X 4?
20
I answered in seconds 6
Try again!

total response 5, correct 3 times
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise4.py

9. Exercise 5 - Multiplication table game machine (improve)

9.1. Problem situation

It was found that there were cases of asking the same multiplication table while playing the game. Let's modify it to eliminate the case of asking for the same multiplication table. Also, if you do not answer within a limited time, try to correct it so that it is regarded as wrong.

I modified the code in Exercise 4:

- Check duplication random number for multiplication
- Update the limit time about 45 second, if user answered the product is over limit time, program will set wrong answer.

9.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

# - Check duplication random number for multiplication
# - Update the limit time about 45 second, if user answered the product is over limit time,
# program will set wrong answer.
import random
import time

correctAns=0#Right count
wrongAns=0#Wrong count
#a list to check duplicate random number (for number a
duplicateListA=[]
#a list to check duplicate random number (for number b)
duplicateListB=[]
#45 second, if user answers over limitTime, the result is refused
limitTime=45
count=int(input("How many times should I do??"))
while count!=0:
    #Generate numbers from 3rd to 9th
    a=random.randint(3,9)
    b=random.randint(3,9)
    # If it is 5th stage, random number is generated again
    if a==5 or b==5:
        continue
    isDuplicated=False
    for i in range(len(duplicateListA)):#loop to check duplicate random number
        if duplicateListA[i]==a and duplicateListB[i]==b:
            isDuplicated=True
            break
```

```

if isDuplicated == True:#if the multiplication value was already asked
    continue#"Computer duplicated random number, it will random again! "
else:
    duplicateListA.append(a)
    duplicateListB.append(b)
count=count-1
print("%d X %d?" %(a,b))
#Measure reaction time
startTime=time.time()
product=int(input())
endTime=time.time()
# get answered time
calculateTime=endTime-startTime
print("I answered in seconds %1.f" % calculateTime)
if calculateTime>limitTime:
    print("You took long time to answer, limit time is ",limitTime,
          " second you answered in %1.f second"%calculateTime)
# Check if the multiplication is correct
if product==a*b and calculateTime<=limitTime:
    correctAns=correctAns+1
    print("Right!\n")
else:
    wrongAns=wrongAns+1
    print("Try again!\n")

print("total response %d, correct %d times"
      %(correctAns+wrongAns,correctAns))

```

9.3. Testing and debugging

Here are test cases for the programming:

Test case 1. (Wrong and over 45 seconds)

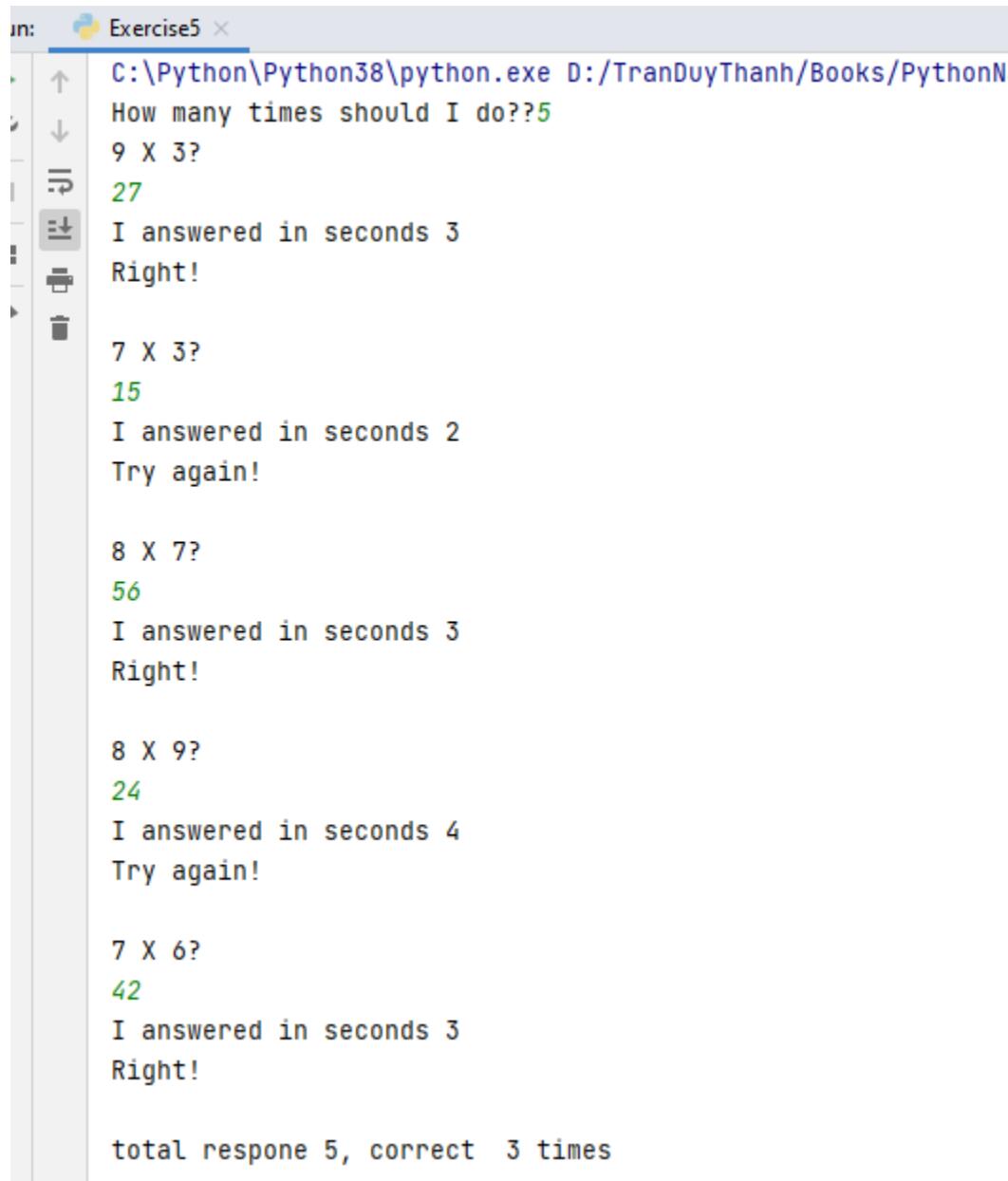
```

Exercise5 >
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Exercise_Par1/Exe>
How many times should I do??2
3 X 4?
23
I answered in seconds 9
Try again!

6 X 6?
36
I answered in seconds 151
You took long time to answer, limit time is 45 second you answered in 151 second
Try again!

total response 2, correct 0 times

```

Test case 2.

In: Exercise5

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonN
How many times should I do??5
9 X 3?
27
I answered in seconds 3
Right!

7 X 3?
15
I answered in seconds 2
Try again!

8 X 7?
56
I answered in seconds 3
Right!

8 X 9?
24
I answered in seconds 4
Try again!

7 X 6?
42
I answered in seconds 3
Right!

total response 5, correct 3 times
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise5.py

10. Exercise 6 - Book Search Machine

10.1. Learning Objectives

10.1.1. Basic learning objectives

- You can use dictionary types appropriately.
- You can use the While statement appropriately
- You can search in the dictionary list type

10.1.2. In-depth learning objectives

- You can add dictionaries to the list of dictionaries.

10.1.3. Problem situation

I have a book I want to buy, so I visited the website of an internet bookstore. If there is a book, I first searched for the book using the information of the name of the book I wanted to buy, the author, and the publisher's name, and then tried to order the book. I'm curious about how internet bookstores provide book search functionality

10.2. Problem analysis

10.2.1. The role of the computer

The computer running the Internet bookstore must do the following to provide the ability to find books:

- Receive user input, such as the name of the book to be searched or the name of the author, and keywords
- Saving book information as a dictionary
- Find the desired book in the dictionary where book information is stored.

10.2.2. User

You have to think about how to present the information so that the user can enter it easily.

- Thinking about what information to present to the user in order to receive information from the book
- Thinking about how to print out the found book information

10.2.3. Input

- Types of book search methods

10.2.4. Output

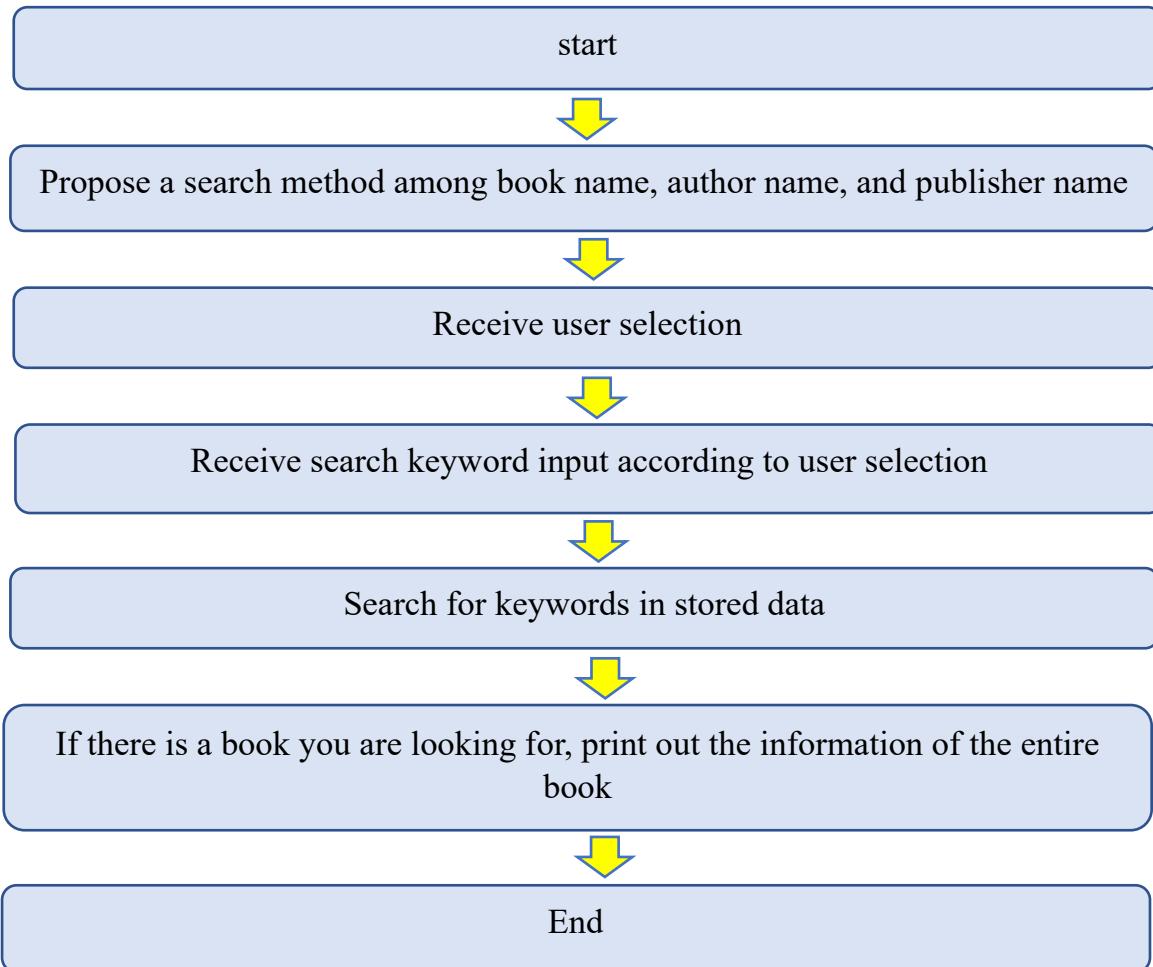
- Information of the searched book

10.2.5.Data

List of the books:

Title	Author	Publisher	Price	Publication year
Android app development	Thanh Tran	VNU	25000	2017
Python	Thanh Tran	VNU	23000	2019
JavaScript	Pham Dieu	SSS	38000	2018
HTML5	Man Nhi	HCM	33000	2012
Compiler	Thanh Tra	VNU	24000	2011
C language	Man Nhi	SSS	29000	2010
Programming linguistics	Pham Dieu	HCM	41000	2009

10.3. Full algorithm skeleton



10.4. Algorithm efficiency

- It is necessary to think of a data structure to facilitate a consistent search for different types of search conditions.

10.5. Programming

10.5.1. Variables

- mybooks: A list that stores information about several books.
- choice: variable that holds the choice value for the search method
- kwd: Variable with the key value (title, author, publisher) of the dictionary for the search method
- find: A variable that has 1 if the book to be searched is found, and 0 if it is not found.
- Onebook: A dictionary containing information on one book from mybooks

10.5.2. Data Structure

Dictionary structure for storing book information

```
mybooks=[  
    {"Title": "Android App Development", "Author": "Thanh Tran",  
     "Publisher": "VNU", "Price": "25000", "Published_Year": "2017"},  
    {"Title": "Python", "Author": "Thanh Tran",  
     "Publisher": "VNU", "Price": "23000", "Published_Year": "2019"},  
    {"Title": "JavaScript", "Author": "Pham Dieu",  
     "Publisher": "SSS", "Price": "38000", "Published_Year": "2018"},  
    {"Title": "HTML5", "Author": "Man Nhi",  
     "Publisher": "HCM", "Price": "33000", "Published_Year": "2012"},  
    {"Title": "Compiler", "Author": "Thanh Tran",  
     "Publisher": "VNU", "Price": "24000", "Published_Year": "2011"},  
    {"Title": "C language", "Author": "Man Nhi",  
     "Publisher": "SSS", "Price": "29000", "Published_Year": "2010"},  
    {"Title": "Programming Linguistics", "Author": "Pham Dieu",  
     "Publisher": "HCM", "Price": "41000", "Published_Year": "2009"},  
    {"Title": "C# language", "Author": "Thanh Tran",  
     "Publisher": "VNU", "Price": "42000", "Published_Year": "2013"},  
    {"Title": "App Inventor", "Author": "Man Nhi",  
     "Publisher": "LD", "Price": "30000", "Published_Year": "2015"},  
]
```

How to receive keyword input for book search

```
while True:#Book Search Keywords  
choice=input("Book Search Keywords  
1. Book name  
2. Author name  
3. Publisher name  
select(1,2,3):")  
if choice=='1':  
    kwd="Title"  
    break
```

```

elif choice=='2':
    kwd="Author"
    break
elif choice=='3':
    kwd="Publisher"
    break
else:
    print("Invalid input.")

```

10.5.3. Programming-the whole program

The whole program is shown as below.

```

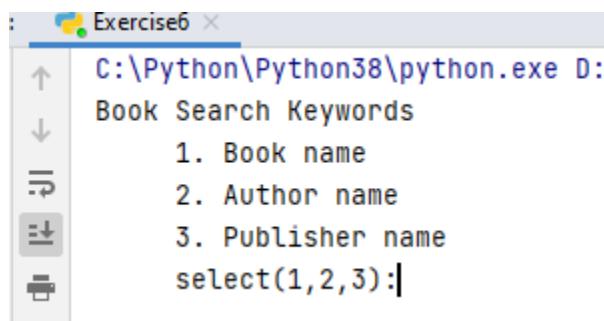
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
mybooks=[{"Title": "Android App Development", "Author": "Thanh Tran",
          "Publisher": "VNU", "Price": "25000", "Published_Year": "2017"}, {"Title": "Python", "Author": "Thanh Tran",
          "Publisher": "VNU", "Price": "23000", "Published_Year": "2019"}, {"Title": "JavaScript", "Author": "Pham Dieu",
          "Publisher": "SSS", "Price": "38000", "Published_Year": "2018"}, {"Title": "HTML5", "Author": "Man Nhi",
          "Publisher": "HCM", "Price": "33000", "Published_Year": "2012"}, {"Title": "Compiler", "Author": "Thanh Tran",
          "Publisher": "VNU", "Price": "24000", "Published_Year": "2011"}, {"Title": "C language", "Author": "Man Nhi",
          "Publisher": "SSS", "Price": "29000", "Published_Year": "2010"}, {"Title": "Programming Linguistics", "Author": "Pham Dieu",
          "Publisher": "HCM", "Price": "41000", "Published_Year": "2009"}, {"Title": "C# language", "Author": "Thanh Tran",
          "Publisher": "VNU", "Price": "42000", "Published_Year": "2013"}, {"Title": "App Inventor", "Author": "Man Nhi",
          "Publisher": "LD", "Price": "30000", "Published_Year": "2015"}, ]
while True:#Book Search Keywords
    choice=input("Book Search Keywords
        1. Book name
        2. Author name
        3. Publisher name
        select(1,2,3):")
    if choice=='1':
        kwd="Title"
        break
    elif choice=='2':
        kwd="Author"
        break
    elif choice=='3':
        kwd="Publisher"
        break

```

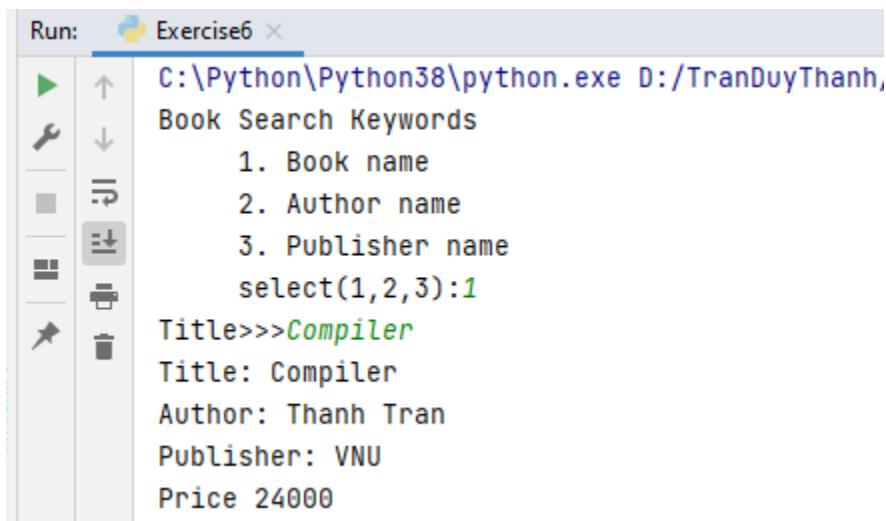
```
else:  
    print("Invalid input.")  
userin=input(kwd+">>>")  
find=False  
for onebook in mybooks:  
    if userin==onebook[kwd]:  
        print("Title:", onebook["Title"])  
        print("Author:", onebook["Author"])  
        print("Publisher:", onebook["Publisher"])  
        print("Price", onebook["Price"])  
        find=True  
if find==False:  
    print("There are no books found..")
```

10.6. Testing and debugging

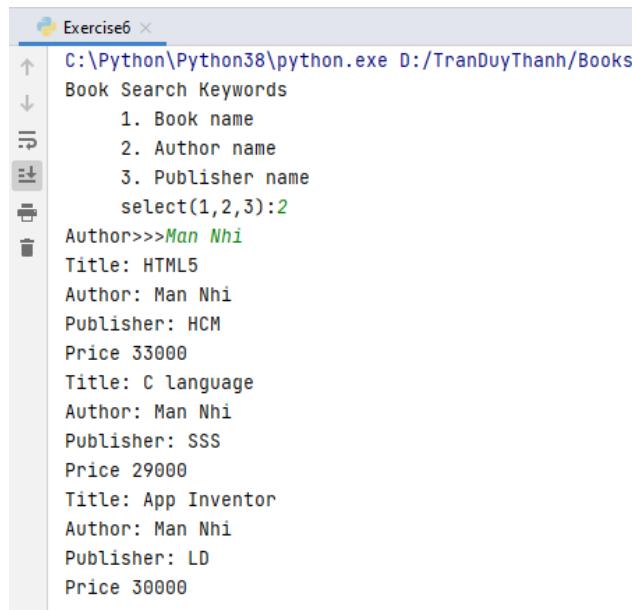
Run program:



Case 1 – search Title.

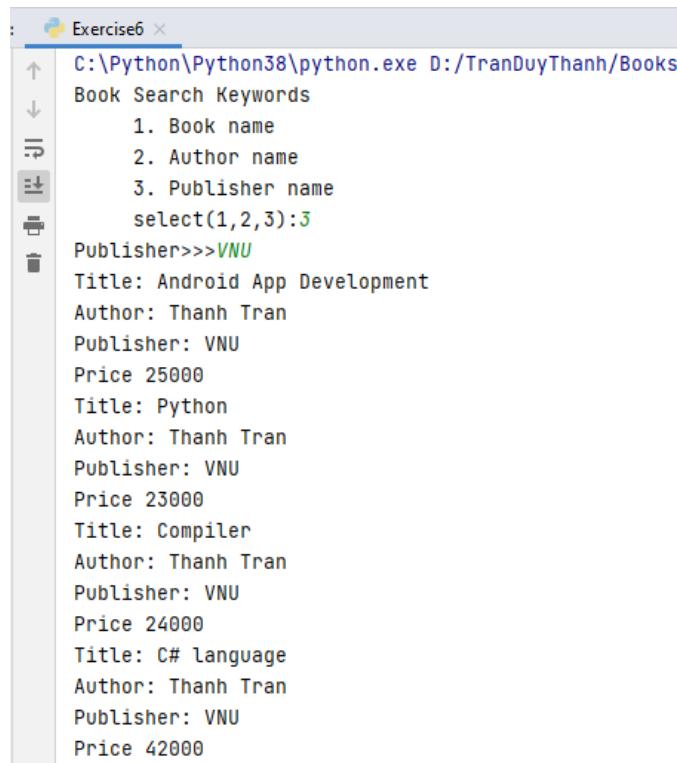


Found the book with title “Compiler”

Case 2 – search author.

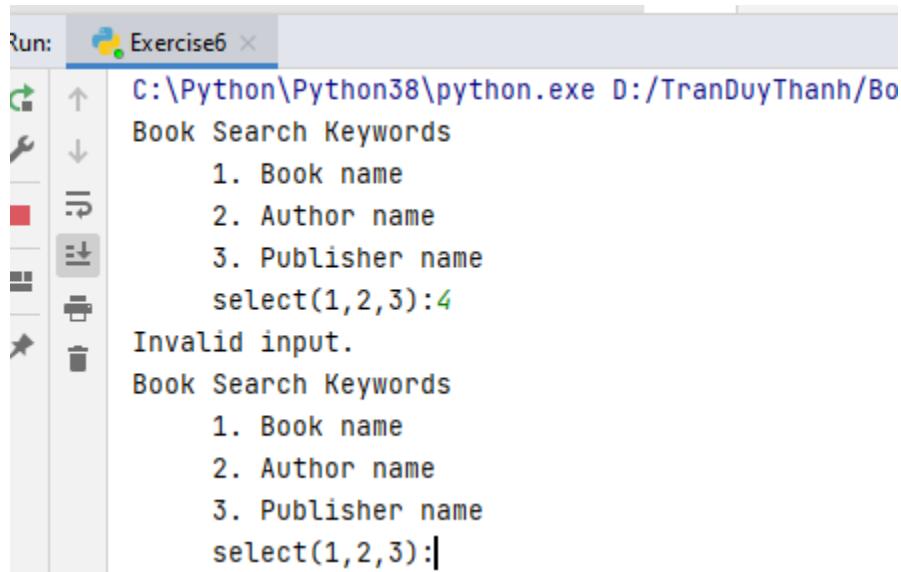
```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books
Book Search Keywords
    1. Book name
    2. Author name
    3. Publisher name
    select(1,2,3):2
Author>>>Man Nhi
Title: HTML5
Author: Man Nhi
Publisher: HCM
Price 33000
Title: C language
Author: Man Nhi
Publisher: SSS
Price 29000
Title: App Inventor
Author: Man Nhi
Publisher: LD
Price 30000
```

Enter “Man Nhi”, results with 2 books.

Case 3 – search publisher.

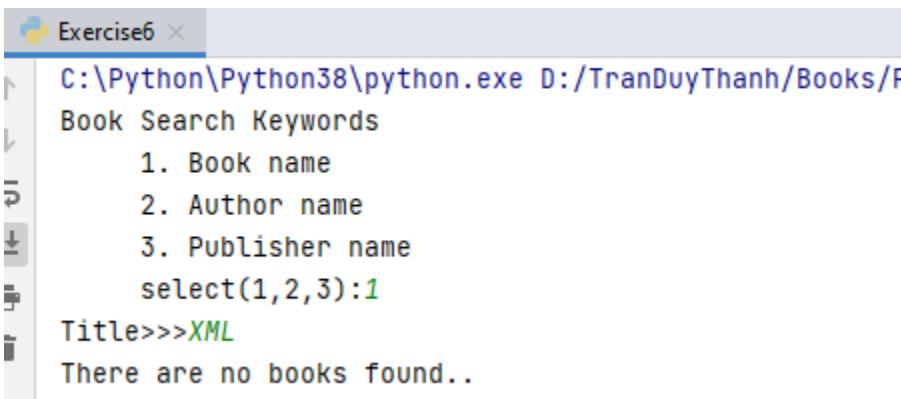
```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books
Book Search Keywords
    1. Book name
    2. Author name
    3. Publisher name
    select(1,2,3):3
Publisher>>>VNU
Title: Android App Development
Author: Thanh Tran
Publisher: VNU
Price 25000
Title: Python
Author: Thanh Tran
Publisher: VNU
Price 23000
Title: Compiler
Author: Thanh Tran
Publisher: VNU
Price 24000
Title: C# language
Author: Thanh Tran
Publisher: VNU
Price 42000
```

There are many books from “VNU” publisher.

Case 4 – invalid input.

```
Run: Exercise6
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/F
Book Search Keywords
1. Book name
2. Author name
3. Publisher name
select(1,2,3):4
Invalid input.
Book Search Keywords
1. Book name
2. Author name
3. Publisher name
select(1,2,3):|
```

Cannot find the 4th option.

Case 5 – Cannot find the books

```
Run: Exercise6
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/F
Book Search Keywords
1. Book name
2. Author name
3. Publisher name
select(1,2,3):1
Title>>>XML
There are no books found..
```

XML book can not find in the database.

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise6.py

11. Exercise 7 - Book Search Machine (improve)

11.1. Problem situation

Let's improve the program in the exercise 6 so that we can add and exit by repeatedly entering books. I modified the code in Exercise 6:

```
while True:#Book input/search
choice=input("Enter/Search for Books
1. Book input
2. Search by book name
3. Search by author name
4. Search by publisher name
5. End
select(1,2,3,4,5);")
```

If you select No. 1, you can enter title, author name, publisher name, price, and publication year as follows:

```
* title>>
* Author's name>>
* Publisher>>
* price>>
* Publication year>>
```

11.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

#Description:
# These codes I improved from exercise 6:
# 1. Book input
# 2. Search by book name
# 3. Search by author name
# 4. Search by publisher name
# 5. End
# with input a new book:
#title>>
#Author's name>>
#Publisher>>
#price>>
#Publication year>>
mybooks=[{"Title": "Android App Development", "Author": "Thanh Tran",
```

```

"Publisher": "VNU", "Price": "25000", "Published_Year": "2017"},  

{"Title": "Python", "Author": "Thanh Tran",  

 "Publisher": "VNU", "Price": "23000", "Published_Year": "2019"},  

 {"Title": "JavaScript", "Author": "Pham Dieu",  

 "Publisher": "SSS", "Price": "38000", "Published_Year": "2018"},  

 {"Title": "HTML5", "Author": "Man Nhi",  

 "Publisher": "HCM", "Price": "33000", "Published_Year": "2012"},  

 {"Title": "Compiler", "Author": "Thanh Tran",  

 "Publisher": "VNU", "Price": "24000", "Published_Year": "2011"},  

 {"Title": "C language", "Author": "Man Nhi",  

 "Publisher": "SSS", "Price": "29000", "Published_Year": "2010"},  

 {"Title": "Programming Linguistics", "Author": "Pham Dieu",  

 "Publisher": "HCM", "Price": "41000", "Published_Year": "2009"},  

 {"Title": "C# language", "Author": "Thanh Tran",  

 "Publisher": "VNU", "Price": "42000", "Published_Year": "2013"},  

 {"Title": "App Inventor", "Author": "Man Nhi",  

 "Publisher": "LD", "Price": "30000", "Published_Year": "2015"},  

]  

while True:#Book input/search  

choice=input("Enter/Search for Books  

1. Book input  

2. Search by book name  

3. Search by author name  

4. Search by publisher name  

5. End  

select(1,2,3,4,5):"  

if choice=='1':#book input  

    while True:#repeatedly enter new books  

        book=dict()  

        title=input("title>>")  

        author=input("Author's name>>")  

        publisher = input("Publisher>>")  

        price = input("price>>")  

        year = input("Publication year>>")  

        book["Title"]=title  

        book["Author"] = author  

        book["Publisher"] = publisher  

        book["Price"] = price  

        book["Published_Year"] = year  

        mybooks.append(book)  

        print("Enter a new book successfully!")  

        continueEnterBook=input("Do you want to continue enter new book or not(Y/N)?>>")  

        if continueEnterBook=='N' or continueEnterBook=='n':  

            break  

    elif choice=='2' :  

        kwd="Title"  

    elif choice=='3':  

        kwd="Author"  

    elif choice=='4':  

        kwd="Publisher"  

    elif choice=='5':  

        break;  

    else:  

        print("input is invalid.")  

# finding the book with keyword:
```

```

if choice=='2' or choice=='3' or choice=='4':
    userin=input(kwd+">>>")
    find=False
    print("Title\tAuthor\tPublisher\tPrice\tPublished Year")
    print("-"*40)
    for onebook in mybooks:
        if userin==onebook[kwd]:
            print(onebook["Title"], "\t", onebook["Author"], "\t",
                  onebook["Publisher"], "\t", onebook["Price"], "\t",
                  onebook["Published_Year"])
            find=True
        print("-" * 40)
    if find==False:
        print("No books were found.")
print("Thanks so much for your using the program!")

```

11.3. Testing and debugging

Here are some test cases when I run the program:

1-Enter a new book (press number 1):

```

In: Exercise7
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangC
Enter/Search for Books
 1. Book input
 2. Search by book name
 3. Search by author name
 4. Search by publisher name
 5. End
select(1,2,3,4,5):1
title>>BlockChain
Author's name>>Duy Thanh
Publisher>>Cloud
price>>100000
Publication year>>2021
Enter a new book successfully!
Do you want to continue enter new book or not(Y/N)?>>Y

```

When user enters a new book, Program will show and allow user enter attribute of the book as above.

And asking user want to continue enter another book or not. If user says ‘Y’ the program will continue entering, says ‘N’ the program will finish enter new book:

```

Do you want to continue enter new book or not(Y/N)?>>Y
title>>Security
Author's name>>Ho TrungThanh
Publisher>>VNU
price>>50000
Publication year>>2021
Enter a new book successfully!
Do you want to continue enter new book or not(Y/N)?>>N
Enter/Search for Books
    1. Book input
    2. Search by book name
    3. Search by author name
    4. Search by publisher name
    5. End
select(1,2,3,4,5):

```

2-Search by book name (press number 2):

```

Enter/Search for Books
    1. Book input
    2. Search by book name
    3. Search by author name
    4. Search by publisher name
    5. End
select(1,2,3,4,5):2
Title>>>BlockChain
Title      Author      Publisher      Price      Published Year
-----
BlockChain  Duy Thanh  Cloud          100000    2021
-----
```

When user press number 2, program will ask user enter the book title. If user enters “BlockChain”, it will show the details of the book that it found out.

3-Search by author name (press number 3):

```

Enter/Search for Books
    1. Book input
    2. Search by book name
    3. Search by author name
    4. Search by publisher name
    5. End
select(1,2,3,4,5):3
Author>>>Thanh Tran
Title      Author      Publisher      Price      Published Year
-----
Android App Development  Thanh Tran  VNU        25000    2017
Python     Thanh Tran  VNU        23000    2019
Compiler   Thanh Tran  VNU        24000    2011
C# language Thanh Tran  VNU        42000    2013
-----
```

User can enter the author's name to find. If program cannot find the book, it will show not found:

```
Enter/Search for Books
1. Book input
2. Search by book name
3. Search by author name
4. Search by publisher name
5. End
select(1,2,3,4,5):3
Author>>>Leo
Title      Author      Publisher      Price      Published Year
-----
-----
No books were found.
```

4-Search by publisher name (press number 4):

```
Enter/Search for Books
1. Book input
2. Search by book name
3. Search by author name
4. Search by publisher name
5. End
select(1,2,3,4,5):4
Publisher>>>VNU
Title      Author      Publisher      Price      Published Year
-----
Android App Development      Thanh Tran      VNU      25000      2017
Python      Thanh Tran      VNU      23000      2019
Compiler      Thanh Tran      VNU      24000      2011
C# language      Thanh Tran      VNU      42000      2013
Security      Ho Trung Thanh      VNU      50000      2021
-----
```

As picture above, user enters “VNU” publisher. Program will show the results that it found out.

5-End program! (press number 5). The program will stop and say thank you!:

```
Enter/Search for Books
1. Book input
2. Search by book name
3. Search by author name
4. Search by publisher name
5. End
select(1,2,3,4,5):5
Thanks so much for your using the program!
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise7.py

12. Exercise 8 - Set time reminder

12.1. Learning Objectives

12.1.1. Basic learning objectives

- You can use the time module's strftime() function to obtain a string representation of the current time in a specific format.
- You can suspend program execution for a specified time using the time.sleep() function.
- You can create a simple sound using the winsound module.

12.1.2. In-depth learning objectives

- By using the time.sleep() function, program execution can be suspended during the time difference between the alarm time and the current time.
- By using the time.strptime() function, you can calculate the time difference between the current time and the alarm time (absolute time) input in "hour:minute:second" format in seconds.

12.1.3. Problem situation

Focusing on computer work often makes it difficult for you to take care of what you need to do on time (for example, to call someone, stop working and start another one). It is possible to check the time from time to time while the work is in progress, but it is not a hassle.

12.2. Problem analysis

12.2.1. The role of the computer

- Support user to set alarm time
- Alerts the user by sounding an alarm at the set alarm time

12.2.2. User

- The current time is output on the display to help set the alarm time
- Alarm time is indicated in 'relative time'. Make it possible to type
- If an unmarked alarm time is input, the corresponding error information is output to induce the correct input of the alarm time in the future

12.2.3. Input

- Alarm time expressed in relative time in "hour:minute:second" format

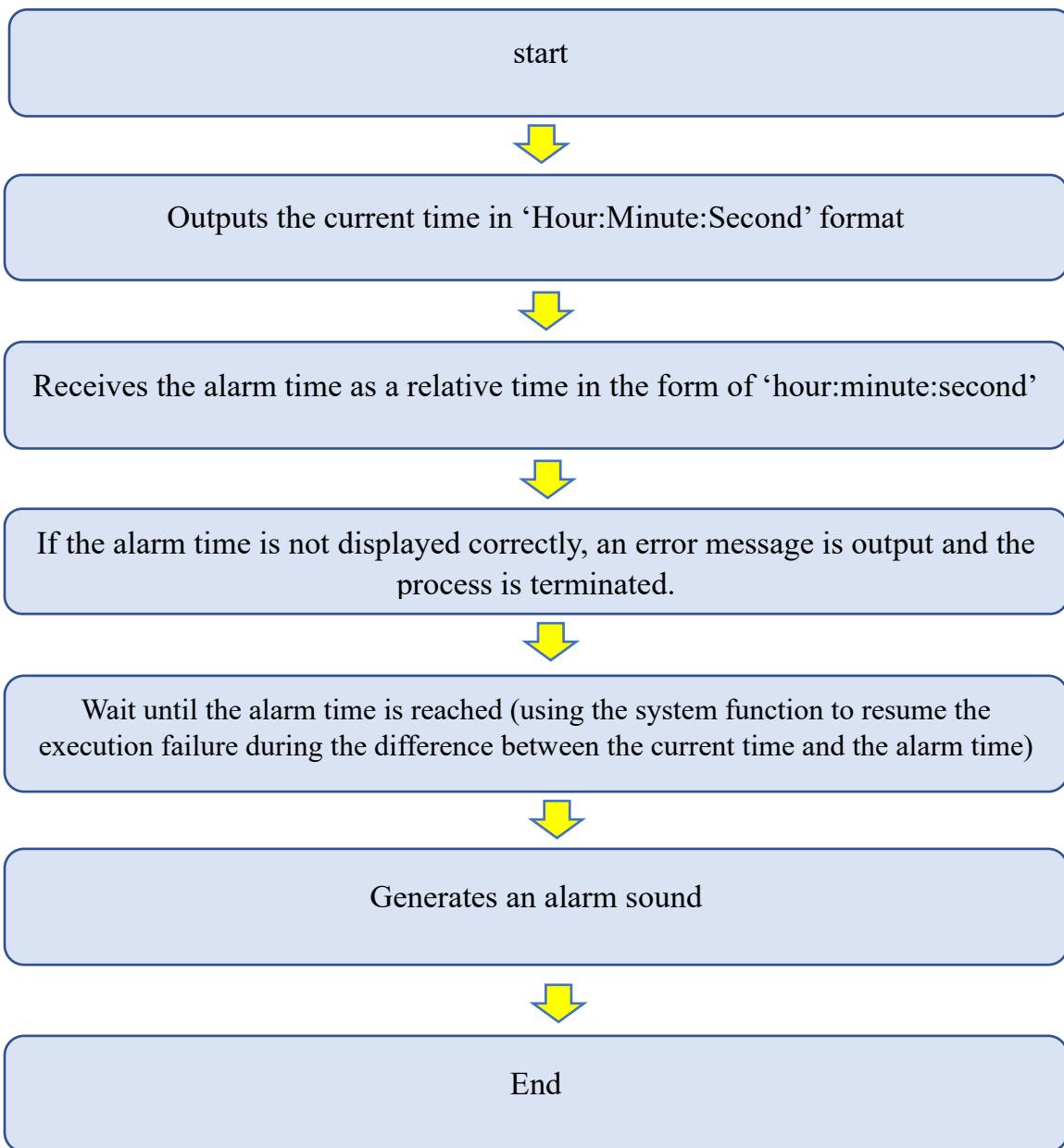
12.2.4. Output

- Input request message
- Current time
- Alarm sound
- In case of error, error message

12.2.5.Data

The time in seconds that should elapse from the current point to the alarm time

12.3. Full algorithm skeleton



12.4. Programming

12.4.1. Variables

- alarm_time: Alarm time string set by the user.
- alarm_hms: To set the alarm time. minute. A list organized by dividing by seconds

12.4.2. Some utilities library

- String creation for the current time-time module's strftime() function
- Alarm time input-input() function

```
print("current time:",time.strftime("%H:%M:%S"))
alarm_time=input("Alarm time:")
```

- Alarm time input format check-String object method split()

```
alarm_time=input("Alarm time:")
alarm_hms=alarm_time.split(":")
if len(alarm_hms)==3 and 0<=int(alarm_hms[0]) \
    and 0<=int(alarm_hms[1]) and 0<=int(alarm_hms[2]):
    pass
else:
    print("There is an error in the entered alarm time display.")
```

Hold program execution for a specified time –time module's sleep() function

Beep() function of winsound module to sound a specific frequency for a specified time

```
time.sleep(int(alarm_hms[0])* 60 * 60+ int(alarm_hms[1]) * 60 + int(alarm_hms[2]))
for i in range(1, 10):
    winsound.Beep(i * 100, 200)
```

12.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import time
import winsound
print("current time:",time.strftime("%H:%M:%S"))
alarm_time=input("Alarm time:")
alarm_hms=alarm_time.split(":")
if len(alarm_hms)==3 and 0<=int(alarm_hms[0]) \
    and 0<=int(alarm_hms[1]) and 0<=int(alarm_hms[2]):
    time.sleep(int(alarm_hms[0])* 60 * 60
               + int(alarm_hms[1]) * 60 + int(alarm_hms[2]))
    for i in range(1, 10):
        winsound.Beep(i * 100, 200)
    else:
        print("There is an error in the entered alarm time display.")
```

12.5. Testing and debugging

Run program.

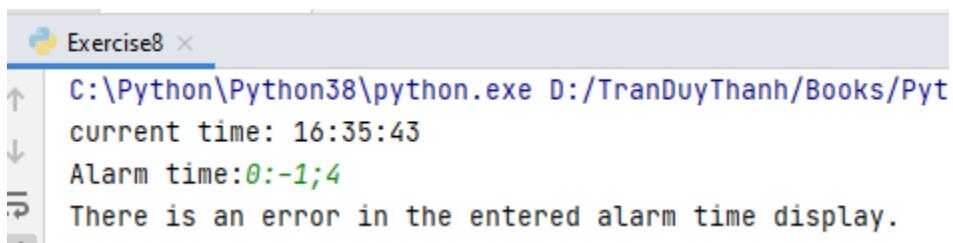
Test case 1.



```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/
current time: 16:34:09
Alarm time:0:0:10
```

After 1 second, we get the sound.

Test case 2. Enter wrong format



```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/Pyt
current time: 16:35:43
Alarm time:0:-1;4
There is an error in the entered alarm time display.
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise8.py

13. Exercise 9 - Set time reminder (improve)

13.1. Problem situation

Improve exercise 8. As the alarm time, the absolute time in the format 'hour:minute:second' (e.g., '14:45:30' for 2:45:30 pm, '7:0:25' for 7:0:25 am, and the corresponding Let's change the program to make an alarm sound at the moment

13.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import time
import winsound

print("current time:",time.strftime("%H:%M:%S"))
alarm_time=input("Alarm time:")
alarm_hms=alarm_time.split(':')

if len(alarm_hms)==3:#check date string input format
    alarmHour=int(alarm_hms[0])#get hour input
    alarmMinute = int(alarm_hms[1])#get hour minute
    alarmSecond = int(alarm_hms[2])#get hour second
    #check valid time
    #hour: 0->23
    #minute: 0->59
    #second: 0->59
    if alarmHour >= 0 and alarmHour < 24 and alarmMinute >= 0 \
        and alarmMinute<= 59 and alarmSecond >= 0 and alarmSecond <= 59:
        #need to get current time again(because the date maybe delayed by user)
        current=time.localtime()
        #get total second of current time
        totalSecondOfCurrent=current.tm_hour*60*60+current.tm_min*60+current.tm_sec
        #get total second of time to alarm
        totalSecondOfAlarm = alarmHour * 60 * 60 + alarmMinute * 60 + alarmSecond
        #get delta time between current and alarm time
        waitingSecond=totalSecondOfAlarm-totalSecondOfCurrent
        if waitingSecond>=0:#if alarm time is in the future
            time.sleep(waitingSecond)#wait waitingsecond
            for i in range(1, 10):
                winsound.Beep(i * 100, 200)#beep! beep
        else:
            print("The time is over!")#alarm is over
        else:
            print("Time is not valid!")
    else:
        print("There is an error in the entered alarm time display.")
```

13.3. Testing and debugging

Here are some test cases when I run the program:

Case 1: User sets the absolute time to get alarm:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/
current time: 16:44:02
Alarm time:16:44:15
```

Program will show beep when it meets the absolute time.

Case 2: User sets the absolute time but the time is over:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Bo
current time: 16:45:58
Alarm time:6:45:20
The time is over!
```

When time is over, program will show message “The time is over”. Because current time is “16:45:58”, but the alarm is “6:45:20”

Case 3: User sets the absolute time but the time is not valid:

```
#hour: 0->23 #minute: 0->59 #second: 0->59
C:\Python\Python38\python.exe D:/TranDuyThanh/Bo
current time: 16:46:46
Alarm time:25:60:20
Time is not valid!
```

The alarm time is not correct, hour= 25, minute = 60.. Program will show message “Time is not valid”

Case 4: User enters not correcting the format of time:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNa
current time: 16:47:13
Alarm time:14:30 36
There is an error in the entered alarm time display.
```

If user enters a not correcting the format of time, program will show “There is an error in the entered alarm time display.”

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise9.py

14. Exercise 10 - Simple calculator

14.1. Learning Objectives

14.1.1. Basic learning objectives

- Apply multiple operations using variables. can be accumulated.
- It is possible to write a program that interactively receives and processes user commands until the user makes a termination request.
- Effective representation of user commands that consist of multiple elements (eg operators, operands, etc.). You can set the format of the name so that it can be delivered

14.1.2. In-depth learning objectives

- Can explain how the structure of a program affects the scalability of the program
- Can describe how the configuration of a program's user interface can affect the interaction between the user and the program.

14.1.3. Problem situation

When you use a computer, sometimes you have to do some simple work. Every time I make a mental calculation or write it down on a piece of paper, I sometimes calculate the desired value, but if the values to be dealt with are large and the number of them increases, the accounting industry is not easy.

14.2. Problem analysis

14.2.1. The role of the computer

- Supports the user to check data values that can be referenced as cumulative variables before entering a new operation request
- Support to set the data value that can be referenced as a cumulative variable to a desired value
- Support for solving the accumulated variable as a reference to the result value of the calculation application
- Support for repetitive application of calculations until the user explicitly quits the calculator using the Calculation End command.

14.2.2. User

- All operation commands consist of one line of text each
- Support to request the application of the desired operation through the operation instruction (calculation instruction) in the form of (+, -, *, /, %)

- Support for referencing a specific value as a cumulative variable through an operation command (substitution command) in the form of “= operand”
- Support to terminate the execution of the calculator program through the "x" command (end command)
- Support for users to enter work commands while checking data values that can be referenced as cumulative variables

14.2.3. Input

- Work command in the prescribed format (calculation command, substitution command, termination command)

14.2.4. Output

- Data values that can be referenced as cumulative variables
- Input request message

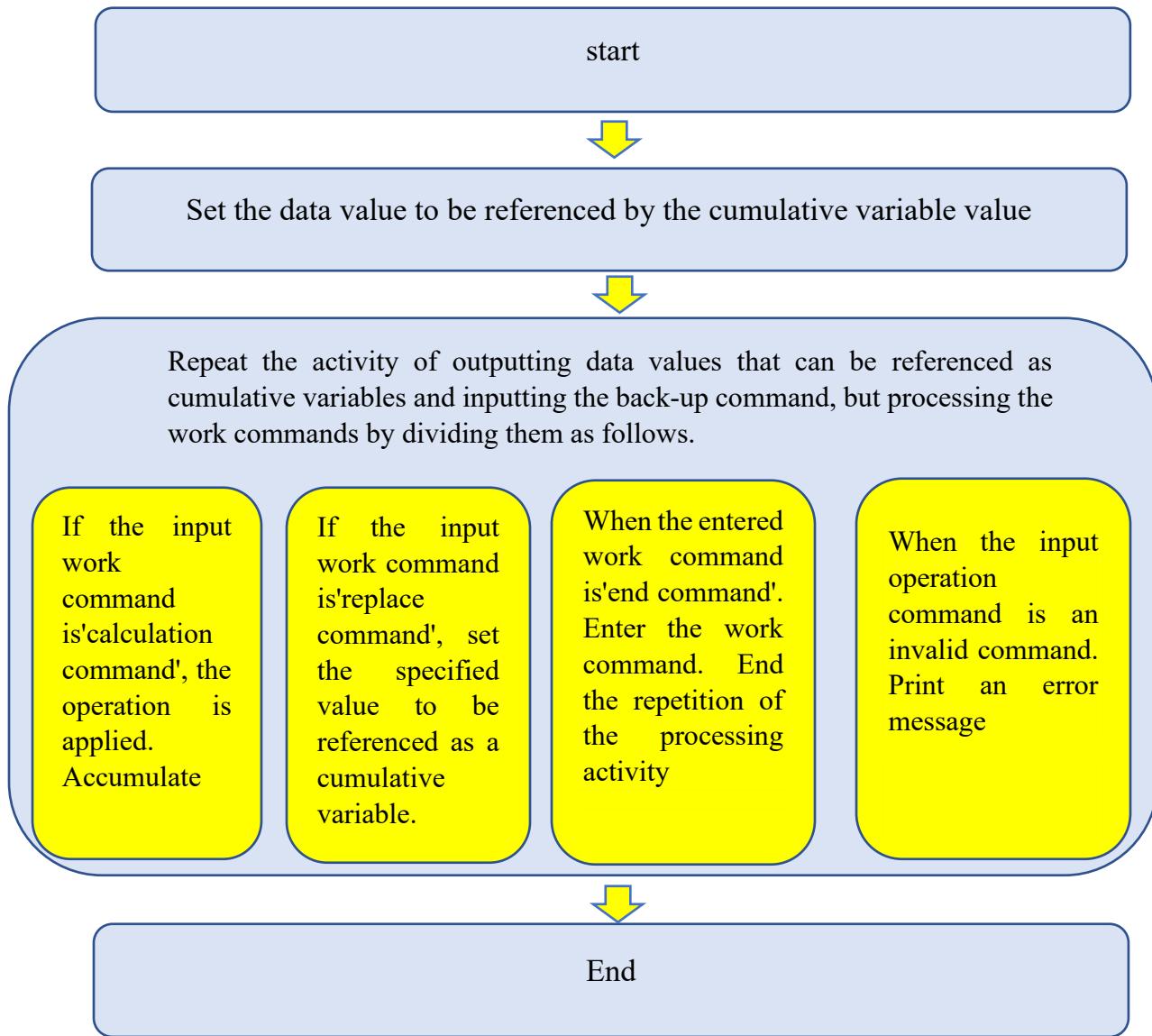
14.2.5. Data

The final result value of all operations accumulated to date

14.3. Algorithm Efficiency

- When a user requests a specific function of the calculator, the number of inputs is reduced by allowing the operator to use the operation command in the form of “operand operand” instead of having to input the operator and operand separately.
- “Operand operand... How to further reduce the number of inputs by enabling the use of “operand” type operation commands is discussed in [next exercise]
- When the user wants to check the final result of the operation applied so far, it outputs the data value that can be referenced as the cumulative number each time a work command is issued so that a specific command is not explicitly used

14.4. Full algorithm skeleton



14.5. Programming

14.5.1. Variables

- value: The final result of all calculations applied cumulatively up to the current point.
- Tokens: A list in which the components (operators and string representations of operands) are stored in order.
- operator: String corresponding to the operator among the components of the inputted each operation command
- operand: The value obtained by converting the operand string to real type among the components of the input operation command

14.5.2. Some explanations

- Accumulated variable initialization
- Infinite repetition of work instructions until user request
- Output of values referenced by accumulated variables
- User action instruction analysis - split() function

```
value=0
while True:
    print("\nCurrent value:",value)
    line=input("Enter the work command:")
    tokens=line.split()
    if len(tokens)>0:
        pass
```

- Classification processing of work instructions
- End command processing-End of infinite repetition through execution of break statement
- Convert String Value to Real Value-float() Function
- Accumulation method calculation using accumulated variables

```
operator=tokens[0]
if len(tokens) ==1:
    if operator =='x':
        break
    print("Wrong work order!!!!")
elif len(tokens)==2:
    operand=float(tokens[1])
    if operator== '=':
        value=operand
    elif operator=='+':
        value+=operand
```

Check the value of 'dividing number' to prevent division by error

```
elif operator=='/' or operator=='%':
    if operand!=0:
        if operator=='/':
            value/=operand
        else:
            value%=operand
    else:
        print("#Illegal operation command (divided
by zero)!!")
```

14.5.3. Programming-the whole program

The whole program is shown as below.

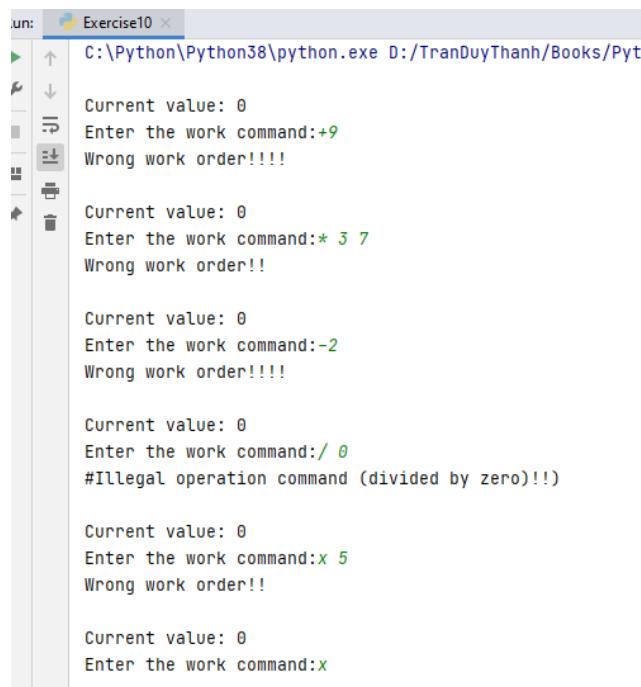
```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
value=0
while True:
    print("\nCurrent value:",value)
    line=input("Enter the work command:")
    tokens=line.split()
    if len(tokens)>0:
```

```
operator=tokens[0]
if len(tokens) ==1:
    if operator =='x':
        break
    print("Wrong work order!!!!")
elif len(tokens)==2:
    operand=float(tokens[1])
    if operator=='=':
        value=operand
    elif operator=='+':
        value+=operand
    elif operator=='*':
        value*=operand
    elif operator=='/' or operator=='%':
        if operand!=0:
            if operator=='/':
                value/=operand
            else:
                value%=operand
        else:
            print("#Illegal operation command (divided by zero)!!")
    else:
        print("Wrong work order!!")
else:
    print("Wrong work order!!")
```

14.6. Testing and debugging

Run program.

Test case 1. Test the wrong work order.



The screenshot shows a terminal window titled "Exercise10" running on Windows. It displays several test cases where the user enters invalid commands, resulting in the program printing "Wrong work order!!!!".

```
Current value: 0
Enter the work command:+9
Wrong work order!!!!

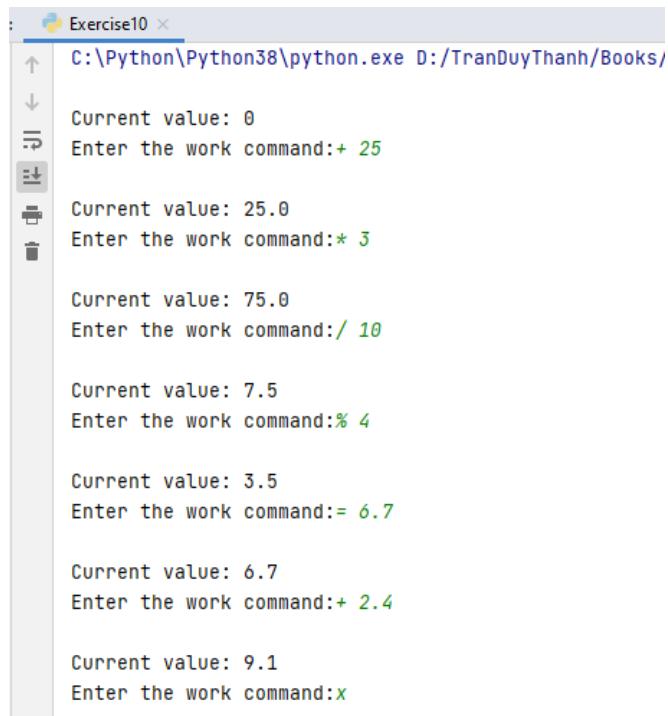
Current value: 0
Enter the work command:* 3 7
Wrong work order!!

Current value: 0
Enter the work command:-2
Wrong work order!!!!

Current value: 0
Enter the work command:/ 0
#Illegal operation command (divided by zero)!!

Current value: 0
Enter the work command:x 5
Wrong work order!!

Current value: 0
Enter the work command:x
```

Test case 2.

```
Exercise10 x
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/
Current value: 0
Enter the work command:+ 25

Current value: 25.0
Enter the work command:* 3

Current value: 75.0
Enter the work command:/ 10

Current value: 7.5
Enter the work command:% 4

Current value: 3.5
Enter the work command:= 6.7

Current value: 6.7
Enter the work command:+ 2.4

Current value: 9.1
Enter the work command:x
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise10.py

15. Exercise 11 - Simple calculator (improve)

15.1. Problem situation

Let's expand the program to support calculation instructions with two or more operands as shown below.

When the calculation command “+ 28 54 76” is applied, there are three data values that can be referenced as cumulative variables.

It should be the same as when the calculation commands “+ 28”, “+ 54”, “+ 76” are applied consecutively

15.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

value=0
while True:
    print("\nCurrent value:",value)
    line=input("Enter the work command:")
    tokens=line.split()
    if len(tokens)>0:
        operator=tokens[0]
        if len(tokens) == 1:
            if operator =='x':
                break
            print("Wrong work order!!")
        else:
            listOperand=[]#a list to store all operands
            isAllOperandValid=True
            # loop to store all operands into listOperand
            for i in range(1,len(tokens)):
                if tokens[i].isdigit() == False:
                    isAllOperandValid=False
                    break
                operand = float(tokens[i])
                listOperand.append(operand)
            if isAllOperandValid == True:
                if operator== '=':
                    # get the last operand to assign for value
                    value=listOperand[len(listOperand)-1]
                elif operator=='+':
                    # loop and plus all operand into value variable
                    for operand in listOperand:
                        value+=operand;
```

```

elif operator=='-':
    # loop and minus all operand into value variable
    for operand in listOperand:
        value-=operand;
elif operator=='*':
    # loop and multiply all operand into value variable
    for operand in listOperand:
        value*= operand;
elif operator=='^':#loop and exponential all operand into value variable
    for operand in listOperand:
        value=pow(value,operand)
elif operator=='/' or operator=='%':
    checkDividedZore=False
    for operand in listOperand: # loop to check diveded by zero
        if operand==0:
            checkDividedZore=True
            break
    if checkDividedZore==False:
        if operator=='/':
            # loop and devide all operand into value variable
            for operand in listOperand:
                value /= operand;
        else:
            # loop and mod all operand into value variable
            for operand in listOperand:
                value %= operand;
        else:
            print("Illegal operation command (divided by zero)!!")
    else:
        print("Operator is not exist![=,+,-,*,/,%,^]")
    else:
        print("Some operand is not valid!")
else:
    print("Invalid string input format")

```

15.3. Testing and debugging

Here are some test cases when I run the program:

Case 1: User enter: = 1 5 8

```

Run: Exercise11 ×
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/
Current value: 0
Enter the work command:= 1 5 8
Current value: 8.0
Enter the work command:|

```

Program will get the last number (8)

Case 2: User enter: + 4 2 7

```
Current value: 8.0
Enter the work command:+ 4 2 7

Current value: 21.0
Enter the work command:|
```

Program will plus 4, 2,7 and 8. So the value is 21

Case 3: User enter: - 1 3

```
Current value: 21.0
Enter the work command:- 1 3

Current value: 17.0
Enter the work command:|
```

Program will take $21-1 \Rightarrow 20$ and after that $20-3 \rightarrow 17$

Case 4: User enter: * 2 5

```
Current value: 17.0
Enter the work command:* 2 5

Current value: 170.0
Enter the work command:|
```

Program will take $17*2 \Rightarrow 34$ and after that $34*5 \rightarrow 170$

Case 5: User enter: / 3 5 4 2

```
Current value: 170.0
Enter the work command:/ 3 5 4 2

Current value: 1.4166666666666665
Enter the work command:|
```

The value is calculated: 1.416666

Case 6: User enter: ^ 3

```
Current value: 1.4166666666666665  
Enter the work command:^ 3
```

```
Current value: 2.8431712962962954  
Enter the work command:|
```

The value is $1.41666^3 = 2.843171$

User can enter multi value: ^ 2 5 3

Case 7: User enter: % 2

```
Current value: 2.8431712962962954  
Enter the work command:% 2
```

```
Current value: 0.8431712962962954  
Enter the work command:|
```

The value is 0.8431

Case 8: User enter not valid operator format

```
Current value: 0.8431712962962954  
Enter the work command:1 2 3  
Operator is not exist![=,+,-,*,/,%,^]
```

```
Current value: 0.8431712962962954  
Enter the work command:|
```

Case 9: User enter value with some invalid operand

```
Current value: 0.8431712962962954  
Enter the work command:+ 1 3 x 8 a 7  
Some operand is not valid!
```

```
Current value: 0.8431712962962954  
Enter the work command:|
```

The format is right, but the value of operand is not correct (it has x, a value)

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise11.py

16. Exercise 12 - Pattern Quiz Game Machine

16.1. Learning Objectives

16.1.1. Basic learning objectives

- Various patterns can be made
- Using the pattern, you can create a function that asks for and matches the pattern.

16.1.2. In-depth learning objectives

- Understand the difference between a first-order list and a two-dimensional list
- Using a two-dimensional list, we can call a function to fit a pattern

16.1.3. Problem situation

I want to play games that match various number patterns. After creating various number patterns, each pattern is displayed in turn, starting with the first number. The player infers the last number in each pattern.

16.2. Problem analysis

16.2.1. Input

- Enter the deduced pattern number

16.2.2. Output

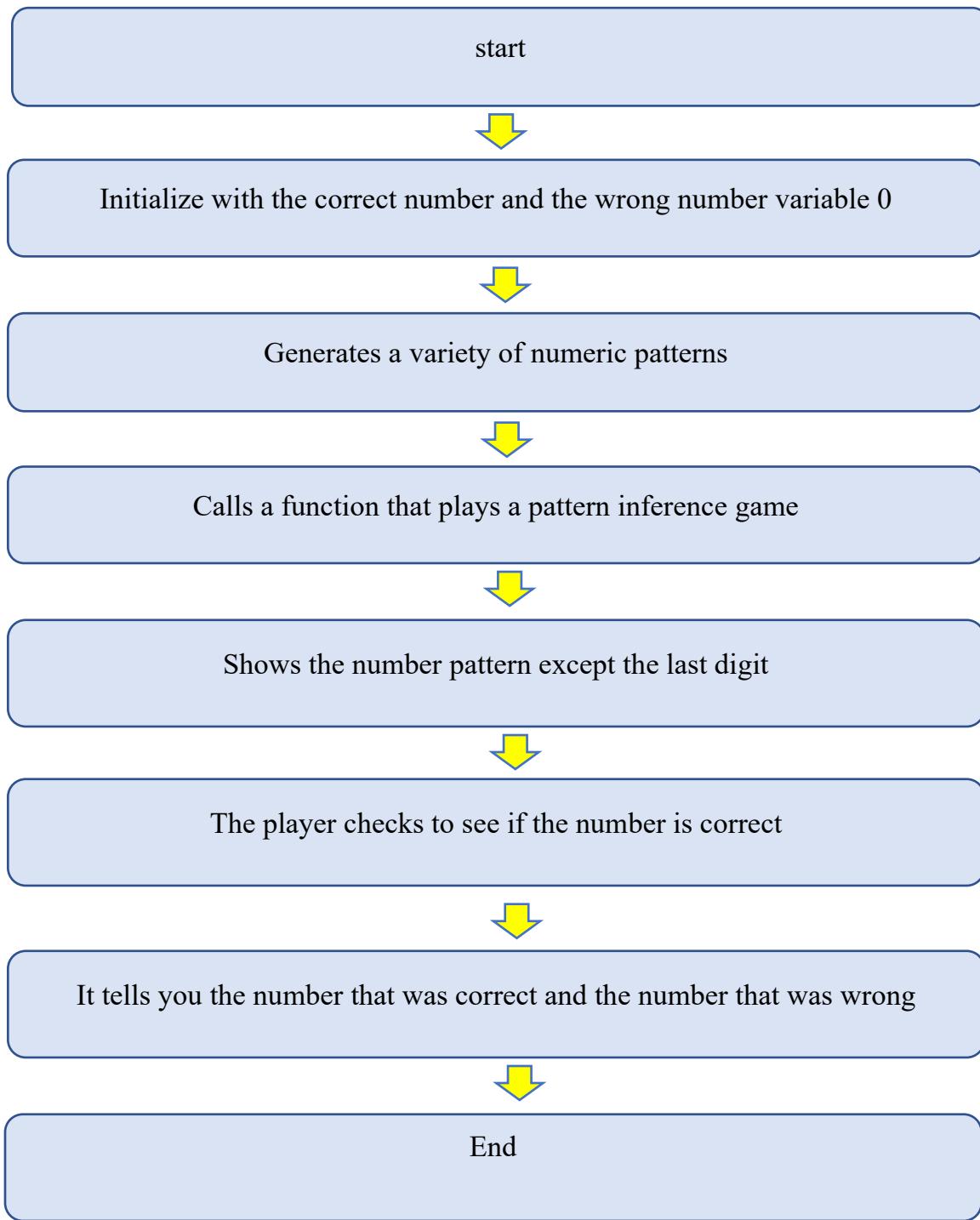
- Whether the pattern reasoning is correct
- Tell us how many hits you got

16.2.3. Problem decomposition

- Generating random number patterns
- Write a function that takes a pattern as an argument and infers a pattern
- Calling a function and passing a pattern as an argument to process it

16.2.4. Data

- Numeric list of various patterns
- Right and wrong count

16.3. Full algorithm skeleton

16.4. Programming

16.4.1. Variables

- correctAns: Number of guessed pattern reasoning.
- wrongAns: Number of hearing pattern inferences.
- Pattern1, pattern2, pattern3, pattern4, pattern5: Numeric pattern storage list

16.4.2. Some explanations

- Variable initialization
- Create numeric pattern

```
correctAns=0
wrongAns=0
#Various number patterns
pattern1=[2, 4, 6, 8]
pattern2=[13, 16, 19, 22]
pattern3=[2, 3, 5, 7, 11]
pattern4=[1, 1, 2, 3, 5, 8]
pattern5=[31, 28, 31, 30]
```

- Calling the pattern inference function
- Return correct and incorrect counts

```
pattern1=[2, 4, 6, 8]
pattern2=[13, 16, 19, 22]
pattern3=[2, 3, 5, 7, 11]
pattern4=[1, 1, 2, 3, 5, 8]
pattern5=[31, 28, 31, 30]

correctAns,wrongAns=patternmatch(pattern1, correctAns, wrongAns)
correctAns,wrongAns=patternmatch(pattern2, correctAns, wrongAns)
correctAns,wrongAns=patternmatch(pattern3, correctAns, wrongAns)
correctAns,wrongAns=patternmatch(pattern4, correctAns, wrongAns)
correctAns,wrongAns=patternmatch(pattern5, correctAns, wrongAns)

print("%d patterns were correct of %d"%(correctAns+wrongAns,correctAns))
```

The patternmatch() function that takes a pattern as an argument and makes it match the pattern.

```
def patternmatch(pattern, correctAns, wrongAns):

    for i in range(len(pattern)-1):
        print(pattern[i],end=" ")

    guessAns=int(input("What's the next number?"))

    if guessAns == pattern[len((pattern))-1]:
        correctAns = correctAns +1
        print("Well done. Congratulations")
```

```
else:  
    wrongAns=wrongAns + 1  
    print("The correct answer is %d" %(pattern[len((pattern))-1]))  
return correctAns,wrongAns
```

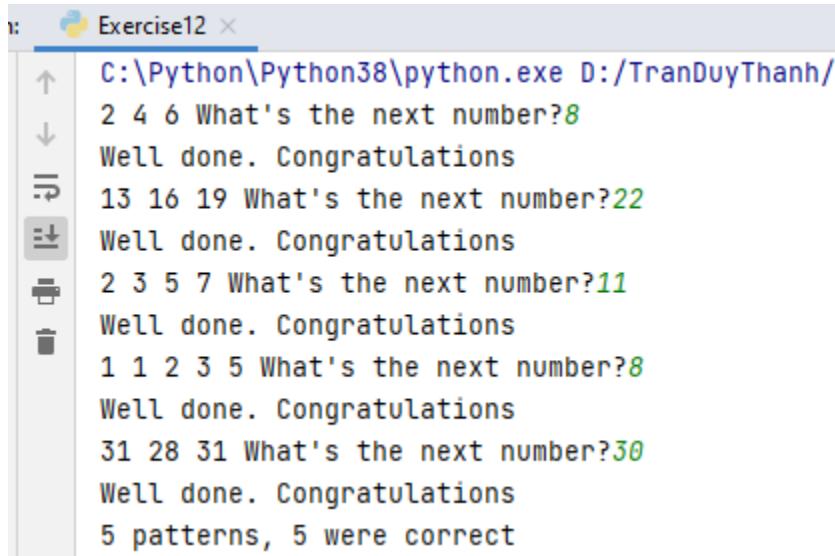
16.4.3. Programming-the whole program

The whole program is shown as below.

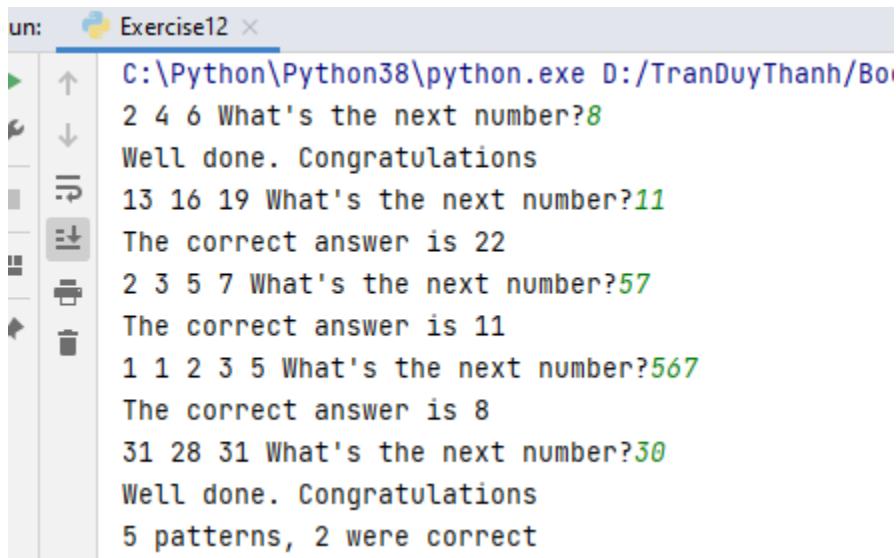
```
#Coder: Tran Duy Thanh  
#Email: thanhtd@uel.edu.vn  
#Phone: 0987773061  
#Blog for self-study: https://duythanhcse.wordpress.com  
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni  
  
def patternmatch(pattern, correctAns, wrongAns):  
  
    for i in range(len(pattern)-1):  
        print(pattern[i],end=" ")  
  
    guessAns=int(input("What's the next number?"))  
  
    if guessAns == pattern[len((pattern))-1]:  
        correctAns = correctAns +1  
        print("Well done. Congratulations")  
    else:  
        wrongAns=wrongAns + 1  
        print("The correct answer is %d" %(pattern[len((pattern))-1]))  
    return correctAns,wrongAns  
  
correctAns=0  
wrongAns=0  
#Various number patterns  
pattern1=[2, 4, 6, 8]  
pattern2=[13, 16, 19, 22]  
pattern3=[2, 3, 5, 7, 11]  
pattern4=[1, 1, 2, 3, 5, 8]  
pattern5=[31, 28, 31, 30]  
  
correctAns,wrongAns=patternmatch(pattern1, correctAns, wrongAns)  
correctAns,wrongAns=patternmatch(pattern2, correctAns, wrongAns)  
correctAns,wrongAns=patternmatch(pattern3, correctAns, wrongAns)  
correctAns,wrongAns=patternmatch(pattern4, correctAns, wrongAns)  
correctAns,wrongAns=patternmatch(pattern5, correctAns, wrongAns)  
  
print("%d patterns, %d were correct "%(correctAns+wrongAns,correctAns))
```

16.5. Testing and debugging

Run program.

Test case 1.

```
C:\Python\Python38\python.exe D:/TranDuyThanh/
2 4 6 What's the next number?8
Well done. Congratulations
13 16 19 What's the next number?22
Well done. Congratulations
2 3 5 7 What's the next number?11
Well done. Congratulations
1 1 2 3 5 What's the next number?8
Well done. Congratulations
31 28 31 What's the next number?50
Well done. Congratulations
5 patterns, 5 were correct
```

Test case 2 – some values are wrong.

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Boi
2 4 6 What's the next number?8
Well done. Congratulations
13 16 19 What's the next number?11
The correct answer is 22
2 3 5 7 What's the next number?57
The correct answer is 11
1 1 2 3 5 What's the next number?567
The correct answer is 8
31 28 31 What's the next number?50
Well done. Congratulations
5 patterns, 2 were correct
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise12.py

17. Exercise 13 - Pattern Quiz Game Machine (improve)

17.1. Problem situation

Improve Exercise 12. Since the pattern is created as a one-dimensional list, several last variables must be created, and it can be seen that there is repeatability by calling each list when calling a function. So, let's find a way to further simplify the program using a two-dimensional resource list. Also, let's modify it to ask for a number at an arbitrary position rather than the last number in the pattern

17.2. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 12.
#1.use 2d array dimension to improve program
#2.give any postion to user guess (use randomn position)
#eg: [2, 4, 6, ?], [13, ?, 19, 22], [?, 3, 5, 7, 11]...
#3.use try..exception to catch user problem input
import random

#pattern 2D array to stored all pattern
patterns=[ [2, 4, 6, 8],[13, 16, 19, 22],
           [2, 3, 5, 7, 11],[1, 1, 2, 3, 5, 8],
           [31, 28, 31, 30]
         ]
#this function use to check matching value between pattern and guessing value
#it will return True (right) or False (wrong)
def patternmatch(pattern):
    #variable to store right or wrong guessing number
    correctAns=False
    #get the len of pattern (1D array)
    lenOfPattern = len(pattern)
    #random a postion
    guessPosition = random.randint(0, lenOfPattern-1)
    #user has to guess value at this position
    valueAtQuestionMark = pattern[guessPosition]
    for i in range(lenOfPattern):
        if i == guessPosition:
            print("?", end=" ")#print ? mark, user must guess this number
        else:
            print(pattern[i],end=" ")
    while True:#user must enter valid value
        try:#use try except
            guessAns = int(input("What is the value at ? position:"))
            break
        except ValueError:
```

```

        print("Value input is not valid! please enter an integer number")
if guessAns == valueAtQuestionMark:
    correctAns = True
    print("Well done! Congratulations")
else:
    correctAns = False
    print("You are wrong, the correct answer is %d" %valueAtQuestionMark)
return correctAns

#this function pass the 2D pattern array
#it will iterate all data in 2D to ask user give guessing number
def playQuiz(patterns):
    # variable to count correct or wrong guessing number
    correctAns = 0
    wrongAns = 0
    lenOfPattern=len(patterns)
    for i in range(lenOfPattern):
        #get an 1D array pattern to reuse the patternmatch function above
        pattern=patterns[i]
        result= patternmatch(pattern)
        if result == True:
            correctAns = correctAns + 1
        else:
            wrongAns = wrongAns + 1
    print("%d patterns, correct %d, wrong %d."%(lenOfPattern,correctAns,wrongAns))
#this function will use while loop to ask user re-play Quiz Game:
def playQuizLoop():
    while True:
        print("Welcome to Quiz Game!")
        playQuiz(patterns)
        confirm=input("Do you want to re-play Quiz Game?[y,n]:")
        if confirm == 'n' or confirm == 'N':
            break
        print("Thank you so much for your playing the Quiz Game!")
#call playQuizLoop functioin:
playQuizLoop()

```

17.3. Testing and debugging

Here are some test cases when I run the program:

```

ch13_Pattern_Quiz_Game_Machine_Improved x
C:\Python\Python39\python.exe "D:/TranDuyThanh,
Welcome to Quiz Game!
2 4 ? 8 What is the value at ? position:6
Well done! Congratulations
? 16 19 22 What is the value at ? position:12
You are wrong, the correct answer is 13
2 3 5 7 ? What is the value at ? position:11
Well done! Congratulations
1 1 2 3 5 ? What is the value at ? position:8
Well done! Congratulations
31 28 ? 30 What is the value at ? position:23
You are wrong, the correct answer is 31
5 patterns, correct 3, wrong 2.
Do you want to re-play Quiz Game?[y,n]:|

```

Each pattern, program will show a random position by? mark. User has to guess this number. When finish 5 patterns, It will give the result with the total of correct and wrong answer. You see 5 patterns, correct 3, wrong 2 in the first testing.

Besides, program will ask User want to re-play the quiz game by enter y letter, let's test **second testing**:

```
Do you want to re-play Quiz Game?[y,n]:y
Welcome to Quiz Game!
2 4 ? 8 What is the value at ? position:6
Well done! Congratulations
? 16 19 22 What is the value at ? position:13
Well done! Congratulations
? 3 5 7 11 What is the value at ? position:2
Well done! Congratulations
1 1 2 3 5 ? What is the value at ? position:8
Well done! Congratulations
31 ? 31 30 What is the value at ? position:28
Well done! Congratulations
5 patterns, correct 5, wrong 0.
Do you want to re-play Quiz Game?[y,n]:
```

In the second testing, I just guessed right 5 value. In this case, the position (?) Mark is random, it is difference with the first testing.

Now, the third testing, we continue to enter 'y' to re-play Quiz Game.

```
Do you want to re-play Quiz Game?[y,n]:y
Welcome to Quiz Game!
? 4 6 8 What is the value at ? position:2
Well done! Congratulations
? 16 19 22 What is the value at ? position:4sa
Value input is not valid! please enter an integer number
What is the value at ? position:15
You are wrong, the correct answer is 13
2 3 5 ? 11 What is the value at ? position:5.5
Value input is not valid! please enter an integer number
What is the value at ? position:
```

In this testing, I enter not valid value, such as: 4as, 5.5 → program will show error as above (red rectangle). I use try... except to catch the user problem. User has to re enter valid value to continue to another pattern.

Move to the **fourth testing**, in this case we will enter 'n' to stop the program:

```
Do you want to re-play Quiz Game?[y,n]:y
Welcome to Quiz Game!
? 4 6 8 What is the value at ? position:2
Well done! Congratulations
? 16 19 22 What is the value at ? position:4sa
Value input is not valid! please enter an integer number
What is the value at ? position:15
You are wrong, the correct answer is 13
2 3 5 ? 11 What is the value at ? position:5.5
Value input is not valid! please enter an integer number
What is the value at ? position:7
Well done! Congratulations
1 1 2 ? 5 8 What is the value at ? position:3
Well done! Congratulations
31 28 ? 30 What is the value at ? position:4
You are wrong, the correct answer is 31
5 patterns, correct 3, wrong 2.
Do you want to re-play Quiz Game?[y,n]:n
Thank you so much for your playing the Quiz Game!
```

When user enter ‘n’, program will say “thank you so much for your playing the Quiz Game” and after that I will stop.

def playQuiz(patterns) function: Because in this case, patterns is a global variable so we can def playQuiz() without parameter, however this function with parameter is better than without parameter because we can pass any pattern 2D array in here to test.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise13.py

18. Exercise 14 - TOEIC Score Analyzer

18.1. Learning Objectives

18.1.1. Basic learning objectives

- Able to decompose a problem to an appropriate size
- You can create a function to find the TOEIC score frequency

18.1.2. In-depth learning objectives

- You can write an algorithm to find the minimum
- Write an algorithm to find the maximum value.

18.1.3. Problem situation

I am trying to analyze the TOEIC score to find out the English proficiency of college students. I want to know the frequency band with the most frequent frequency and the score band with the smallest frequency by calculating the frequency for each score band.

18.2. Problem analysis

18.2.1. Input

- TOEIC score provided by data in advance

18.2.2. Output

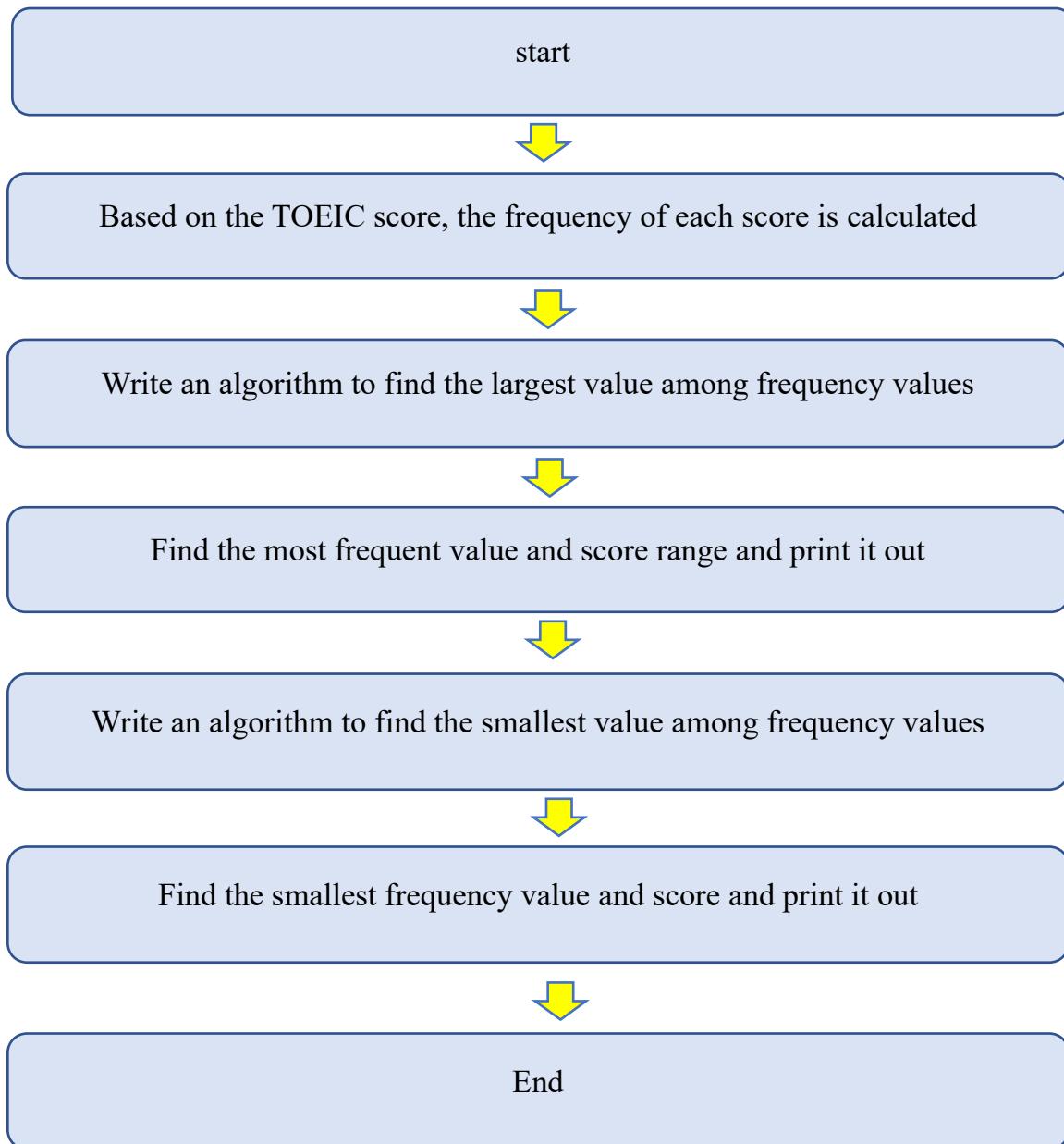
- The most frequent score band and frequency
- The least frequent score and frequency

18.2.3. Problem decomposition

- Finding the frequency of each gain score
- Find the largest frequency and score band
- Finding the least frequent and scoring bands

18.2.4. Data

- TOEIC score list
- Frequency list

18.3. Full algorithm skeleton

18.4. Programming

18.4.1. Variables

- toiecScores: TOEIC score
- counters: frequency by score
- scoreBase: score base
- maxCount: largest frequency count
- minCount: least frequent

18.4.2. Some explanations

- Finding the frequency in units of 100 points
- Distribution from 0 point to 900 point
- Divided by 100 by operator //

```
def frequency(toiecScores):
    counters=[0, 0, 0, 0, 0, 0, 0, 0, 0]
    for toiecScore in toiecScores:
        counters[toiecScore//100]+=1
    return counters
```

- The following function finds the most frequent
- The index of the list is multiplied by 100 to determine the TOEIC score.

```
def max_frequency(counters):
    max=0
    scoreBase=0
    N=len(counters)
    for i in range(N):
        if max<counters[i]:
            max=counters[i]
            scoreBase=i * 100
    return scoreBase,max
```

- The following function finds the smallest frequency
- Initialize the min value to 11

```
def min_frequency(counters):
    scoreBase=0
    N = len(counters)
    min=11
    for i in range(N):
        if counters[i] !=0 and min >counters[i]:
            scoreBase = i * 100
            min = counters[i]
    return scoreBase, min
```

- TOEIC score is given as a list
- Call frequency() function to find frequency
- Find the largest frequency and score

- Find the smallest frequency and score

```
toiecScores=[510,630,750,780,620,805,930,650,840,670]

counters =frequency(toiecScores)
scoreBase,maxCount=max_frequency(counters)

print("#Most score band=%d, frequency=%d" % (scoreBase,maxCount))

scoreBase,minCount=min_frequency(counters)

print("#Least score=%d, frequency=%d" %(scoreBase,minCount))
```

18.4.3.Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
def frequency(toiecScores):
    counters=[0, 0, 0, 0, 0, 0, 0, 0, 0]
    for toiecScore in toiecScores:
        counters[toiecScore//100]+=1
    return counters

def max_frequency(counters):
    max=0
    scoreBase=0
    N=len(counters)
    for i in range(N):
        if max<counters[i]:
            max=counters[i]
            scoreBase=i * 100
    return scoreBase,max

def min_frequency(counters):
    scoreBase=0
    N = len(counters)
    min=11
    for i in range(N):
        if counters[i] !=0 and min >counters[i]:
            scoreBase = i * 100
            min = counters[i]
    return scoreBase, min

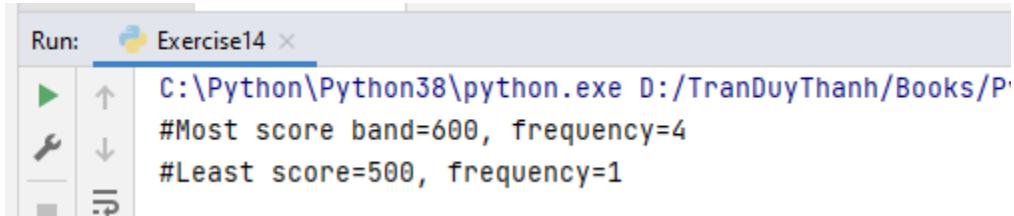
toiecScores=[510,630,750,780,620,805,930,650,840,670]

counters =frequency(toiecScores)
scoreBase,maxCount=max_frequency(counters)
```

```
print("#Most score band=%d, frequency=%d" % (scoreBase,maxCount))  
scoreBase,minCount=min_frequency(counters)  
print("#Least score=%d, frequency=%d" %(scoreBase,minCount))
```

18.5. Testing and debugging

Run program.



The screenshot shows a terminal window titled "Run: Exercise14". The command entered is "C:\Python\Python38\python.exe D:/TranDuyThanh/Books/P". The output displayed is:
#Most score band=600, frequency=4
#Least score=500, frequency=1

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise14.py

19. Exercise 15 - TOEIC Score Analyzer (improve)

19.1. Problem situation

Improve Exercise 14. There may be cases where the frequency of each group is the same. Let's modify it so that we can see all the scores with the same frequency. Also, let's find the average by adding a function to find the average of the TOEIC students acquired.

19.2. Programming-the whole program

These codes I improved from exercise 14:

- 1. List all Toiec Score for each Student acquired: Score-Frequency- Frequency rate
- 2. List all Score with range complex for Student acquired: Base Score-Frequency- Frequency rate
- 3. See all the scores with the same max frequency.
- 4. See all the scores with the same min frequency.
- 5. A function to find the average of the TOEIC students acquired.

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 14.
#1.List all Toiec Score for each Student acquired: Score-Frequency- Frequency rate
#2.List all Score with range complex for Student acquired: Base Score-Frequency- Frequency rate
#3.See all the scores with the same max frequency.
#4.See all the scores with the same min frequency.
#5.A function to find the average of the TOEIC students acquired.
#initialize data for Toiec Score that Student acquired
toiecScores=[510,630,750,910,620,805,930,650,840,670,510,780,990,990,720,510,780,200,170,400,250,256,750,915]
#this function to calcuate the Frequency, store value in an array
def frequency(toiecScores):
    counters=[0]*10
    for toiecScore in toiecScores:
        counters[toiecScore//100]+=1
    return counters
#this function use to reduce and sort the score
# (for print all score student acquired: Score, frequency,rate)
def reduceAndSortScores(toiecScores):
    scoreList=[]
    for score in toiecScores:
        if scoreList.__contains__(score) == False:
            scoreList.append(score)
    scoreList.sort()
    return scoreList
```

```
#This function use to count the frequency of each Score that Student acquired
def countFrequency(toiecScores,toiecScoreCheck):
    count=0
    for toiecScore in toiecScores:
        if toiecScoreCheck==toiecScore:
            count = count +1
    return count

#this function print all score of Student and calculate the Frequency and Rate:
def frequencyIndividual(toiecReduceAndSortScores):
    print("Score\tFre\tFre. Rate")
    lenOfToiecScores=len(toiecScores)
    for toiecScoreCheck in toiecReduceAndSortScores:
        fre=countFrequency(toiecScores,toiecScoreCheck)
        rate=round((fre/lenOfToiecScores)*100,2)
        print (toiecScoreCheck," \t",fre," \t",rate,"%")

#this function use to show the Frequency and the Rate with complex range:
def frequencyClassified(toiecScores):
    counters =frequency(toiecScores)
    print("\tScore\t\tFre\t\tFre. Rate")
    for i in range(len(counters)):
        if counters[i]!=0:
            baseScore=i*100
            rate=round((counters[i]/len(toiecScores))*100,2)
            rightRange=baseScore+99
            if rightRange >990 :
                rightRange=990
            print("[",baseScore, "-",rightRange, "]", "\t",counters[i], "\t\t",rate,"%")

#this function use to find a max value in an array
def findMaxValue(counters):
    max = 0
    for i in counters:
        if max<i:
            max=i
    return max

#this function get all max frequency that the same value
def max_frequency_all(toiecScores):
    counters = frequency(toiecScores)
    maxFrequency=findMaxValue(counters)
    list_scoreBase=[] #the list to store all scorebase that the same max frequency
    N=len(counters)
    for i in range(N):
        if maxFrequency==counters[i]:#if the same max value, we store in to the list
            list_scoreBase.append(i * 100)
    return list_scoreBase,maxFrequency

#this function gets the value: list_scoreBase,maxFrequency from max_frequency_all function and print data out
def printAllMaxFrequency():
    list_scoreBase,maxFrequency=max_frequency_all(toiecScores)
    print("Max Frequency = ",maxFrequency)
    print("The range of the Scores base have the same max frequency:")
    print("Score\tFre\t\tFre. Rate")
```

```

rate = round((maxFrequency / len(toiecScores)) * 100, 2)
for baseScore in list_scoreBase:
    print(baseScore, "\t", maxFrequency, "\t\t", rate, "%")

#this function use to find a min value in an array
def minValue(counters):
    min = findMaxValue(counters)
    for i in counters:
        if i!=0 and i<min:
            min=i
    return min

#this function get all min frequency that the same value
def min_frequency_all(toiecScores):
    counters = frequency(toiecScores)
    minFrequency=findMinValue(counters)
    list_scoreBase=[] #the list to store all scorebase that the same min frequency
    N=len(counters)
    for i in range(N):
        if minFrequency==counters[i]:#if the same max value, we store in to the list
            list_scoreBase.append(i * 100)
    return list_scoreBase,minFrequency

#this function gets the value: list_scoreBase,minFrequency from min_frequency_all function and print data out
def printAllMinFrequency():
    list_scoreBase,minFrequency=min_frequency_all(toiecScores)
    print("Min Frequency =",minFrequency)
    print("The range of the Scores base have the same min frequency:")
    print("Score\tFre\t\tFre. Rate")
    rate = round((minFrequency / len(toiecScores)) * 100, 2)
    for baseScore in list_scoreBase:
        print(baseScore, "\t", minFrequency, "\t\t", rate, "%")

#this function use to calculate the Average Toiec Score that Student Acquired
def averageOfToiecScore(toiecScores):
    avg=sum(toiecScores)/len(toiecScores)
    return int(round(avg,0))

#test and debug functions:
# 1.List all Toiec Score for each Student acquired: Score-Frequency- Frequency rate

toiecReduceAndSortScores=reduceAndSortScores(toiecScores)
frequencyIndividual(toiecReduceAndSortScores)

# 2.List all Score with range comple for Student acquired: Base Score-Frequency- Frequency rate
frequencyClassified(toiecScores)

# 3.See all the scores with the same max frequency.
printAllMaxFrequency()

# 4.See all the scores with the same min frequency.
printAllMinFrequency()

# 5.A function to find the average of the TOEIC students acquired.
print("Average Toeic Score that Student acquired=",averageOfToiecScore(toiecScores))

```

19.3. Testing and debugging

Here are some test cases when I run the program:

Case 1: List all Toiec Score for each Student acquired: Score-Frequency- Frequency rate.
In this case I call 2 statement:

```
toiecReduceAndSortScores=reduceAndSortScores(toiecScores)
frequencyIndividual(toiecReduceAndSortScores)
```

The result is here:

Score	Fre	Fre. Rate
170	1	4.17 %
200	1	4.17 %
250	1	4.17 %
256	1	4.17 %
400	1	4.17 %
510	3	12.5 %
620	1	4.17 %
630	1	4.17 %
650	1	4.17 %
670	1	4.17 %
720	1	4.17 %
750	2	8.33 %
780	2	8.33 %
805	1	4.17 %
840	1	4.17 %
910	1	4.17 %
915	1	4.17 %
930	1	4.17 %
990	2	8.33 %

You can see all data is printed with 3 columns: Score, Frequency and Frequency Rate. Program calculate data and print all, we can easily watch the data. However, when the score has many values, it is difficult to statistical. In this case, please go to test case 2.

Case 2: List all Score with range complex for Student acquired: Base Score-Frequency- Frequency rate:

```
frequencyClassified(toiecScores)
```

Score	Fre	Fre. Rate
[100 - 199]	1	4.17 %
[200 - 299]	3	12.5 %
[400 - 499]	1	4.17 %
[500 - 599]	3	12.5 %
[600 - 699]	4	16.67 %
[700 - 799]	5	20.83 %
[800 - 899]	2	8.33 %
[900 - 990]	5	20.83 %

The result is: Scores is grouped by the base score; program will show frequency and the rate foreach base score. That is easy to statistical.

Case 3: See all the scores with the same max frequency:

```
printAllMaxFrequency()
```

```
Max Frequency = 5
The range of the Scores base have the same max frequency:
Score   Fre   Fre. Rate
700     5     20.83 %
900     5     20.83 %
```

In case 2, you see base scores [700-799] and [900-990] have the same max frequency. So in case 3, the program will show these base scores.

Case 4: See all the scores with the same min frequency:

```
printAllMinFrequency ()
```

```
Min Frequency = 1
The range of the Scores base have the same min frequency:
Score   Fre   Fre. Rate
100    1     4.17 %
400    1     4.17 %
```

In case 2, you see base scores [100-199] and [400-499] have the same min frequency. So in case 4, the program will show these base scores.

Case 5: A function to find the average of the TOEIC students acquired.:.

```
print("Average Toeic Score that Student acquired=",averageOfToiecScore(toiecScores))
```

```
Average Toeic Score that Student acquired= 647
```

In case 5, the program will calculate the average score and round up to integer value and print out.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise15.py

20. Exercise 16 - Time difference calculator

20.1. Learning Objectives

20.1.1. Basic learning objectives

- understand the datetime module
- Now() function can be used to calculate negative time difference with Korea.
- You can understand the % operator and write a function that calculates leap years.

20.1.2. In-depth learning objectives

- You can calculate both the positive and negative time difference by storing the time difference between Korea and several countries in the list.

20.1.3. Problem situation

One month later, when the summer vacation began, I was invited to participate in the Jinhae-ha University Exchange Program at a university in California, USA. Before the trip, I was curious about the weather and time difference in the city to visit. I want to write a program that easily calculates the time difference for countries where the time is later than Korea, such as California. The time difference between Korea and California is 16 hours during summer time, which is faster than California time.

20.2. Problem analysis

20.2.1. The role of the computer

- Understanding the Python module that can save time
- Parallax to understand the process of calculating the date and time
- Understanding how to write Python functions

20.2.2. User

- Think about how to get the time difference input. In other words, think about how to express the time difference in countries that are later than Korea.

20.2.3. Input

- Parallax

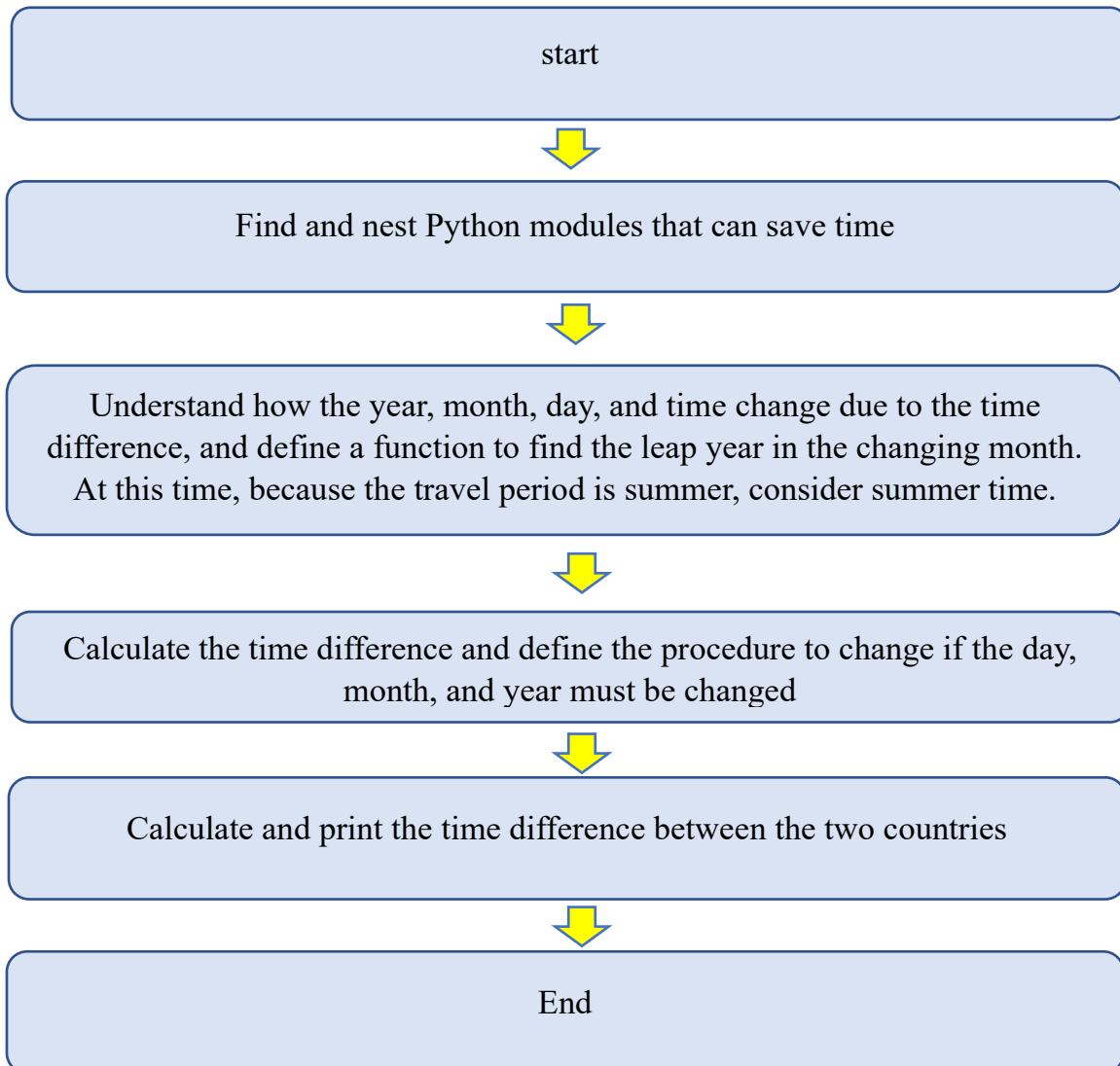
20.2.4. Output

- Korea time and current date and time in California

20.2.5. Data

- The Republic of Korea uses UTC+9, the time zone of which 9 hours are added to Coordinated Universal Time (UTC), and the time is called Pacific Standard Time (PST). Pacific Daylight-Saving Time (PDT) is used during summer time
- In the summer time period, there is a time difference of -16 hours, and in other periods, an additional hour is added.
- Since the program only considers countries with a later time than Korea, if the time becomes negative when the time is subtracted, it is considered that the correction should be made in order of day, month, and year

20.3. Full algorithm skeleton



20.4. Programming

20.4.1. Variables

- yy: The year in which the parallax was applied as a parameter of the function `is_leap_year()`
- yy, mm, dd, hh, min: The current year, month, hour, and minute of Korea are displayed as local variables of the function `jetlag()`.
- today: Current time in Korea
- date: A string containing the current date and time in California as a local variable of the function `jetlag()`
- gap: The time difference between the two countries, negative numbers are the time difference between the countries that are later in time than Korea.

20.4.2. Some explanations

module `datetime`: import `datetime`

`datetime.datetime.now()`: Python's `datetime` module provides a `datetime` class that stores both date and time information. The method `now()` of the `Datetime` class gives the current time. The `Datetime` class contains properties such as year (year), month (month), day (day), hour (hour), minute (minute), second (second), and microsecond (multiple seconds)

current time: Displays the current time as year/month/date/hour:minute.

- `today=datetime.datetime.now()`
- `today.strftime('%Y%m%d %H:%M')`

Function to find leap year `is_leap_year()`:

The function `is_leap_year()` that finds the number of years tells us that the value of the parameter `yy` is divided by 400 and is not divided by 400 or is not divided by 100, but is divided by 4, which means that it is a leap year.

```
def is_leap_year(yy):
    return yy % 400 == 0 or (yy%100) and yy%4==0
```

Function `jetlag()`

- The current country's year, month, day, hour, minute, and time difference are taken as parameters and the time of the region (California) is calculated.
- Time adjustment
- Adjustment of the day

```

def jetlag(yy, mm, dd, hh, mi, gap): #Function to calculate parallax
    hh +=gap                         #Korea and California time difference
    if hh < 0 :                      #If the time is negative, adjust the date
        dd = dd - 1
        hh += 24
    if dd ==0:                        #When the adjusted date reaches 0, the month is adjusted.
        mm = mm-1
    if mm ==0:                        #If the adjusted month is 0, adjust the year
        mm = 12
        yy -= 1
    if mm in [4, 6, 9, 11]:          #Calculate the last day of the adjusted month
        dd = 30                         #30 days last
    elif mm in [1,3,5,7,8,10,12]:#The last day is the 31st
        dd=31
    else:                            #The leap month needs to be calculated
        if is_leap_year(yy):          #Leap year
            dd = 29
        else:
            dd =28                  #If it's not a leap month
    date= "%d/%d/%d %d:%d" % (yy,mm,dd, hh,mi)
    return date

```

20.4.3.Programming-the whole program

The whole program is shown as below.

```

#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import datetime
def is_leap_year(yy):
    return yy % 400 == 0 or (yy%100) and yy%4==0

def jetlag(yy, mm, dd, hh, mi, gap): #Function to calculate parallax
    hh +=gap                         #Korea and California time difference
    if hh < 0 :                      #If the time is negative, adjust the date
        dd = dd - 1
        hh += 24
    if dd ==0:                        #When the adjusted date reaches 0, the month is adjusted.
        mm = mm-1
    if mm ==0:                        #If the adjusted month is 0, adjust the year
        mm = 12
        yy -= 1
    if mm in [4, 6, 9, 11]:          #Calculate the last day of the adjusted month
        dd = 30                         #30 days last
    elif mm in [1,3,5,7,8,10,12]:#The last day is the 31st
        dd=31
    else:                            #The leap month needs to be calculated
        if is_leap_year(yy):          #Leap year
            dd = 29
        else:
            dd =28                  #If it's not a leap month

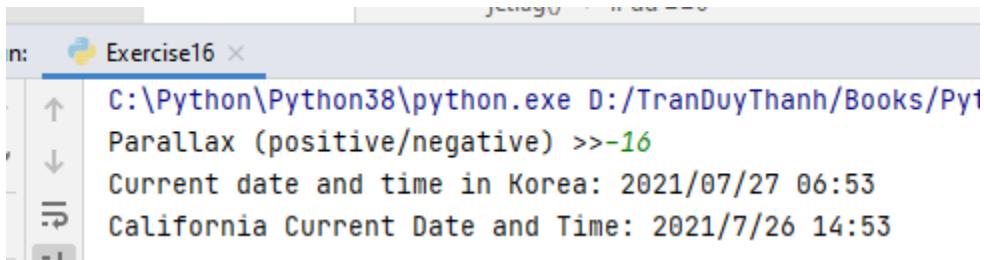
```

```
dd =28      #If it's not a leap month
date= "%d/%d/%d %d:%d" % (yy,mm,dd, hh,mi)
return date
today=datetime.datetime.now()
gap=int(input("Parallax (positive/negative) >>"))
print("Current date and time in Korea:",today.strftime("%Y/%m/%d %H:%M"))

date=jetlag(today.year,today.month,today.day,today.hour,today.minute,gap)
print("California Current Date and Time:",date)
```

20.5. Testing and debugging

Run program.



```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/Py1
Parallax (positive/negative) >>-16
Current date and time in Korea: 2021/07/27 06:53
California Current Date and Time: 2021/7/26 14:53
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise16.py

21. Exercise 17 - Time difference calculator (improve)

21.1. Problem situation

Let's calculate the time difference for each country by saving the time difference of different countries in a list and selecting a country as follows. Let's consider a country with a faster time than our country

Jetlag=[-16, -8, 2, -2]

countries = ["San Francisco", "London", "Sydney", "Jakarta"]

21.2. Programming-the whole program

These codes I improved from exercise 16:

- 1.calculate the time difference for each country by saving the time difference of different countries in a list and selecting a country
- 2.save all country over the world into Excel to calculate the time difference for each country
- 3.Make a menu for user choosing.

Excel file for all Countries GMT time:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise17/Timezone-countries.xlsx

A	B	C	D	E	F	G
1 Country- City	GMT TimeZone					
2 Afghanistan – Kabul	4					
3 Albania – Tirane	1					
4 Algeria – Algiers	1					
5 Andorra – Andorra La Vella	1					
6 Angola – Luanda	1					
7 Antigue and Barbuda – Saint John's	-4					
8 Argentina – Buenos Aires	-3					
9 Armenia – Yerevan	4					
10 Australia –Capital Territory – Canberra	10					
11 Austria – Vienna	1					
12 Azerbaijan – Baku	4					
13 Bahamas – Nassau	-5					
14 Bahrain – Al Manamah	3					
15 Bangladesh – Dhaka	6					
16 Barbados – Bridgetown	-4					
17 Belarus – Minsk	2					
18 Belgium – Brussels	1					
19 Belize – Belmopan	-6					
20 Benin – Porto Novo	1					
21 Bhutan – Thimphu	6					
22 Bolivia – La Paz	-4					
23 Bosnia – Herzegovina – Sarajevo	1					
24 Botswana – Gaborone	2					
25 Brazil – Distrito Federal – Brasilia	-3					
26 Brunei Darussalam – Bandar Seri Begawan	8					
27 Bulgaria – Sofia	2					
28 Burkina Faso – Ouagadougou	0					
29 Burundi – Bujumbura	2					
30 Cambodia – Phnom Penh	7					
31 Cameroon – Yaounde	1					
32 Canada – Ontario – Ottawa	-5					
33 Cape Verde – Praia	-1					
34 Central African Republic – Bangui	1					
35 Chad – N'djamena	1					
36 Chile – Santiago	-4					
37 China – Beijing	8					

Countries with GMT timezone is saved in excel, we use pandas library to read.

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
    #These codes I improved from Exercise 16.
    #1.calculate the time difference for each country by saving the time difference of
    #   different countries in a list and selecting a country
    #2.save all country over the world into Excel to calculate the time difference for each country
    #3.Make a menu for user choosing
import datetime
import pandas as pd
#some default country and jetlag
jetlagCountries = [-16, -8, 2, -2, -2, -1, -13, 1, -7]
countries = ["San Francisco", "London", "Sydney", "Jakarta", "Viet Nam", "China", "New York",
"Vladivostok", "Amsterdam"]
#for loading excel data (for all countries over the world)
#in this case I use international GMT time to calculate different time over the World with Korea
#all data will be read from EXCEL file with pandas library
jetlagMoreCountries = []
morecountries = []
korea_gmt=9

#this function use to check the year is leap year
def is_leap_year(yy):
    return yy % 400 == 0 or (yy%100) and yy%4==0
#find the date string for country diffirent time with Korea
def jetlag(yy, mm, dd, hh, mi, gap): #Function to calculate parallax
    hh +=gap           #Korea and California time difference
    if hh < 0 :       #If the time is negative, adjust the date
        dd = dd - 1
        hh += 24
    if dd ==0:         #When the adjusted date reaches 0, the month is adjusted.
        mm = mm-1
    if mm ==0:         #If the adjusted month is 0, adjust the year
        mm = 12
        yy -= 1
    if mm in [4, 6, 9, 11]: #Calculate the last day of the adjusted month
        dd = 30          #30 days last
    elif mm in [1,3,5,7,8,10,12]:#The last day is the 31st
        dd=31
    else:              #February: The leap month needs to be calculated
        if is_leap_year(yy): #Leap
            dd = 29
        else:
            dd =28          #If it's not a leap month
date= "%d/%d/%d %d:%d" % (yy,mm,dd, hh,mi)
return date
#this function use to print all countries
#user can enter the index or the name of country to see the Date
```

```

def printCountries(countries,isSmall=True):
    print("Countries:")
    for i in range(len(countries)):
        if isSmall:
            print(f'{i+1}.{countries[i]:<20}',end="")
        else:
            print(f'{i + 1}.{countries[i]:<50}', end="")
    if isSmall and (i+1) % 3 ==0:
        print()
    elif isSmall==False and ((i+1) % 4) ==0:
        print()
    print()
#this function use to show the current date for country by index
def showDateTimeForCountry(country,gap):
    today = datetime.datetime.now()
    print("You choose country:" + country)
    print("Current date and time in Korea:", today.strftime("%Y/%m/%d %H:%M"))
    date = jetlag(today.year, today.month, today.day, today.hour, today.minute, gap)
    print(country, "Current Date and Time:", date)
#this function will show list of country over the world or list of default,
#user can see and choose easily
#function will find the gap and country to calculate the different time
#isLittle=True, mean: countries,jetlagCountries is from default
#isLittle=False, mean: countries and GMT time is all over the World (Excel file)
def showDiffrentTimeCountries(countries,jetlagCountries,isLittle=True):
    while True:#loop to re-enter many countries to test
        printCountries(countries,isLittle) #print all countries, user can choose easily
        country=input("Please enter [index] or [name of country] or [exit] to break function:")
       isOkToShow=False
        index=-1
        if country.isdigit()==True: # if use find country by index
            index=int(country)
            if index>0 and index <= len(countries):
                country = countries[index - 1]
                if isLittle ==True:
                    gap = jetlagCountries[index-1]
                else:
                    gmtCountry = jetlagCountries[index - 1]
                    gap = korea_gmt - gmtCountry
                isOkToShow=True
        elif country.lower() == "exit":#if exit
            break
        else:#if user find country by name
            country = country.lower()
            for i in range(len(countries)):
                if (countries[i].lower() == country):
                    index = i
                    break
            if index>=0:
                country = countries[index]
                if isLittle ==True:
                    gap = jetlagCountries[index]
                else:
                    gmtCountry = jetlagCountries[index]
                    gap = korea_gmt - gmtCountry

```

```

isOkToShow=True
if isOkToShow==True:#is ok to show date for country
    if isLittle==False:
        gap = gap *-1;
    showDateTimeForCountry(country, gap)
else:
    print("The index or country name [%s] is not exit!"%country)
#function to read all countries from Excel
#in this case, gap must be recalculated before GMT is used
def showMoreCountries():
    if len(morecountries)==0:#the first time we load data from Excel
        data = pd.read_excel(r'data_exercise17/Timezone-countries.xlsx')#read data
        df = pd.DataFrame(data, columns=['Country- City','GMT TimeZone'])
        for i in range(len(df.index)-1):
            morecountries.append(df.iloc[i]["Country- City"])
            jetlagMoreCountries.append(float(df.iloc[i]["GMT TimeZone"])))
    #Reuse showDiffrentTimeCountries
    showDiffrentTimeCountries(morecountries,jetlagMoreCountries,False)
#this function is main function
#give 3 options for user
def doDiffrentTimeCountry():
    while True:
        print("1.Little Countries")#do with default countries in the list in memory
        print("2.More Countries")#do with all country over the world in Excel
        print("3.Exit application")#exit application
        choice=input("Please enter your choice:")
        if choice=="1":
            showDiffrentTimeCountries(countries,jetlagCountries,True)
        elif choice=="2":
            showMoreCountries()
        elif choice=="3":
            print("Thank you for using the app!")
            exit()
        else:
            print("Please choose correct number")
#call main function to use the application
doDiffrentTimeCountry()

```

21.3. Testing and debugging

Here are some test cases when I run the program, there are 3 menus:



- 1.Little Countries → test countries with the default list
- 2.More Countries → test countries over the world, it is read from Excel file
- 3.Exit application → exit the application

Test case 1: Now We choose [1] 1.Little Countries

```

1. Little Countries
2. More Countries
3. Exit application
Please enter your choice:1
Countries:
1. San Francisco      2. London          3. Sydney
4. Jakarta            5. Viet Nam         6. China
7. New York           8. Vladivostok    9. Amsterdam

Please enter [index] or [name of country] or [exit] to break function:

```

All countries in the default list will be showed, user can enter index (1→n) or and the [name of country] or exit to break the function.

Test case 2: Now we press 5 and enter:

```

Please enter [index] or [name of country] or [exit] to break function:5
You choose country:Viet Nam
Current date and time in Korea: 2021/04/18 15:37
Viet Nam Current Date and Time: 2021/4/18 13:37
Countries:
1. San Francisco      2. London          3. Sydney
4. Jakarta            5. Viet Nam         6. China
7. New York           8. Vladivostok    9. Amsterdam

Please enter [index] or [name of country] or [exit] to break function:

```

Datetime of Vietnam will be showed and comparing with the Korea time

Test case 3: We can find by country name (press VietName instead 5):

```

Please enter [index] or [name of country] or [exit] to break function:VIET NAM
You choose country:Viet Nam
Current date and time in Korea: 2021/04/18 15:40
Viet Nam Current Date and Time: 2021/4/18 13:40
Countries:
1. San Francisco      2. London          3. Sydney
4. Jakarta            5. Viet Nam         6. China
7. New York           8. Vladivostok    9. Amsterdam

Please enter [index] or [name of country] or [exit] to break function:

```

Datetime of Vietnam will be showed and comparing with the Korea time, about 2 hours

Test case 4:

The time in Vietnam is after Korea, now We just test Sydney:

```
Please enter [index] or [name of country] or [exit] to break function:sydney
You choose country:Sydney
Current date and time in Korea: 2021/04/18 15:45
Sydney Current Date and Time: 2021/4/18 17:45
```

Countries:

1.San Francisco	2.London	3.Sydney
4.Jakarta	5.Viet Nam	6.China
7.New York	8.Vladivostok	9.Amsterdam

```
Please enter [index] or [name of country] or [exit] to break function:|
```

The time in Sydney is before Korea. About 1 hour

Test case 5: We enter a country not in the default list, example “Cambodia”

```
Please enter [index] or [name of country] or [exit] to break function:cambodia
The index or country name [cambodia] is not exit!
```

Countries:

1.San Francisco	2.London	3.Sydney
4.Jakarta	5.Viet Nam	6.China
7.New York	8.Vladivostok	9.Amsterdam

```
Please enter [index] or [name of country] or [exit] to break function:|
```

Program can not find the country and show “The index or country name [cambodia] is not exit!”

Test case 6: User can enter “exit” to break the function little countries, I will be backed to the main menu list:

```
Please enter [index] or [name of country] or [exit] to break function:cambodia
The index or country name [cambodia] is not exit!
```

Countries:

1.San Francisco	2.London	3.Sydney
4.Jakarta	5.Viet Nam	6.China
7.New York	8.Vladivostok	9.Amsterdam

```
Please enter [index] or [name of country] or [exit] to break function:exit
```

- 1.Little Countries
- 2.More Countries
- 3.Exit application

```
Please enter your choice:|
```

Test case 7: User choose 2 to use “More countries”, this function will read data of all countries over the World from Excel file by pandas library:

1.Little Countries
2.More Countries
3.Exit application
Please enter your choice:2

The results:

1.Little Countries	2.More Countries	3.Exit application
Please enter your choice:2		
Countries:		
1.Afghanistan - Kabul	2.Albania - Tirane	3.Algeria - Algiers
5.Angola - Luanda	6.Antigued and Barbuda - Saint John's	7.Argentina - Buenos Aires
9.Australia - Capital Territory - Canberra	10.Austria - Vienna	11.Azerbaijan - Baku
13.Bahrain - Al Manamah	14.Bangladesh - Dhaka	15.Barbados - Bridgetown
17.Belgium - Brussels	18.Belize - Belmopan	19.Benin - Porto Novo
21.Bolivia - La Paz	22.Bosnia - Herzegovina - Sarajevo	23.Botswana - Gaborone
25.Brunei Darussalam - Bandar Seri Begawan	26.Bulgaria - Sofia	27.Burkina Faso - Ouagadougou
29.Cambodia - Phnom Penh	30.Cameroun - Yaounde	31.Canada - Ontario - Ottawa
33.Central African Republic - Bangui	34.Chad - Ndjamena	35.Chile - Santiago
37.Colombia - Bogota	38.Comoros - Moroni	39.Congo - Brazzaville
41.Cook Islands - Rarotonga	42.Costa Rica - San Jose	43.Cote D'Ivoire - Yamoussoukro
45.Cuba - Havana	46.Cyprus - Nicosia	47.Czech Republic - Prague
49.Djibouti - Djibouti	50.Dominica - Roseau	51.Dominican Republic - Santo Domingo
53.Ecuador - Quito	54.Egypt - Cairo	55.El Salvador - San Salvador
57.Eritrea - Asmara	58.Estonia - Tallinn	59.Ethiopia - Addis Ababa
61.Finland - Helsinki	62.France - Paris	63.Gabon - Libreville
65.Georgia - Tbilisi	66.Germany - Berlin - Berlin	67.Ghana - Accra
69.Grenada - Saint George's	70.Guatemala - Guatemala	71.Guinea - Conakry
73.Guyana - Georgetown	74.Haiti - Port-au-Prince	75.Honduras - Tegucigalpa
77.Iceland - Reykjavik	78.India - New Delhi	79.Indonesia - Java - Jakarta
81.Iraq - Baghdad	82.Ireland - Dublin	83.Isole di Man - Douglas
85.Italy - Rome	86.Jamaica - Kingston	87.Japan - Tokyo
89.Kazakhstan - Astana	90.Kenya - Nairobi	91.Kiribati - Tarawa
93.Kuwait - Kuwait City	94.Kyrgyzstan - Bishkek	95.Laos - Vientiane
97.Lebanon - Beirut	98.Lesotho - Maseru	99.Liberia - Monrovia
101.Liechtenstein - Vaduz	102.Lithuania - Vilnius	103.Luxembourg - Luxembourg
105.Madagascar - Antananarivo	106.Malawi - Lilongwe	107.Malaysia - Kuala Lumpur
109.Malta - Valletta	110.Mauritania - Nouakchott	111.Mauritius - Port Louis
113.Moldova - Kishinev	114.Monaco - Monaco	115.Mongolia - Ulaanbaatar
117.Mozambique - Maputo	118.Nyammar - Rangoon	119.Namibia - Windhoek
121.Nepal - Kathmandu	122.Netherlands - Amsterdam	123.New Zealand - Wellington
125.Niger - Niamey	126.Nigeria - Abuja	127.Niue - Alofi
129.Norway - Oslo	130.Oman - Muscat	131.Pakistan - Islamabad
133.Panama - Panama	134.Papua New Guinea - Port Moresby	135.Paraguay - Asuncion
137.Philippines - Manila	138.Poland - Warsaw	139.Portugal - Lisbon
141.Qatar - Ad Dawhah	142.Reunion (French) - Saint - Denis	143.Romania - Bucharest
145.Rwanda - Kigali	146.Saint Kitts & Nevis - Basseterre	147.Saint Lucia - Castries
149.Samoa - Apia	150.San Marino - San Marino	151.Sao Tome and Principe - Sao Tome
153.Senegal - Dakar	154.Seychelles - Victoria	155.Sierra Leone - Freetown
157.Slovak Republic - Bratislava	158.Slovenia - Ljubljana	159.Solomon Islands - Honiara
161.South Africa - Pretoria	162.Spain - Madrid	163.Sri Lanka - Colombo
165.Suriname - Paramaribo	166.Swaziland - Mbabane	167.Sweden - Stockholm
169.Syria - Damascus	170.Taiwan - Taipei	171.Tajikistan - Dushanbe
173.Thailand - Bangkok	174.Togo - Lome	175.Tonga - Nukualofa
177.Tunisia - Tunis	178.Turkey - Ankara	179.Turkmenistan - Ashgabat
181.U.K. - England - London	182.U.S.A. - Washington D.C	183.Uganda - Kampala
185.United Arab Emirates - Abu Dhabi	186.Uruguay - Montevideo	187.Uzbekistan - Tashkent
189.Vatican City State - Vatican City	190.Venezuela - Caracas	191.Vietnam - Hanoi
193.Yugoslavia - Serbia - Belgrade	194.Zambia - Lusaka	192.Yemen - Sana

Please enter [index] or [name of country] or [exit] to break function:[]

All the test case in mode 2 “More Countries” are the same mode 1 “Little Countries”.
Now let's take a look some test case:

Test case 8: I enter 191 (it means Vietnam)

157.Slovak Republic - Bratislava	158.Slovenia - Ljubljana	159.Solomon Islands - Honiara
161.South Africa - Pretoria	162.Spain - Madrid	163.Sri Lanka - Colombo
165.Suriname - Paramaribo	166.Swaziland - Mbabane	167.Sweden - Stockholm
169.Syria - Damascus	170.Taiwan - Taipei	171.Tajikistan - Dushanbe
173.Thailand - Bangkok	174.Togo - Lome	175.Tonga - Nukualofa
177.Tunisia - Tunis	178.Turkey - Ankara	179.Turkmenistan - Ashgabat
181.U.K. - England - London	182.U.S.A. - Washington D.C	183.Uganda - Kampala
185.United Arab Emirates - Abu Dhabi	186.Uruguay - Montevideo	187.Uzbekistan - Tashkent
189.Vatican City State - Vatican City	190.Venezuela - Caracas	191.Vietnam - Hanoi
193.Yugoslavia - Serbia - Belgrade	194.Zambia - Lusaka	

Please enter [index] or [name of country] or [exit] to break function:191

You choose country:Vietnam - Hanoi

Current date and time in Korea: 2021/04/18 15:59

Vietnam - Hanoi Current Date and Time: 2021/4/18 13:59

The datetime is showed about. Time of Vietnam is after Korean time about 2 hours

Test case 9: Press exit to break the function

```
177.Tunisia - Tunis          178.Turkey - Ankara          179.Turkmenistan - Ashgabat
181.U.K. - England - London  182.U.S.A. - Washington D.C  183.Uganda - Kampala
185.United Arab Emirates - Abu Dhabi 186.Uruguay - Montevideo 187.Uzbekistan - Tashkent
189.Vatican City State - Vatican City 190.Venezuela - Caracas 191.Vietnam - Hanoi
193.Yugoslavia - Serbia - Belgrade 194.Zambia - Lusaka

Please enter [index] or [name of country] or [exit] to break function:exit
1.Little Countries
2.More Countries
3.Exit application
Please enter your choice:|
```

Test case 10: press 3 to exit application:

```
Please enter [index] or [name of country] or [exit] to break function:exit
1.Little Countries
2.More Countries
3.Exit application
Please enter your choice:3
Thank you for using the app!
```

Maybe some data I collected in the excel file is not 100% exactly, but the data format is GMT, user can easily to choose and see the time for each country. Sorry about that.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise17.py

22. Exercise 18 - Achievement processor

22.1. Learning Objectives

22.1.1. Basic learning objectives

- You can create a student class
- You can create student objects
- You can set the initial value when creating an object using the `__init__` constructor.

22.1.2. In-depth learning objectives

- Appropriate access and change of object property values
- You can add a new method for assigning grades to student classes

22.1.3. Problem situation

I want to process student grades. Students' grades are determined by midterm, final, and assignments. I would like to inform each student by entering the student's name, midterm, final, and assignment grades to obtain the sum of the scores and evaluation of the student's scores.

22.2. Problem analysis

22.2.1. Input

- Enter the student's name as a string
- Enter midterm, final, and assignment grades in integer type

22.2.2. Output

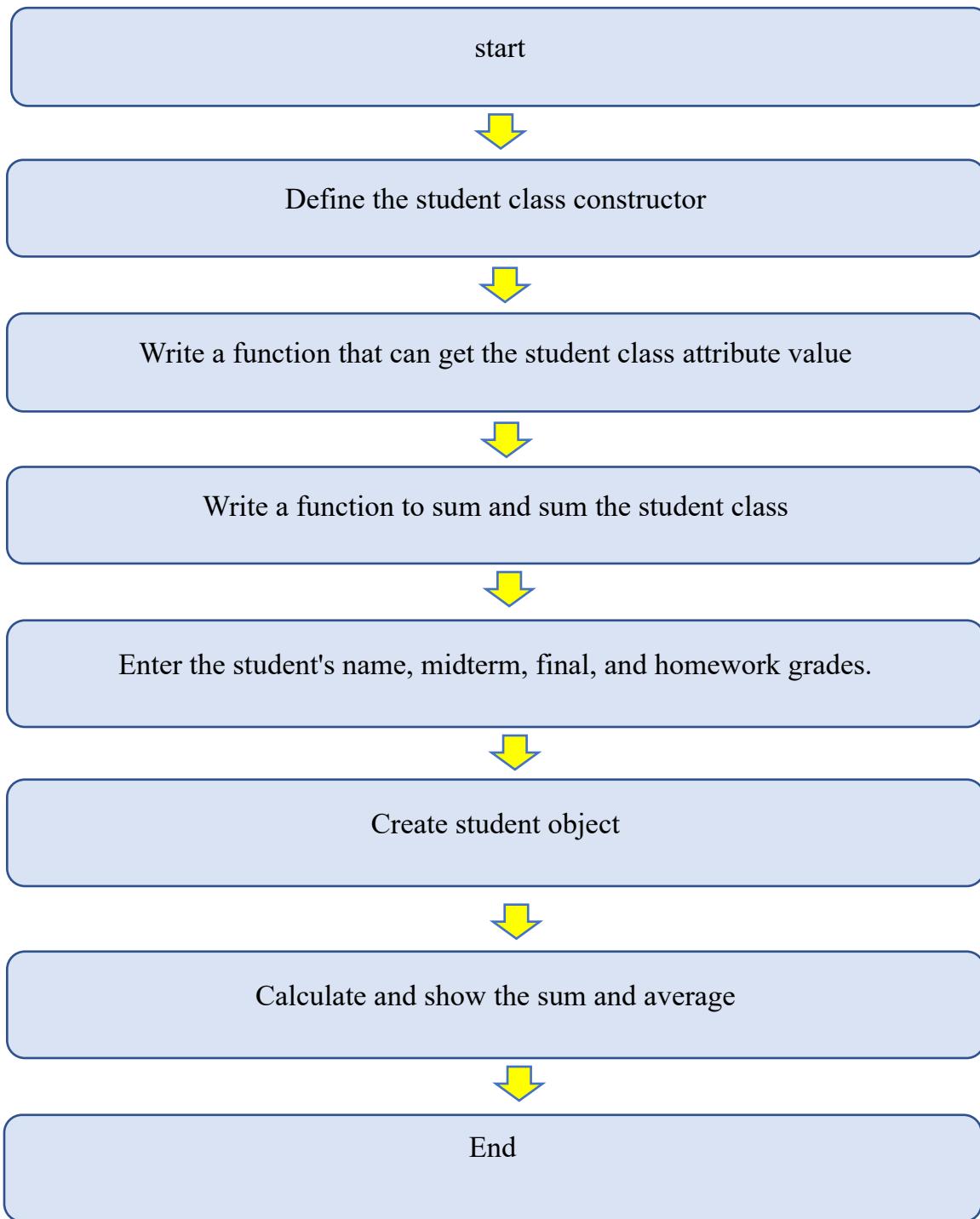
- Student name
- Calculated sum
- Calculated average

22.2.3. Problem decomposition

- Define a student class.
- Enter grades and create student objects
- Find and show sums and averages

22.2.4. Data

- Any length string representing the student's name
- An integer representing the score from 0 to 100
- An integer representing the sum from 0 to 300
- Real number representing the mean from 0.0 to 100.0

22.3. Full algorithm skeleton

22.4. Programming

22.4.1. Variables

- name: Student name
- midScore, finalScore, projectScore: Student score
- student1: Student object.

22.4.2. Some explanations

Student class definition

Initialization using `__init__()`

Write method (function) to inform attribute value

Write sum and average method (function)

```
#Class definition
class Student:
    def __init__(self, name, midScore, finalScore, projectScore):
        self.name = name
        self.midScore = midScore
        self.finalScore = finalScore
        self.projectScore = projectScore

    def get_name(self):
        return self.name

    def get_sum(self):
        return self.sum

    def get_avg(self):
        return self.avg

    def calculate(self):
        self.sum = self.midScore + self.finalScore + self.projectScore
        self.avg = self.sum / 3
```

Students' name, midterm, and final exams are entered.

Create object student1

```
name = input("Enter your name:")
midScore = int(input("Enter midterm grades:"))
finalScore = int(input("Enter final exam grades:"))
projectScore = int(input("Enter assignment grade"))

#Object creation
student1 = Student(name, midScore, finalScore, projectScore)
```

Call `calculate()` using the created object student1

Define `get()` method and `set()` to access or change property data

```
student1.calculate()  
  
print("Student name=",student1.get_name())  
print("Sum=",student1.get_sum())  
print("Average=",student1.get_avg())
```

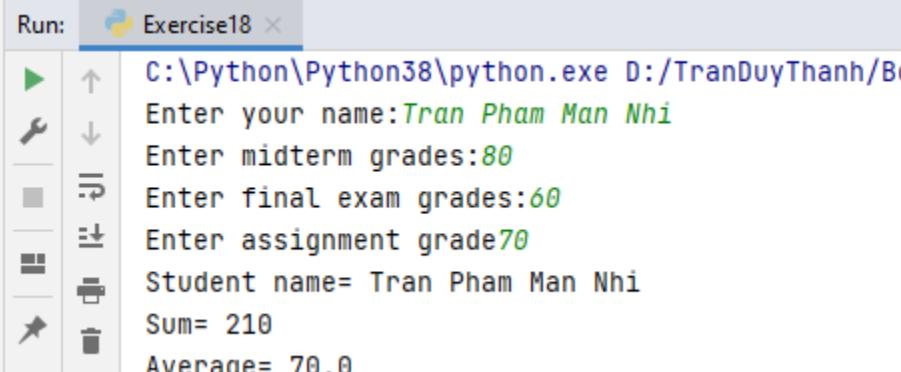
22.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh  
#Email: thanhtd@uel.edu.vn  
#Phone: 0987773061  
#Blog for self-study: https://duythanhcse.wordpress.com  
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni  
  
#Class definition  
class Student:  
    def __init__(self,name, midScore, finalScore, projectScore):  
        self.name=name  
        self.midScore=midScore  
        self.finalScore=finalScore  
        self.projectScore=projectScore  
  
    def get_name(self):  
        return self.name  
  
    def get_sum(self):  
        return self.sum  
  
    def get_avg(self):  
        return self.avg  
  
    def calculate(self):  
        self.sum=self.midScore+ self.finalScore + self.projectScore  
        self.avg=self.sum/3  
  
name=input("Enter your name:")  
midScore=int(input("Enter midterm grades:"))  
finalScore=int(input("Enter final exam grades:"))  
projectScore=int(input("Enter assignment grade"))  
  
#Object creation  
student1=Student(name,midScore,finalScore,projectScore)  
#Sum and average calculation method call  
student1.calculate()  
  
print("Student name=",student1.get_name())  
print("Sum=",student1.get_sum())  
print("Average=",student1.get_avg())
```

22.5. Testing and debugging

Run program.



The screenshot shows a Python terminal window titled "Exercise18". The terminal displays the following output:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/B...
Enter your name:Tran Pham Man Nhi
Enter midterm grades:80
Enter final exam grades:60
Enter assignment grade70
Student name= Tran Pham Man Nhi
Sum= 210
Average= 70.0
```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise18.py

23. Exercise 19 - Achievement processor (improve)

23.1. Problem situation

Let's modify the grades to be graded as A, B, C, D, and F. However, A grade is 90 points or more, B grade is 80 or more and less than 90, C grade is 70 or more and less than 80, D grade is 60 or more and less than 70, F grade is less than 60

23.2. Programming-the whole program

These codes I improved from exercise 18:

- 1.Create an enum to calculate Ranking for Student
- 2.Create getRanking method to calculate ranking for Student
- 3.Create printInfor and printInforWithTitle to print information for Student
- 4.make a loop with menu to test some test case for Student
 - 4.1. Make a new Student
 - 4.2. Make a list to store and print all Student that entered
 - 4.3. Find a student
 - 4.4. Exit the application.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
```

```
#Description:
#These codes I improved from Exercise 18.
# 1.Create an enum to calculate Ranking for Student
# 2.Create getRanking method to calculate ranking for Student
# 3.Create printInfor and printInforWithTitle to print information for Student
# 4.make a loop with menu to test some test case for Student
# 4.1.make a new Student
# 4.2.make a list to store and print all Student that entered
# 4.3.Find a student
# 4.4.Exit the application
from enum import Enum
#enum for ranking
#name is A,B,C,D, F level
#value is description for the level
class Ranking(Enum):
    A = "90->100"
    B = "80->89"
    C = "70->79"
    D = "60->69"
    F = "<60"
#Class definition
class Student:
    def __init__(self, name, midScore, finalScore, projectScore):
        self.name = name
```

```
self.midScore=midScore
self.finalScore=finalScore
self.projectScore=projectScore
def get_name(self):
    return self.name
def get_sum(self):
    return self.sum
def get_avg(self):
    return self.avg
def calculate(self):
    self.sum=self.midScore+ self.finalScore + self.projectScore
    self.avg=round(self.sum/3)
def getRanking(self):
    if self.avg >=90:
        self.ranking=Ranking.A
    elif self.avg>=80:
        self.ranking = Ranking.B
    elif self.avg>=70:
        self.ranking=Ranking.C
    elif self.avg>=60:
        self.ranking = Ranking.D
    else:
        self.ranking = Ranking.F
    return self.ranking
# print information only
def printInfor(self):
    print(f'{self.get_name()[:15]}{self.get_sum()[:10]}{self.get_avg()[:10]}'
          f'{self.getRanking().name[:10]}{self.getRanking().value[:10]}' )
# print information with title
def printInforWithTitle(self):
    print(f'{"Name":<15}{"Sum":<10}{"Avg":<10}{"Rank":<10}{"(Note)":<10}')
    self.printInfor()
listOfStudent=[] #list to save student
while True:#loop to test student
    print("1.Enter a new grade Student")
    print("2.Print All grade Student")
    print("3.Find a Student")
    print("4.Exit")
    choose=input("Please choose[1..4]>>")
    if choose=="1":
        name=input("Student Name:")
        #Enter midterm grades
        midScore=int(input("Enter midterm grades:"))
        #Enter final exam grades
        finalScore=int(input("Enter final exam grades:"))
        #Enter assignment grade
        projectScore=int(input("Enter assignment grade:"))
        # Object creation
        stObj=Student(name,midScore,finalScore,projectScore)
        #Sum and average calculation method call
        stObj.calculate()
        print("Student infor that you entered:")
        stObj.printInforWithTitle()
        listOfStudent.append(stObj)
    elif choose == "2":
```

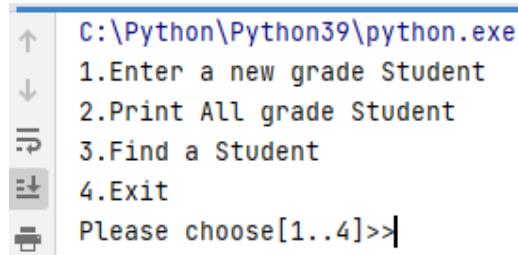
```

print("List Of Students:")
print(f"{'Name':<15}{'Sum':<10}{'Avg':<10}{'Rank':<10}{'(Note)':<10}")
for stObj in listOfStudent:
    stObj.printInfor()
elif choose == "3":
    name = input("Enter Student Name you want to find:")
    foundObj=None
    for stObj in listOfStudent:
        if stObj.get_name().lower()==name.lower():
            foundObj=stObj
            break
    if foundObj==None:
        print("Can not find student name ",name)
    else:
        foundObj.printInforWithTitle()
elif choose == "4":
    print("Thank you for your using the app")
    break

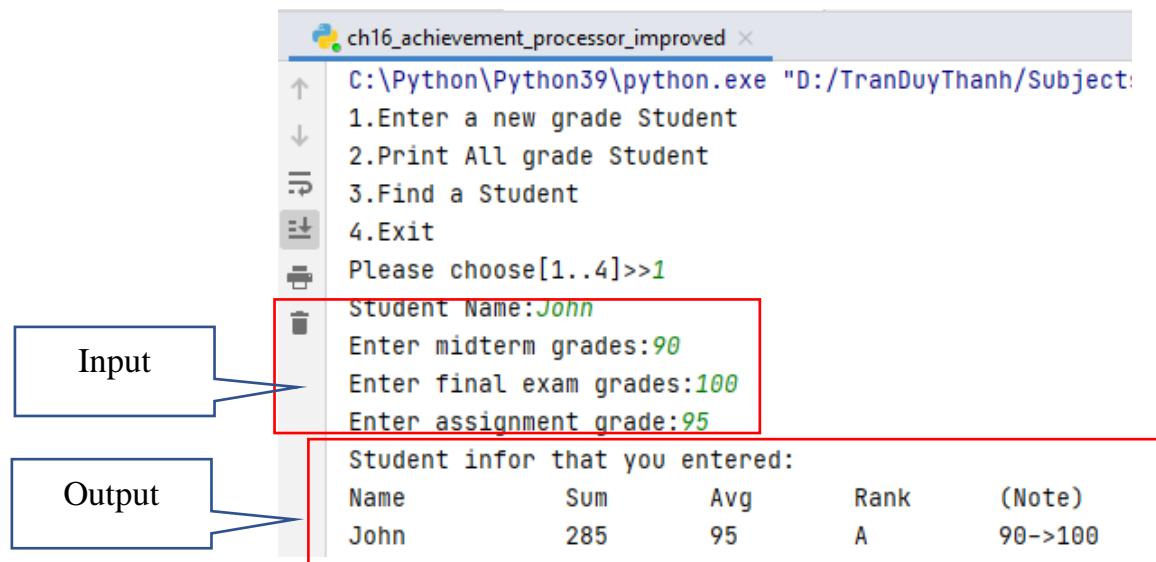
```

23.3. Testing and debugging

Here are some testing cases for the programming, there are 4 menus when application start:

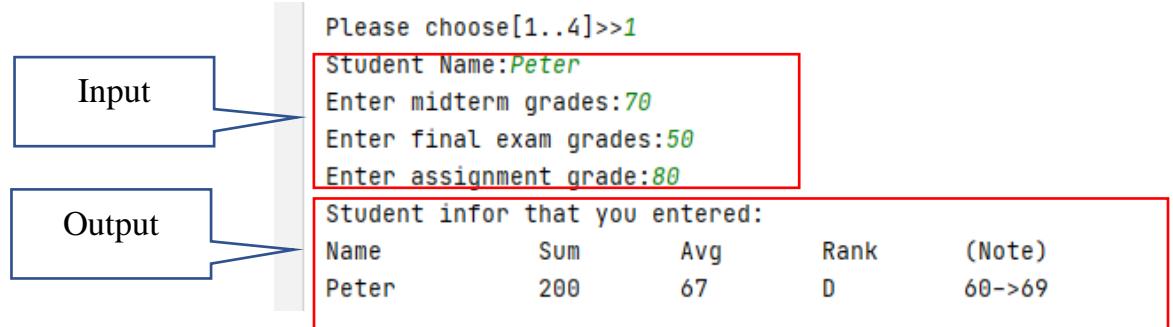


Test case 1: Enter a new Grade Student with A level

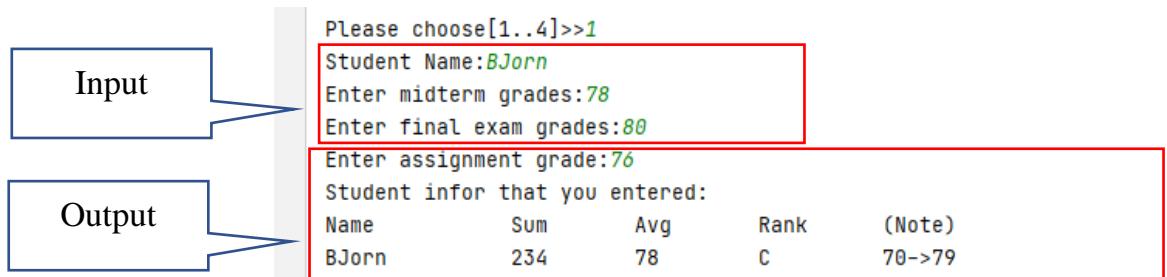


In test case 1, we enter a new Student as above. When finishing input the data, information for this Student will be showed.

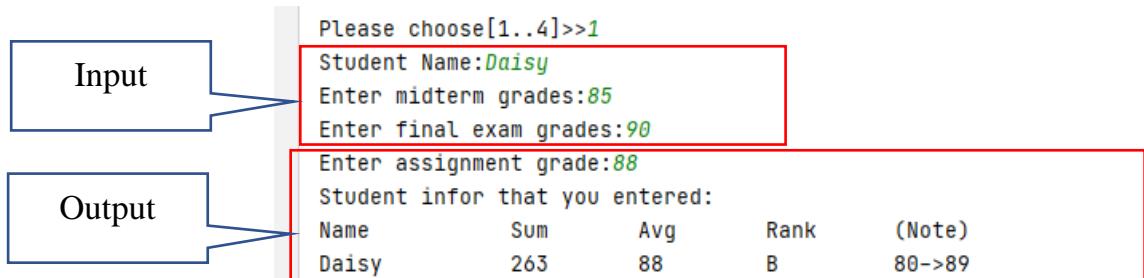
Test case 2: Enter a new more Grade Student with D level, press 1



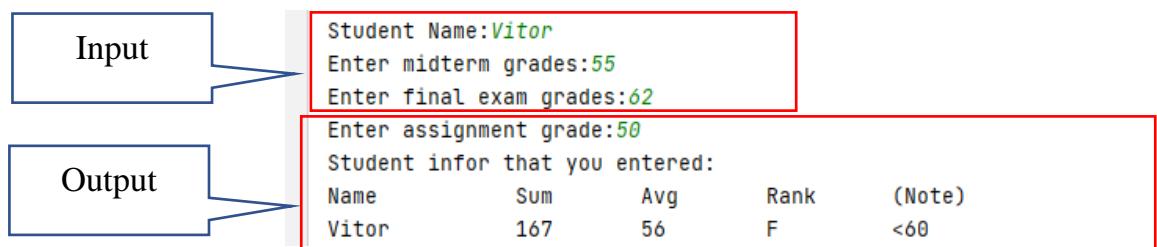
Test case 3: Enter a new more Grade Student with C level, press 1



Test case 4: Enter a new more Grade Student with B level, press 1



Test case 5: Enter a new more Grade Student with F level, press 1



Test case 6: Show the list of Student, press 2:

```

1.Enter a new grade Student
2.Print All grade Student
3.Find a Student
4.Exit
Please choose[1..4]>>2
List Of Students:
Name      Sum   Avg   Rank   (Note)
John      285   95    A      90->100
Peter     200   67    D      60->69
BJorn     234   78    C      70->79
Daisy     263   88    B      80->89
Vitor     167   56    F      <60

```

Output

Test case 7: Find a Student (FOUND), press 3:

```

1.Enter a new grade Student
2.Print All grade Student
3.Find a Student
4.Exit
Please choose[1..4]>>3
Enter Student Name you want to find:BJorn

```

Name	Sum	Avg	Rank	(Note)
BJorn	234	78	C	70->79

Input

Result

Test case 8: Find a Student (NOT FOUND), press 3:

```

1.Enter a new grade Student
2.Print All grade Student
3.Find a Student
4.Exit
Please choose[1..4]>>3
Enter Student Name you want to find:doremon
Can not find student name doremon

```

Application cannot find the student's name [doremon]

Test case 9: Exit application, press 4

```
1.Enter a new grade Student  
2.Print All grade Student  
3.Find a Student  
4.Exit  
Please choose[1..4]>>4  
Thank you for your using the app
```

Application will be stopped and say thank you.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise19.py

24. Exercise 20 - Online shopping cart

24.1. Learning Objectives

24.1.1. Basic learning objectives

- You can define a list as a class attribute
- You can create multiple shopping cart objects.
- You can change the attribute value of the shopping cart object using an operation

24.1.2. In-depth learning objectives

- You can define a method to manage shopping cart information more efficiently

24.1.3. Problem situation

Several people shop in an online shopping mall, but I am curious about how each customer puts the ordered item in the shopping cart and calculates it. I want to program a G program that calculates the final amount after memorizing the items that several people add to or delete from the shopping cart.

24.2. Problem analysis

24.2.1. The role of the computer

- Shopping cart class
- Defining methods and properties in a class
- Creating class objects and using methods

24.2.2. User

- Think about the attributes of the shopping cart by thinking about what information to give when users put things in the shopping cart.
- Thinking about what users can do with things to buy

24.2.3. Input

- None

24.2.4. Output

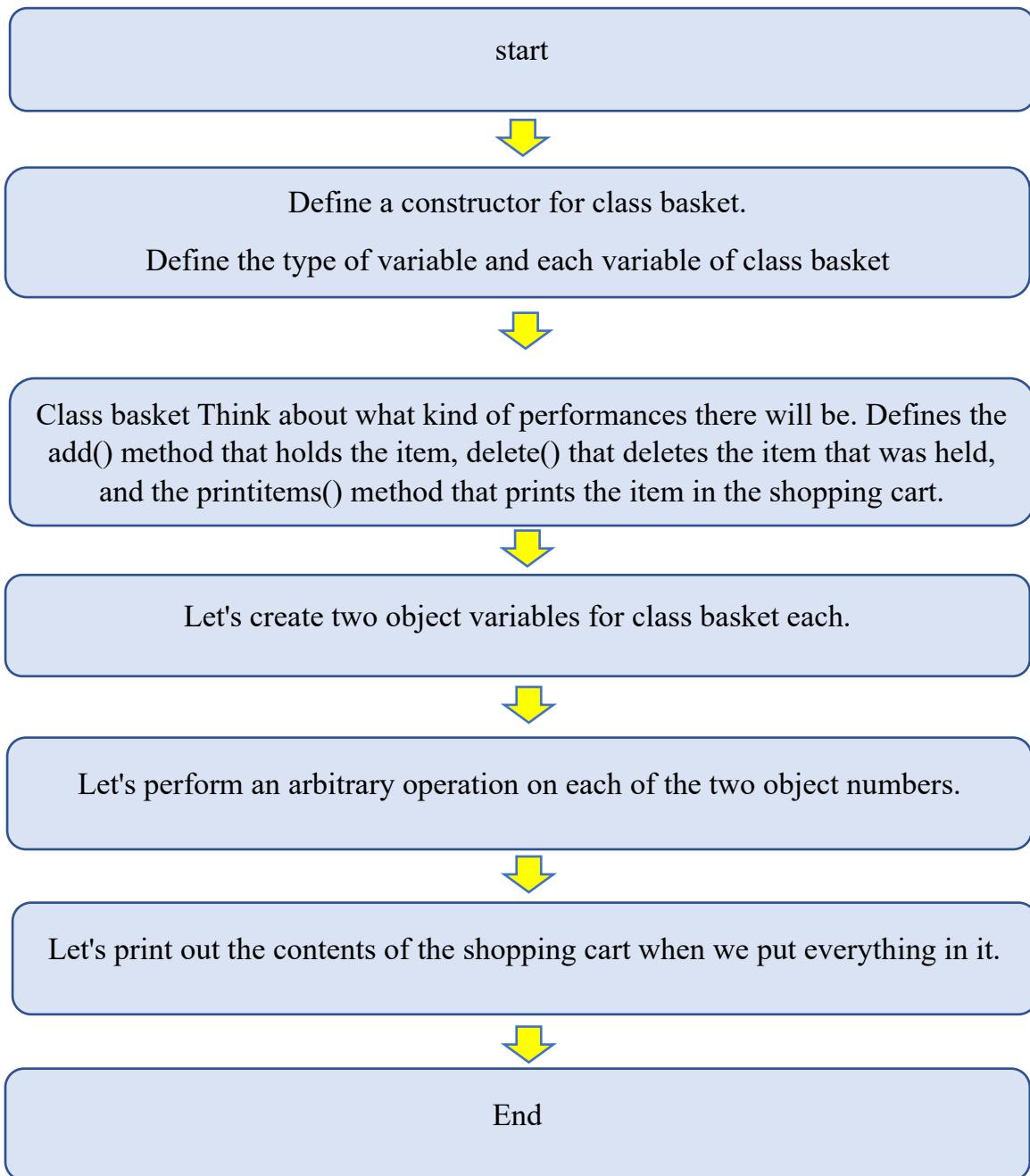
- Amount in the shopping cart and total payment

24.2.5. Data

- The price of the thing. For example, 1 box of ramen (10): 5900 won, 1 box of coffee mix (100): 10000 won, 1 bottle of milk: 2,500 won, 5 kg of apples: 23,000 won, 1 banana: 5,000 won

- List to process the data in the shopping cart

24.3. Full algorithm skeleton



24.4. Programming

24.4.1. Variables

- Basket: class name
- Id, total, noitems: properties of class basket
- This indicates whose shopping cart (id) is, how much is the total amount (total), and how many items (noitems) are contained in it.
- Items, prices, quantity: properties of class Basket
- Each item's name (items), unit price (prices), and quantity (quantity) are displayed. It is defined as a list because it can contain multiple items for the same item.
- cjBasket, jsBasket: object variables of class basket

24.4.2. Some explanations

Constructor definition of class Basket:

```
class Basket:
    def __init__(self,id):
        self.id=id
        self.items=[]
        self.prices=[]
        self.quantity=[]
        self.total=0
        self.noitems=0
```

Define add method of class Basket

The method add takes the name of the item, the price, how many items to buy, item, price, and qty as parameters and adds each as an attribute variable value of class Basket.

```
def add(self,item,price,qty):
    self.items.append(item)
    self.prices.append(price)
    self.quantity.append(qty)
    self.total += price * qty
    self.noitems += 1
```

Define delete method of class Basket

```
def delete(self,item,qty):
    for i in range(self.noitems):
        if item == self.items[i]:
            self.quantity[i]-=qty
            self.total -= self.prices[i]*qty
            if self.quantity[i]==0:
                self.noitems -=1
                del self.items[i]
                del self.quantity[i]
                del self.prices[i]
            break
```

When the user designates an item to be deleted, it is deducted from the total amount using the quantity and unit price, and after reducing the number of items by one, when the number of items reaches 0, they are deleted from the shopping cart.

Create 2 Basket class instances

```
cjBasket =Basket("Thanh Tran")
jsBasket =Basket("Pham Dieu")
cjBasket.add("banana",5000,2)
cjBasket.add("Milk",3000,1)
jsBasket.add("Ramen",5900,1)
jsBasket.add("Coffee",10000,2)
cjBasket.printitems()
jsBasket.printitems()
```

Delete 1 bottle of milk from cjBasket and 1 box of coffee mix from jsBasket.

```
cjBasket.delete("Milk",1)
cjBasket.printitems()
jsBasket.delete("Coffee Mix",1)
jsBasket.printitems()
```

Printitems() method that prints the contents of the shopping cart

```
def printitems(self):
    print(self.id,"shopping cart")
    for i in range(self.noitems):
        print(self.items[i],self.prices[i],self.quantity[i])
    print("** total =",self.total,", noitems=",self.noitems)
```

24.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
class Basket:
    def __init__(self,id):
        self.id=id
        self.items=[]
        self.prices=[]
        self.quantity=[]
        self.total=0
        self.noitems=0
    def add(self,item,price,qty):
        self.items.append(item)
        self.prices.append(price)
        self.quantity.append(qty)
        self.total += price * qty
        self.noitems += 1
    def delete(self,item,qty):
```

```

for i in range(self.noitems):
    if item == self.items[i]:
        self.quantity[i]-=qty
        self.total -=self.prices[i]*qty
        if self.quantity[i]==0:
            self.noitems -=1
            del self.items[i]
            del self.quantity[i]
            del self.prices[i]
        break
def printitems(self):
    print(self.id,"shopping cart")
    for i in range(self.noitems):
        print(self.items[i],self.prices[i],self.quantity[i])
    print("** total = ",self.total, ", noitems= ",self.noitems)

cjBasket =Basket("Thanh Tran")
jsBasket =Basket("Pham Dieu")
cjBasket.add("banana",5000,2)
cjBasket.add("Milk",3000,1)
jsBasket.add("Ramen",5900,1)
jsBasket.add("Coffee",10000,2)
cjBasket.printitems()
jsBasket.printitems()
cjBasket.delete("Milk",1)
cjBasket.printitems()
jsBasket.delete("Coffee Mix",1)
jsBasket.printitems()

```

24.5. Testing and debugging

Run program and see the results:

```

Exercise20 x
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/F
Thanh Tran shopping cart
banana 5000 2
Milk 3000 1
** total = 13000 , noitems= 2
Pham Dieu shopping cart
Ramen 5900 1
Coffee 10000 2
** total = 25900 , noitems= 2
Thanh Tran shopping cart
banana 5000 2
** total = 10000 , noitems= 1
Pham Dieu shopping cart
Ramen 5900 1
Coffee 10000 2
** total = 25900 , noitems= 2

```

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise20.py

25. Exercise 21 - Online shopping cart (improve)

25.1. Problem situation

Improve Exercise 20. If you add an item of the same item to the shopping cart, let's modify the method add() of the class Basket that retrieves the contents of the shopping cart and updates the number of items in the shopping cart.

Also, what will happen if you execute jsBasket.printitems() in Program 43 after adding the sentence jsBasket.delete("Coffee Mix",2) in program line 42? If there is a problem, try modifying the program.

25.2. Programming-the whole program

These codes I improved from exercise 20:

- 1. improve add method when item is the same name->update quantity and total
- 2. improve delete method->check quantity is valid or not, return True/False
- 3. improve printitems method -> text align to print
- 4. call Baseket object and test all method of Baset 1.Create an enum to calculate

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
```

```
#Description:
#These codes I improved from Exercise 20.
#1. improve add method when item is the same name->update quantity and total
#2. improve delete method->check quantity is valid or not, return True/False
#3. improve printitems method -> text align to print
#4. call Baseket object and test all method of Baset
class Basket:
    def __init__(self,id):
        self.id=id
        self.items=[]
        self.prices=[]
        self.quantity=[]
        self.total=0
        self.noitems=0
    def add(self,item,price,qty):
        #if item is the same name
        if self.items.__contains__(item) ==True:
            position=self.items.index(item)
            totalPrevious=self.prices[position]*self.quantity[position]
            self.prices[position]=price
            #update quantity for item
            self.quantity[position]=qty+self.quantity[position]
            #remove total of previous
            self.total=self.total-totalPrevious
```

```

#update new total
self.total+=price *self.quantity[position]
else:
    self.items.append(item)
    self.prices.append(price)
    self.quantity.append(qty)
    self.total +=price *qty
    self.noitems += 1
#this function use to delete item from basket
#return True when deleting successful, False is not successful
def delete(self,item,qty):
    for i in range(self.noitems):
        if item == self.items[i]:
            #make sure the quantity is valid
            if qty>0 and qty<=self.quantity[i]:
                self.quantity[i]-=qty
                self.total -=self.prices[i]*qty
                if self.quantity[i]==0:
                    self.noitems -=1
                    del self.items[i]
                    del self.quantity[i]
                    del self.prices[i]
    return True
    return False
#this function to print all product in baset:
def printitems(self):
    print(self.id,"Shopping cart")
    print(f'{"Item name":<15}{"Price":<10}{"Quanity":<10}{"Total":<10}')
    print("-"*40)
    for i in range(self.noitems):
        print(f'{self.items[i]:<15}{self.prices[i]:<10}{self.quantity[i]:<10}'
              f'{self.quantity[i]*self.prices[i]:<10}')
    print("-" * 40)
    print("** total = ",self.total, ", noitems= ",self.noitems)
#create a new Basket object
cjBasket =Basket("John")
#call add function
cjBasket.add("banana",5000,2)
#call add function
cjBasket.add("milk",2000,3)
#call add function
cjBasket.add("apple",3000,1)
#call print items function
cjBasket.printitems()
#call add function with the same item (banana already input)
cjBasket.add("banana",5000,3)
#call print items function
cjBasket.printitems()
#call add function with the same item (apple already input)
cjBasket.add("apple",3000,3)
#call print items function
cjBasket.printitems()
#test delete function:
if cjBasket.delete("milk",1)==True:
    print("Delete 1 quanity of milk is successful")

```

```

cjBasket.printitems()
else:
    print("delete 1 quanlity of milk is NOT successful")
if cjBasket.delete("milk",2)==True:
    print("delete 2 quanlity of milk is successful")
    cjBasket.printitems()
else:
    print("delete 2 quanlity of milk is NOT successful")

if cjBasket.delete("milk",2)==True:
    print("delete 2 quanlity of milk is successful")
    cjBasket.printitems()
else:
    print("delete 2 quanlity of milk is NOT successful")

```

25.3. Testing and debugging

Here are some testing cases for the programming:

Test case 1: Run code create Basket object and call add method:

```

#create a new Basket object
cjBasket =Basket("John")
#call add function
cjBasket.add("banana", 5000, 2)
#call add function
cjBasket.add("milk", 2000, 3)
#call add function
cjBasket.add("apple", 3000, 1)
#call print items function
cjBasket.printitems()

```

The result:

John Shopping cart				
	Item name	Price	Quanity	Total

+	banana	5000	2	10000
-	milk	2000	3	6000
+	apple	3000	1	3000

	** total = 19000 , noitems= 3			

All products will be shown as picture above.

Test case 2: Run code and call add method with item same name “banana”:

```

#call add function with the same item (banana already input)
cjBasket.add("banana", 5000, 3)
#call print items function
cjBasket.printitems()

```

Quantity of banana will be updated,

The result:

John Shopping cart				
	Item name	Price	Quanity	Total
↑	banana	5000	5	25000
↓	milk	2000	3	
☰	apple	3000	1	
✖				
	** total = 34000 , noitems= 3			

Quantity of banana is updated to 5 (previous is 2)

Test case 3: Run code and call add method with item same name “apple”:

```
#call add function with the same item (apple already input)
cjBasket.add("apple", 3000, 3)
#call print items function
cjBasket.printitems()
```

The result:

John Shopping cart				
	Item name	Price	Quantity	
▼	banana	5000	5	
☰	milk	2000	3	
✖	apple	3000	4	12000
✖				
	** total = 43000 , noitems= 3			

Quantity of banana is updated to 4 (previous is 1)

Test case 2, 3. Quantity is updated for item. And the total is also updated

Test case 4: Run code and call delete method with item name “milk” and quantity is 1:

```
#test delete function:
if cjBasket.delete("milk", 1)==True:
    print("delete 1 quanlity of milk is successful")
    cjBasket.printitems()
else:
    print("delete 1 quanlity of milk is NOT successful")
```

The result:

John Shopping cart			
Item name	Price	Quanity	
banana	5000	5	
milk	2000	2	
apple	3000	4	12000

Test case 5: Run code and continue call delete method with item name “milk” and quantity is 2:

```
if cjBasket.delete("milk", 2)==True:
    print("delete 2 quanlity of milk is successful")
    cjBasket.printitems()
else:
    print("delete 2 quanlity of milk is NOT successful")
```

The result:

Item name	Price	Quanity	Total
banana	5000	5	25000
apple	3000	4	12000

In this test case, all quantity of milk is removed, so the Milk item is also removed from the list.

Test case 6: Run code and continue call delete method with item name “milk” and quantity is 2:

```
if cjBasket.delete("milk", 2)==True:
    print("delete 2 quanlity of milk is successful")
    cjBasket.printitems()
else:
    print("delete 2 quanlity of milk is NOT successful")
```

The result:

```
delete 2 quanlity of milk is NOT successful
```

Because milk Item is removed from the list, so we cannot remove it anymore

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise21.py

26. Exercise 22 - Word memorization helper

26.1. Learning Objectives

26.1.1. Basic learning objectives

- Random shuffle() function can be used to implement a randomization function for multiple items.
- You can register an appropriate number of words to keep in mind, and define an object class that supports memorizing the question-and-answer method for the words
- Set objects can be used to manage sets of less memorized words

26.1.2. In-depth learning objectives

- There are too many words registered in the vocabulary book. It is possible to define a class of objects that support repeating learning in a way to memorize and memorize an appropriate number of words.
- Classify words to set multiple sets (set of registered words, set of words selected for memorization, set of words that are not sufficiently memorized among words selected for memorization), and manage using set target operations such as chime set can do.

26.1.3. Problem situation

When reading English materials or studying English, there are words that you find in the dictionary because you don't know the meaning well. I think it would be nice if I could properly collect the words I looked for and memorized any number of words whenever I had time.

26.2. Problem analysis

26.2.1. The role of the computer

- Support for randomly registering pairs of spellings and meanings of English words as a learning target by memorization
- Support to memorize all registered words through random traversal question-and-answer (the activity of constructing a ‘traversal set’ with words to be learned by memorization and randomly circulating the words in the set)
- Set the cycle of random traversal Q&A (e.g. 3 rounds, 4 rounds, etc.) and update the iterative set when the end of each cycle and the next cycle starts (remove the words with correct answers in all questions and answers from the existing itinerary set)
- Guidance for checking whether the given answer is correct or incorrect for each question

26.2.2.User

- Whenever the main story of the random round question-and-answer begins, the words to memorize are printed out in the corresponding cycle.
- When the cycle of random question-and-answer starts, it supports the user to decide how many times to repeat the question-and-answer cycle.
- Guidance on correct answers when incorrect answers are given to questions about words.

26.2.3.Input

- Spelling and meaning of words to be learned by memorization
- The answer to each question (the meaning of a word) (the spelling of the word)
- Number of rounds for each random Q&A cycle

26.2.4.Output

- Input request message for words to be registered
- A list of words to cycle through
- Input request message for the number of question-and-answer rounds
- Questions about the target word for memorization learning
- Whether or not the answer to the question is correct
- A message indicating that memorization learning has ended

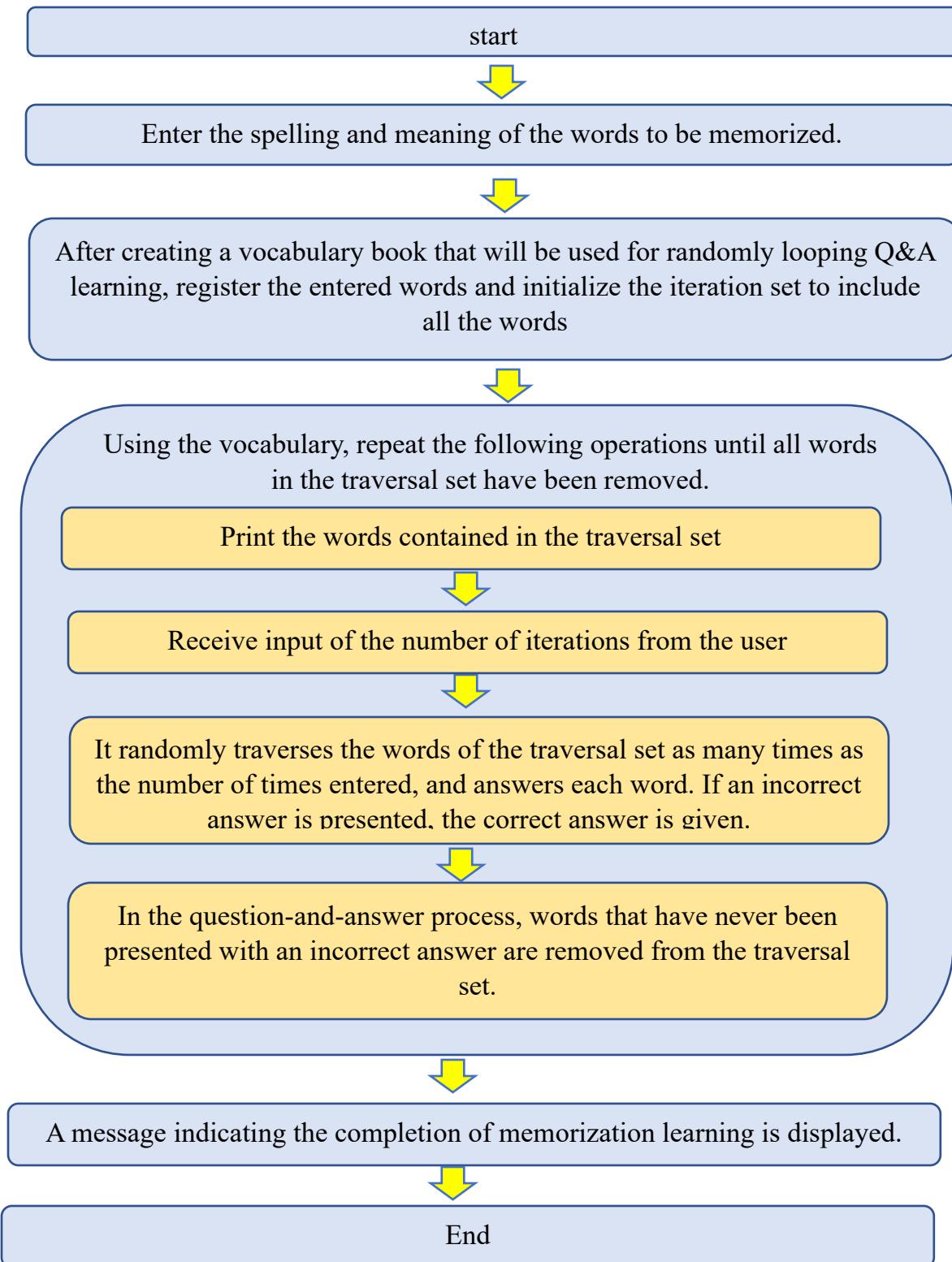
26.2.5.Data

- Persistent data: spelling and meaning of registered words
- Data updated by random traversal question-and-answer cycle: a traversal set, a set of words with incorrect answers more than once

26.3. Algorithm Efficiency

- Supports the user to set the cycle (number of question-and-answer rounds) to the most suitable value for himself/herself
- As an efficient implementation method for random traversal question-and-answer, the traversal set is composed of a list, the words on the list are randomly rearranged, and then the question-and-answer process is used in turn.

26.4. Full algorithm skeleton



26.5. Programming

26.5.1. Variables

- wdict: A dictionary of input words to learn by memorization
- voc: Vocabulary object used for managing and learning registered words
- nQNA: The number of Q&As entered by the user to set the random Q&A period.
- meaning: meaning of the word being questioned and answered
- answer: Spell the word entered by the user as an answer to the question.

26.5.2. Some explanations

- Constructor of vocabulary registration management/learning support class Vocabulary
 - self.words : a copy of the dictionary object where the registered words are stored
 - self.untrained : A set of words that are incorrect at least once during a random iterative question-and-answer cycle
- Vocabulary's method renew() : Prepares a random question-and-answer cycle of a new cycle
 - self.target : Selects as the learning target only the words that are wrong at least once among the iterative questions and answers of the previous cycle
 - Random relocation of learning words for random traversal Q&A : shuffle() function of random module

```
def __init__(self, wdict):
    self.words = wdict.copy()
    self.untrained = set(self.words)
    self.renew()

def renew(self):
    self.target = list(self.untrained)
    random.shuffle(self.target)
    self.untrained = set() #empty set
```

- Vocabulary's method check() : Checks the noon of the answer given by the user
 - Arguments: the key suggested as a question (the meaning of the word) and the value entered as an answer (the spelling of the word)
 - If the argument key is not registered in the registered word dictionary self.words, a None object is returned.
 - If the word value passed as an argument is the correct answer, an Ellipsis object is returned. If the answer is incorrect, it is added to the set self.untrained.

- If the argument value is a None object or an incorrect answer, the correct answer (self.words[key]) is returned.

```
def check(self, key, value=None):
    if key not in self.words:
        return None
    if value is not None:
        if self.words[key] == value:
            return Ellipsis
        self.untrained.add(key)
    return self.words[key]
```

- Creation of Vocabulary object and registration of target words for memorization learning
 - It receives repeated input of words (pairs of spelling and meaning) from the user, sets the meaning of each word as a key and spelling value, and stores it in a dictionary object
 - Creates a Vocabulary object with a dictionary object storing user input words as an argument

```
wdict = {} #empty dictionary
while True:
    line = input("Spelling and meaning of words to register/memorize:")
    tokens = line.split()
    if len(tokens) != 2:
        break
    wdict[tokens[1]] = tokens[0]

voc = Vocabulary(wdict)
```

- Iterative Q&A processing in one cycle: Repeat Q&A of all words as much as the value referenced by the variable **nQNA**
 - At the beginning of each traversal, the **Vocabulary** object's method **shuffle()** is called to randomly rearrange the words to be answered.
 - When the iteration question and answer of one cycle is over, the method of **Vocabulary** object is called **renew()** to update the iteration set.

```
for index in range(0, nQNA):
    voc.shuffle()
    for meaning in voc.target_keys():
        answer = input("What word means " + meaning + "?")
        word = voc.check(meaning, answer)
        if word is Ellipsis:
            print("is the correct answer")
        else:
            print("The correct answer is " + word + "!!!")
    voc.renew()
```

26.5.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

import random

class Vocabulary:
    def __init__(self, wdict):
        self.words = wdict.copy()
        self.untrained = set(self.words)
        self.renew()

    def renew(self):
        self.target = list(self.untrained)
        random.shuffle(self.target)
        self.untrained = set() #empty set

    def target_keys(self):
        return self.target

    def check(self, key, value=None):
        if key not in self.words:
            return None
        if value is not None:
            if self.words[key] == value:
                return Ellipsis
            self.untrained.add(key)
        return self.words[key]

    def shuffle(self):
        random.shuffle(self.target)

wdict = {} #empty dictionary
while True:
    line = input("Spelling and meaning of words to register/memorize:")
    tokens = line.split()
    if len(tokens) != 2:
        break
    wdict[tokens[1]] = tokens[0]

voc = Vocabulary(wdict)
while len(voc.target_keys()) > 0:
    print("\nwords to memorize:", voc.target_keys())
    nQNA = int(input("Number of questions and answers per word:"))
    print()
    for index in range(0, nQNA):
        voc.shuffle()
        for meaning in voc.target_keys():
            answer = input("What word means " + meaning + "?")
```

```
word=voc.check(meaning,answer)
if word is Ellipsis:
    print("is the correct answer")
else:
    print("The correct answer is"+word+"")
voc.renew()
print("\nWord learning is over")
```

26.6. Testing and debugging

Run program.

Test case 1

```
run: Exercise22
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Ex
Spelling and meaning of words to register/memorize:test 시험
Spelling and meaning of words to register/memorize:error 오류
Spelling and meaning of words to register/memorize:modify 수정하다
Spelling and meaning of words to register/memorize:

words to memorize: ['수정하다', '오류', '시험']
Number of questions and answers per word:2

What word means '오류'?error
is the correct answer
What word means '시험'?modify
The correct answer is'test'
What word means '수정하다'?modfy
The correct answer is'modify'
What word means '오류'?test
The correct answer is'error'
What word means '수정하다'?modfy
is the correct answer
What word means '시험'?test
is the correct answer

words to memorize: ['오류', '수정하다', '시험']
Number of questions and answers per word:1
```

Test case 2:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/Books/PythonNangCao/Exercise_Par  
Spelling and meaning of words to register/memorize:Father Bo  
Spelling and meaning of words to register/memorize:Mother Me  
Spelling and meaning of words to register/memorize:Fan Quat  
Spelling and meaning of words to register/memorize:  
  
words to memorize: ['Me', 'Quat', 'Bo']  
Number of questions and answers per word:3  
  
What word means 'Me'?Mother  
is the correct answer  
What word means 'Bo'?Father  
is the correct answer  
What word means 'Quat'?PHANG  
The correct answer is 'Fan'  
What word means 'Me'?Mother  
is the correct answer  
What word means 'Bo'?Father  
is the correct answer  
What word means 'Quat'?Fan  
is the correct answer  
What word means 'Quat'?Fan  
is the correct answer  
What word means 'Bo'?Father  
is the correct answer  
What word means 'Me'?Mother  
is the correct answer  
  
words to memorize: ['Quat']  
Number of questions and answers per word:
```

Test Case 2 Quat is wrong 1 time. So next time go to test case 3 (you should do yourself)

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise22.py

27. Exercise 23 - Word memorization helper (improve)

27.1. Problem situation

If there are too many registered words (e.g., more than 5), please refer to the functions presented below in order to support the learning of English vocabulary memorization more effectively by applying the method below. Utilizing the word memorization program (class Vocabulary, etc.) complemented. Let's improve

Applicable method: A separate set of 'not memorized' words among the words registered in the Vocabulary object (the question and answer has not been completed at the present time since registration, or the word that is incorrect at least once in the most recent periodic question and answer) Saved in object self.unmemorized. At the beginning of the loop Q&A cycle, an appropriate number of words (e.g., 5) are randomly selected from among unmemorized words to form a circuit set. When the cycle of Q&A cycles, the words that answered all the Q&A of the cycle with correct answers are excluded from the list of words to be memorized.

Reference utilization function

-random.sample(sequence,k): Randomly selects k items among the items constituting the sequence. Returns the composed list object. A list object, a tuple object, a string object, a set object, etc. can be given as a sequence.

-Difference operator '-': When there are set objects a and b, performing the expression 'a-b' creates a set object consisting of all elements that belong to set a and do not belong to set b.

27.2. Programming-the whole program

These codes I improved from exercise 22: Improved the word memorization program if there many vocabularies to learn, take 5 sample vocabularies to learn

1. Update Vocabulary class

1.1 use random.sample (see takeVocabularyToLearn(self) function)

1.2 Difference operator '-' (see updateUnmemorized(self) functin)

1.3 updated 2 new attribute: self.unmemorized (any vocabularies is unmemorized),
and self.memorized (for memorized for each periodic

1.4 update check to save memorized word or not

1.5 static method buildVocabularyDatabase **to read vocabularies from Excel** (many vocabularies)

1.6 method printAllVocabularies(self) -> print all original vocabularies

1.7 method printStatusProgressingLearning(self)-> show status progressing learning

2. Create a new class VocabularyExecutor

this class use to create 4 option menuitem

2.1 Overview Vocabulary -> show all vocabularies from Vocabulary Object

2.2 Re-Learn Vocabulary -> learners can re learn vocabularies

2.3 Continue learning Vocabulary -> learners can continue to learn vocabulary

It means: learner can learn vocabulary day by day, can stop software and continue learn from history

program saved vocabulary to object file and restored for learner to continue to learn

2.4 Exit program

database for Vocabularies (from Excel structure):

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise23/Database_Vocabulary.xlsx

This is some example list English and Korean language:

	A	B
1	English Language	Korean Language
2	refrigerator	냉장고
3	bag	가방
4	table	책상
5	chair	의자
6	book	책
7	notebook	공책
8	pencil	연필
9	pen	볼펜
10	hat	모자
11	t-shirt	티셔츠
12	pants	바지
13	apple	사과
14	grape	포도
15	juice	주스
16	hotdog	핫도그
17	meat	고기
18	milk	우유
19	water	물
20	school	학교
21	student	학생
22	teacher	선생님
23	professor	교수님
24	student hall	학생회관
--		

Because we have to test with many vocabularies so it is difficult to enter from the program. I create database in excel. And coding to read all vocabularies from excel.

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

#Description:
#These codes I improved from Exercise 22.
#Improved the word memorization program if there many vocabularies to learn, take 5 sample vocabularies
to learn
#1.Update Vocabulary class
# 1.1 use random.sample (see takeVocabularyToLearn(self) function)
# 1.2 Difference operator '-'(see updateUnmemorized(self) functin)
# 1.3 updated 2 new attribute: self.unmemorized (any vocabularies is unmemorized),
#           and self.memorized (for memorized for each periodic
# 1.4 update check to save memorized word or not
# 1.5 static method buildVocabularyDatabase to read vocabularies from Excel (many vocabularies)
# 1.6 method printAllVocabularies(self) -> print all original vocabularies
# 1.7 method printStatusProgressingLearning(self)-> show status progressing learning
#2. Create a new class VocabularyExecutor
# this class use to create 4 option menuitem
# 2.1 Overview Vocabulary -> show all vocabularies from Vocabulary Object
# 2.2 Re-Learn Vocabulary -> learners can re learn vocabularies
# 2.3 Continue learning Vocabulary -> learners can continue to learn vocabulary
#     It means: learner can learn vocabulary day by day, can stop software and continue learn from history
#     program saved vocabulary to object file and restored for learner to continue to learn
# 2.4 Exit program
import pickle
import random
import pandas as pd

class Vocabulary:
    def __init__(self, wdict):
        self.words = wdict.copy()
        self.untrained = set(self.words)
        self.unmemorized = self.untrained.copy()
        self.memorized = set() #variable save to correct vocabulary for each periodic

    #this function to get sample 5 vocabulary in the unmemorized list
    def takeVocabularyToLearn(self):
        #difference unmemorized and memorized
        if len(self.unmemorized) <= 5:
            self.target = list(self.unmemorized)
        else:
            self.target = random.sample(self.unmemorized, 5)
        random.shuffle(self.target)
        self.memorized = set() #reset correct vocabulary for feach periodic
        self.untrained = set() # empty set

    #this function use to difference operator to get the unmemorized vocabulary
```

```

def updateUnmemorized(self):
    self.unmemorized = self.unmemorized - self.memorized

def target_keys(self):
    return self.target

def check(self, key, value=None):
    if key not in self.words:
        return None
    if value is not None:
        if self.words[key] == value:
            #saved memorized if learning correcting
            #at least one time is unmemorized, memorized doesn't save the word
            if self.untrained.__contains__(key) == False:
                self.memorized.add(key)
            return Ellipsis
        self.untrained.add(key)
        #remove again memorized for next time if learning wrongting
        #eg: 1st is memorized, but 2rd is wrongunmemorized-> total is unmemorized
        if self.memorized.__contains__(key):
            self.memorized.remove(key)
    return self.words[key]

def shuffle(self):
    random.shuffle(self.target)
#this static method used to build vocabulary database from Excel
@staticmethod
def buildVocabularyDatabase():
    wdict = {} # empty dictionary
    data = pd.read_excel(r'data_exercise23/Database_Vocabulary.xlsx') # read data
    df = pd.DataFrame(data, columns=['English Language', 'Korean Language'])
    for i in range(len(df.index)):
        english = df.iloc[i]['English Language']
        korean = df.iloc[i]['Korean Language']
        wdict[korean] = english
    voc = Vocabulary(wdict)
    return voc

#this function use to print all orginial vocabularies
def printAllVocabularies(self):
    print("There are ", len(self.words), " vocabularies")
    print(f'{len(self.words)} English Language:{<20}{len(self.words)} Korean Language:{<20}')
    print("-" * 40)
    for k, v in self.words.items():
        print(f'{v:{<20}}{k:{<20}}')
    print("-" * 40)

#this function use to print all status progressing learning
def printStatusProgressingLearning(self):
    print("There are ", len(self.words), " vocabularies")
    memorizeds=set(self.words)-self.unmemorized
    print("You have not yet memorized ", len(self.unmemorized), " vocabularies")
    print("You already memorized ", len(memorizeds), " vocabularies")

#this class use to parse vocabulary from Excel File

```

```

#and return Vocabulary Object
class VocabularyExecutor:
    voc=None
    filename = "vocabulary.dat"
    #this function give 4 option menuitem
    def runVocabularyExecutor(self):
        while True:
            print("1.Overview Vocabulary")
            print("2.Re-Learn Vocabulary")
            print("3.Continue learning Vocabulary")
            print("4.Exit program")
            choose=input("Please choose [1..4]:")
            if choose=="1":
                self.overviewVocabulary();
            elif choose=="2":
                self.reLearnVocabulary()
            elif choose=="3":
                self.continueLearnVocabulary()
            elif choose=="4":
                break
        print("Thank you so much for your using programming!")

#this function: Learner will learn vocabulary
def learnVocabulary(self):
    #before learning, program will show the status progressing learning
    self.voc.printStatusProgressingLearning()
    while len(self.voc.unmemorized)>0:
        self.voc.takeVocabularyToLearn()
        #if there are unmemorized vocabulary to learn
        if len(self.voc.target_keys())>0:
            print("\nWords to memorize:",self.voc.target_keys())#Words to memorize:
            nQNA=int(input("Number of questions and answers by word:"))#
            print()
            for index in range(0,nQNA):
                self.voc.shuffle()
                for meaning in self.voc.target_keys():
                    answer=input("What is the meaning of "+meaning+"?")#What word means """+meaning+"?"
                    word=self.voc.check(meaning,answer)
                    if word is Ellipsis:
                        print(""+answer+" is correct")
                    else:
                        print("Wrong! "+meaning+" has meaning "+word+"")#"The correct answer is "+word+""
            print("You finihed a periodic learning")
        # Words unmemorized:
        if len(self.voc.untrained) > 0:
            print("Words unmemorized:", self.voc.untrained)
        else:
            print("Words unmemorized:0")
        #words memoried
        if len(self.voc.memorized)>0:
            print("Words memorized:", self.voc.memorized)
        else:
            print("Words memorized:0")
        #update the unmemorized word
        self.voc.updateUnmemorized()

```

```

#save status progressing learning to file
self.persistenceVocabulary()
#when finishing a periodic, we ask learner to continue or not learning
choose=input("Do you want to continue learning?(y/n):")
if choose=="N" or choose=="n":
    break
if len(self.voc.unmemorized)==0:
    print("Congratulations, you memorized all vocabulary")
    self.voc.printAllVocabularies()
else:
    #print status progressing learning again when user paused learning
    self.voc.printStatusProgressingLearning()

#this function use to overview vocabularies
def overviewVocabulary(self):
    if self.voc is None:
        self.voc = Vocabulary.buildVocabularyDatabase()
    if self.voc is not None:
        self.voc.printAllVocabularies()

# this function use to open Vocabulary object from file
# learner can continue learning from history
def continueLearnVocabulary(self):
    try:
        with open(self.filename, 'rb') as input:
            self.voc = pickle.load(input)
    except FileNotFoundError:
        self.voc = Vocabulary.buildVocabularyDatabase()
    self.learnVocabulary()

#this function: leaner re learns vocabulary
#rebuild vocabulary database
def reLearnVocabulary(self):
    self.voc = Vocabulary.buildVocabularyDatabase()
    self.learnVocabulary()

#this function use to save the vocabularies and progressing learning to file
#so we can restore to continue learning
def persistenceVocabulary(self):
    with open(self.filename, 'wb') as output: # Overwrites any existing file.
        pickle.dump(self.voc, output, pickle.HIGHEST_PROTOCOL)

#create a VocabularyExecutor object
vexecutor=VocabularyExecutor()
#call runVocabularyExecutor to start program
vexecutor.runVocabularyExecutor()

```

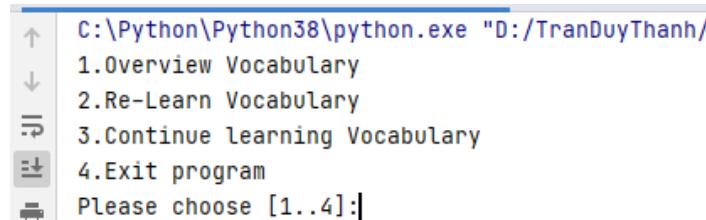
27.3. Testing and debugging

Here are some testing cases for the programming:

Test case 1: Run code create VocabularyExecutor object and call runVocabularyExecutor method:

```
#create a VocabularyExecutor object
vexecutor=VocabularyExecutor()
#call runVocabularyExecutor to start program
vexecutor.runVocabularyExecutor()
```

The result with 4 option menu items:



Test case 2: Choose “1.Overview Vocabulary” option:

```
Please choose [1..4]:1
There are 23 vocabularies
English Language      Korean Language
-----
refrigerator          냉장고
bag                   가방
table                 책상
chair                의자
book                 책
notebook              공책
pencil                연필
pen                   볼펜
hat                   모자
t-shirt               티셔츠
pants                바지
apple                사과
grape                포도
juice                주스
hotdog               핫도그
meat                 고기
milk                 우유
water                물
school               학교
student              학생
teacher              선생님
professor             교수님
student hall          학생회관
-----
1.Overview Vocabulary
2.Re-Learn Vocabulary
3.Continue learning Vocabulary
4.Exit program
Please choose [1..4]:
```

All vocabularies in Excel file “Database_Vocabulary.xlsx” will be loaded in memory to Vocabulary object and shown for learner to see (overview only).

The option 2, 3 → learner can practice to learn vocabulary

Test case 3: Choose “2.Re-Learn Vocabulary” option:

```

1.Overview Vocabulary
2.Re-Learn Vocabulary
3.Continue learning Vocabulary
4.Exit program
Please choose [1..4]:2
There are 23 vocabularies
You have not yet memorized 23 vocabularies
You already memorized 0 vocabularies

Words to memorize: ['책상', '학교', '책', '공책', '학생회관']
Number of questions and answers by word:|

```

Program will pick 5 vocabularies from list (using random.sample method). Now learner can enter the number of question and answers by word, eg: 2

```

Words to memorize: ['책상', '학교', '책', '공책', '학생회관']
Number of questions and answers by word:2

What is the meaning of '책상'?table
'table' is correct
What is the meaning of '학생회관'?학생회관
Wrong! '학생회관' has meaning 'student hall'
What is the meaning of '학교'?people
Wrong! '학교' has meaning 'school'
What is the meaning of '공책'?notebook
'notebook' is correct
What is the meaning of '책'?book
'book' is correct
What is the meaning of '학교'?mouse
Wrong! '학교' has meaning 'school'
What is the meaning of '공책'?notebook
'notebook' is correct
What is the meaning of '책'?book
'book' is correct
What is the meaning of '책상'?table
'table' is correct
What is the meaning of '학생회관'?student hall
'student hall' is correct
You finished a periodic learning
Words unmemorized: {'학교', '학생회관'}
Words memorized: {'공책', '책상', '책'}
Do you want to continue learning?(y/n):|

```

After testing with some vocabularies

We have some results:

Words unmemorized: {'학교', '학생회관'} → because at least we enter wrong 1 time

Words memorized: {'공책', '책상', '책'} → right anytime

Test case 4: Say YES(y) to continue learning

Words unmemorized: {'학교', '학생회관'}

Words memorized: {'공책', '책상', '책'}

Do you want to continue learning?(y/n):**y**

Words to memorize: ['학교', '연필', '냉장고', '볼펜', '바지']

Number of questions and answers by word:**1**

Program will show 5 another vocabulary: Only word is wrong from previous periodic + new vocabularies. But program pick randomizes vocabularies.

This time we enter number of questions and answers by word is 1

Words to memorize: ['학교', '연필', '냉장고', '볼펜', '바지']

Number of questions and answers by word:**1**

What is the meaning of '학교'?**school**

'school' is correct

What is the meaning of '바지'?**pants**

'pants' is correct

What is the meaning of '냉장고'?**refrigerator**

'refrigerator' is correct

What is the meaning of '볼펜'?**pen**

'pen' is correct

What is the meaning of '연필'?**pencil**

'pencil' is correct

You finished a periodic learning

Words unmemorized: 0

Words memorized: {'바지', '볼펜', '냉장고', '학교', '연필'}

Do you want to continue learning?(y/n):**|**

This case 5 vocabularies is memorized.

Now we enter NO (n) to break the function

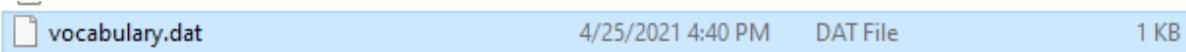
```
Words memorized: {'바지', '볼펜', '냉장고', '학교', '연필'}  
Do you want to continue learning?(y/n):n  
There are 23 vocabularies  
You have not yet memorized 15 vocabularies  
You already memorized 8 vocabularies  
1.Overview Vocabulary  
2.Re-Learn Vocabulary  
3.Continue learning Vocabulary  
4.Exit program  
Please choose [1..4]:
```

In this case:

You have not yet memorized 15 vocabularies

You already memorized 8 vocabularies

For each periodic learning, Vocabulary object will be stored “vocabulary.dat”:



This file will be used for “3.Continue learning Vocabulary”

Test case 5: Press “4.Exit program” to exit program

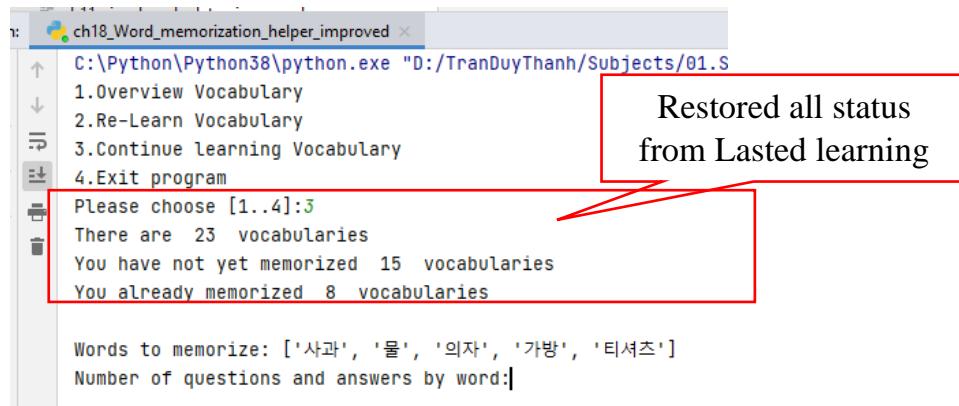
Lasted status
progressing learning

```
There are 23 vocabularies  
You have not yet memorized 15 vocabularies  
You already memorized 8 vocabularies  
1.Overview Vocabulary  
2.Re-Learn Vocabulary  
3.Continue learning Vocabulary  
4.Exit program  
Please choose [1..4]:4  
Thank you so much for your using programming!
```

Program will be stopped and say thank you.

Test case 6: Re run application, and Press “3.Continue learning Vocabulary”

Program will be continued to learn vocabulary. Because Vocabulary object will be stored in the file. In this case, program read Vocabulary object from file and restore all status from lasted history→it is easily for learner to learn Vocabulary.



Now Learner continue to learn.

```
There are 23 vocabularies
You have not yet memorized 15 vocabularies
You already memorized 8 vocabularies

Words to memorize: ['사과', '물', '의자', '가방', '티셔츠']
Number of questions and answers by word:1

What is the meaning of '티셔츠'?t-shirt
't-shirt' is correct
What is the meaning of '물'?grape
Wrong! '물' has meaning 'water'
What is the meaning of '사과'?apple
'apple' is correct
What is the meaning of '가방'?bag
'bag' is correct
What is the meaning of '의자'?chair
'chair' is correct
You finihsed a periodic learning
Words unmemorized: {'물'}
Words memorized: {'의자', '사과', '티셔츠', '가방'}
Do you want to continue learning?(y/n):
```

Now say NO (n), you can see the summary:

```
You finihsed a periodic learning
Words unmemorized: {'물'}
Words memorized: {'의자', '사과', '티셔츠', '가방'}
Do you want to continue learning?(y/n):n
There are 23 vocabularies
You have not yet memorized 11 vocabularies
You already memorized 12 vocabularies
```

Test case 7: Press “4.Exit program” to exit program

```
Do you want to continue learning?(y/n):n
There are 23 vocabularies
You have not yet memorized 11 vocabularies
You already memorized 12 vocabularies
1.Overview Vocabulary
2.Re-Learn Vocabulary
3.Continue learning Vocabulary
4.Exit program
Please choose [1..4]:4
Thank you so much for your using programming!
```

We can test learning vocabulary more time by re run application!

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise23.py

28. Exercise 24 – Rock Paper Scissors Game

28.1. Learning Objectives

28.1.1. Basic learning objectives

- You can change the shape of a Turtle to an image
- You can write event handling functions for keystroke events
- Computers and humans can calculate the winner

28.1.2. In-depth learning objectives

- You can play the rock-paper-scissors game repeatedly.
- It is possible to understand the characteristics of the write() function among Turtle commands.

28.1.3. Problem situation

I'm trying to program so that I can use a computer, scissors, rock, and paper. I'm curious how to make my choices using the keys r(rock), s(scissors), and p(paper) on the keyboard, and let the computer randomly select r, s, and p. Let's change the shape of the turtle into scissors, rock, and paper, and program the image so that it can be accurately displayed on the screen according to the computer's choice.

28.2. Problem analysis

28.2.1. The role of the computer

- Replace the turtle's image with a user-specified image
- Recognize user's keyboard keystrokes
- Make the corresponding function run when a specific key is pressed

28.2.2. User

- Think of ways to quickly communicate computer choices to users.
- Define a key that can easily represent the user's choice of scissors, rock, or paper

28.2.3. Input

- Scissors:s
- rock: r
- bo:p

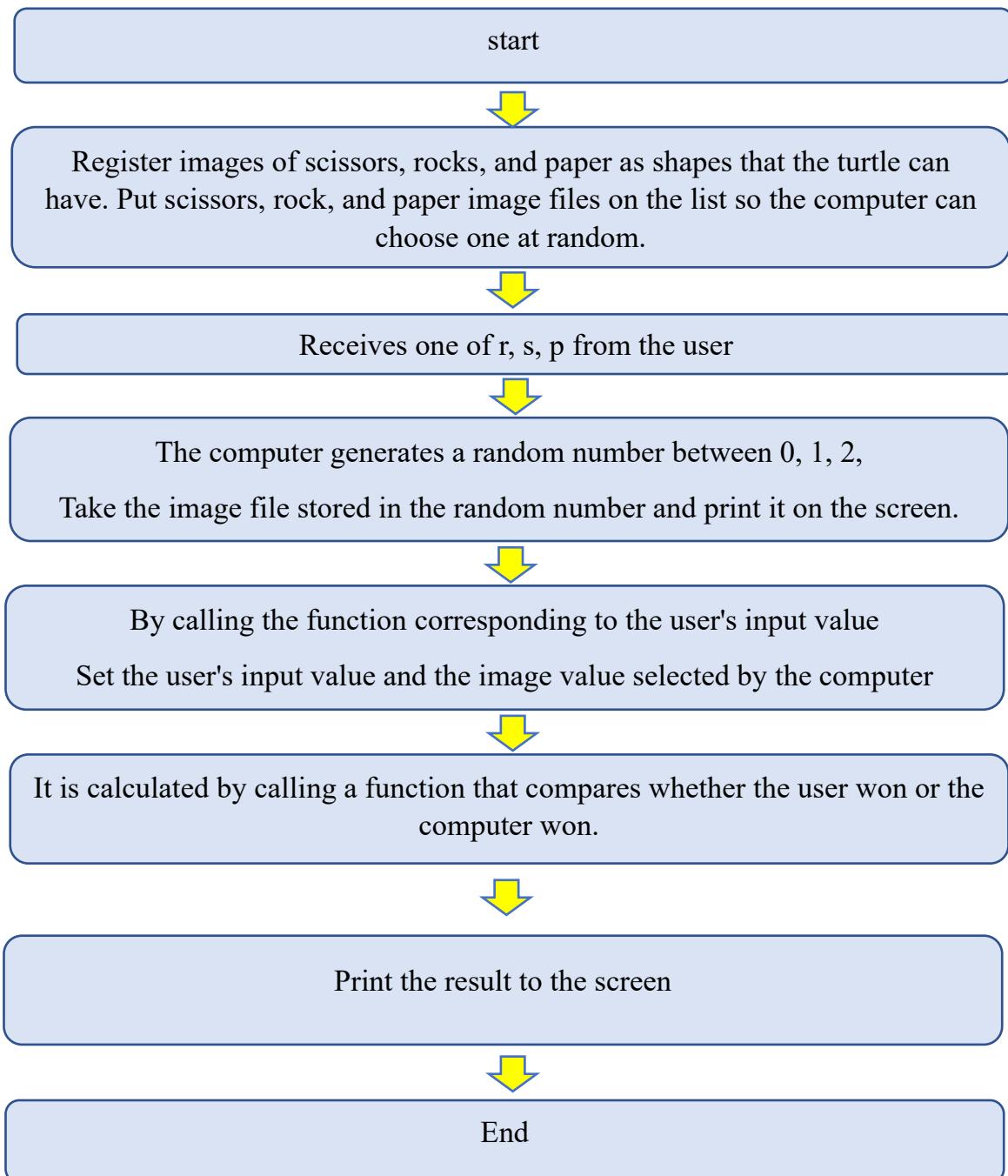
28.2.4. Output

- An image of one of scissors, rock, and paper for a selection of computers and the winner

28.2.5.Data

- A list of turtle shapes

28.3. Full algorithm skeleton



28.4. Programming

28.4.1. Variables

- s: A list to save the file names of the scissors, rock, and paper images used by the computer. The image file format must be gif.
- cno: a random number chosen by the computer among 0, 1, 2
- myno: 0, 1, 2, selected by the user (paper:0, scissors:1, rock:2, paper:0)
- result: Variable for calculating the winner (Scissors <rock <paper <scissors)

28.4.2. Some explanations

Methods of importing modules:

```
import turtle
import turtle as t
from turtle import *
```

How to register multiple images of Turtle

here is the Link of images:

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise24

```
s=["data_exercise24/paper_machine.gif",
 "data_exercise24/scissor_machine.gif",
 "data_exercise24/rock_machine.gif"]
#Set the size of the screen
t.setup(300,300)
for img in s:
    #Register image to be #turtle shape
    t.addshape(img)

function show_result():
```

```
def show_result(myno,cno):
    t.shape(s[cno])
    # Calculating the Win
    result=myno-cno
    msg=""
    if result ==2:
        # I rock, look at the computer, I'm Jim
        result =-1
    elif result ==-2:
        # I see, the computer rocks. I win
        result=1
    if result ==0:
        print("A draw with a computer. Do it again.")
        msg="A draw with a computer. Do it again."
    elif result <0: #result =-1
        print("You lost")
        msg="You lost"
    else: #result =1
        #You won
```

```

print("You won")
msg = "You won"
turtle.write(msg, False, align="center")

```

0 is paper, 1 is scissors, and 2 is rock. Actually, 0 (beam) < 1 (scissors) < 2 (rock). Since the beam beats the rock, result = myno - cno to find the difference results in -2 or 2. If the result value is 1, it is a win, 0 is a draw, -1 is a loss, 2 is a loss, -2 is a win case

Turtle's onkeypress() function:

```

#I make rocks
t.onkeypress(rock, 'r')
#I make sissor
t.onkeypress(scissor,'s')
#I make paper
t.onkeypress(paper,'p')

```

Confirmation of scissors, rock, and paper:

```

def rock():
    cno=randint(0,2)
    myno=2
    print("Your choice is [rock]",end="")
    show_result(myno,cno)
def scissor():
    cno=randint(0,2)
    myno=1
    print("Your choice is [scissors].",end="")
    show_result(myno,cno)
def paper():
    cno=randint(0,2)
    myno=0
    print("Your choice is [paper]",end="")
    show_result(myno, cno)

```

28.4.3.Programming-the whole program

The whole program is shown as below.

```

#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

import turtle
import turtle as t
from turtle import *

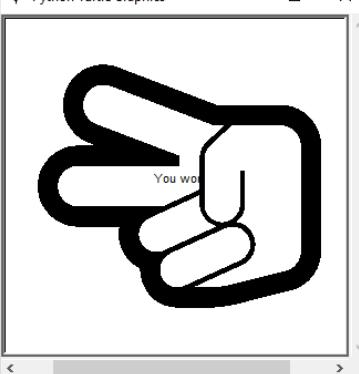
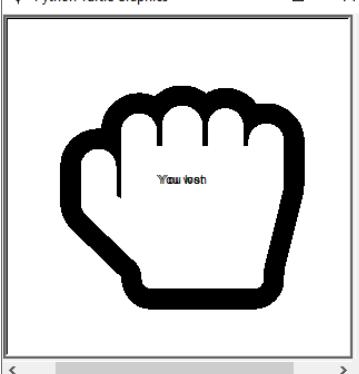
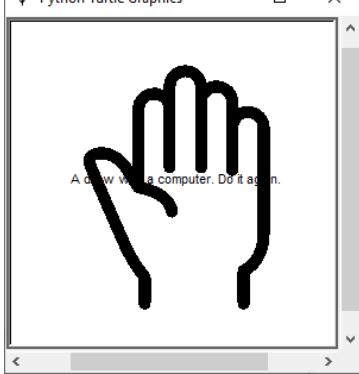
from random import randint
s=[ "data_exercise24/paper_machine.gif",
  "data_exercise24/scissor_machine.gif",
  "data_exercise24/rock_machine.gif"]
#Set the size of the screen

```

```
t.setup(300,300)
for img in s:
    #Register image to be #turtle shape
    t.addshape(img)
print("If you want to play rock paper scissors, press s for scissors, r for rock, and p for jaws.")
def show_result(myno,cno):
    t.shape(s[cno])
    # Calculating the Win
    result=myno-cno
    msg=""
    if result ==2:
        # I rock, look at the computer, I'm Jim
        result =-1
    elif result ==-2:
        # I see, the computer rocks. I win
        result=1
    if result ==0:
        print("A draw with a computer. Do it again.")
        msg="A draw with a computer. Do it again."
    elif result <0: #result =-1
        print("You lost")
        msg="You lost"
    else: #result =1
        #You won
        print("You won")
        msg = "You won"
    turtle.write(msg, False, align="center")
def rock():
    cno=randint(0,2)
    myno=2
    print("Your choice is [rock]",end="")
    show_result(myno,cno)
def scissor():
    cno=randint(0,2)
    myno=1
    print("Your choice is [scissors].",end="")
    show_result(myno,cno)
def paper():
    cno=randint(0,2)
    myno=0
    print("Your choice is [paper]",end="")
    show_result(myno, cno)
#I make rocks
t.onkeypress(rock, 'r')
#I make sissor
t.onkeypress(scissor,'s')
#I make paper
t.onkeypress(paper,'p')
t.listen()
t.mainloop()
```

28.5. Testing and debugging

Run program and see the results:

Test case 1: Press r Key.	Test case 2: Press s Key	Test case 3: Press p Key.
		
Your choice is [rock] You won	Your choice is [scissors]. You lost	Your choice is [paper] A draw with a computer. Do it again.

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise24.py

29. Exercise 25 - Rock Paper Scissors (improve)

29.1. Problem situation

Instead of using the result of the game as the print() function, let's program it using the function write() defined in the Turtle module. Also, let's play the game with 2 wins of 3 games.

29.2. Programming-the whole program

I improved exercise 24, updated new code by using write() method of Turtle for **Rock Paper Scissors Game Machine**

- 1.Create class GameGUI
- 2.Update GUI for turtle
 - 2.1. All the results are drawn on the GUI
 - 2.2. Load image rock, scissors, paper for Computer and human
3. Show the result for each playing

Here is the link of images:

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise25

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

#Description:
#These codes I improved from Exercise24.
#Improved the rock-scissors - paper
#1.create class GameGUI
#2.Update GUI for turtle
# 2.1. All the results are drawn on the GUI
# 2.2. Load image rock, scissors, paper for Computer and human
#3. Show the result for each playing
import turtle
from random import randint
class GameUI:
    #list to store image for Machine
    arrImageMachine = ["data_exercise25/paper_machine.gif",
                      "data_exercise25/scissor_machine.gif",
                      "data_exercise25/rock_machine.gif"]
    # list to store image for Human
    arrImageHuman = ["data_exercise25/paper_human.gif",
                    "data_exercise25/scissor_human.gif"]
```

```
"data_exercise25/rock_human.gif"]  
  
# turtle_machine: draw image for machine  
turtle_machine = None  
# turtle_human: draw image for human  
turtle_human = None  
# turtle_machine_label: draw label image for machine  
turtle_machine_label = None  
# turtle_human_label: draw label image for human  
turtle_human_label = None  
# turtle_result: draw result between human and machine  
turtle_result = None  
def __init__(self):  
    self.turtle_machine = turtle.Turtle()  
    self.turtle_human = turtle.Turtle()  
    self.turtle_machine_label = turtle.Turtle()  
    self.turtle_human_label = turtle.Turtle()  
    self.turtle_result = turtle.Turtle()  
  
    turtle.title("Rock Paper Scissors Game")  
    #Set the size of the screen  
    turtle.setup(800, 600)  
    self.registerImage(self.arrImageMachine)  
    self.registerImage(self.arrImageHuman)  
  
    # Register image to be #turtle shape  
    def registerImage(self,arrImage):  
        for img in arrImage:  
            turtle.addshape(img)  
  
    #this function converts number to label of rock, paper, scissors  
    def getLabel(self,num):  
        if num ==0:  
            return "[paper]"  
        if num ==1 :  
            return "[scissor]"  
        else:  
            return "[rock]"  
    #this function use to show Welcome and help gamer  
    def welcome(self):  
        msg=""">  
        Welcome to Rock paper Scissors Game!  
        Please press [r] or [p] or [s]  
        r ->rock  
        p ->paper  
        s ->scissor  
        """  
        self.turtle_result.reset()  
        self.turtle_result.penup()  
        style = ('batang', 20, 'italic')  
        self.turtle_result.setposition(0,100)  
        self.turtle_result.color('blue')  
        self.turtle_result.write(msg, move=False, font=style, align="center")  
        self.turtle_result.hideturtle()
```

```
#this function use to draw label
def drawLabel(self,turtleItem,msg,x,y,color,style,align):
    turtleItem.reset()
    turtleItem.penup()
    turtleItem.setposition(x, y)
    turtleItem.color(color)
    turtleItem.write(msg, move=False, font=style, align=align)
    turtleItem.hideturtle()

#this function use to:
#show human and computer result
def show_result(self,myno,cno):
    self.turtle_machine.penup()
    self.turtle_machine.goto(100, 0)
    self.turtle_machine.shape(self.arrImageMachine[cno])
    self.turtle_human.penup()
    self.turtle_human.goto(-200,0)
    self.turtle_human.shape(self.arrImageHuman[myno])
    # Calculating the Win
    result=myno-cno
    msg=""
    if result==2:
        # I rock, look at the computer
        result=-1
    elif result ==-2:
        # I see, the computer rocks. I win
        result=1
    if result ==0:
        msg="A draw with a computer. Do it again"
    elif result <0: #result =-1
        #You lost
        msg="You lost!"
    else: #result =1
        msg = "You won!"
    msgHuman="Your choice "+self.getLabel(myno)
    msgMachine="Computer selects "+self.getLabel(cno)
    style = ('batang', 15, 'italic')
    self.drawLineLabel(self.turtle_human_label,msgHuman,-200, -150,'red',style,"center")
    self.drawLineLabel(self.turtle_machine_label, msgMachine, 100, -150, 'red', style, "center")
    style = ('batang', 20, 'italic')
    self.drawLineLabel(self.turtle_result, msg,0,200, 'blue', style, "center")

#this function is shown paper
def paper(self):
    cno=randint(0,2)
    myno=0
    #Your choice is [paper]
    self.show_result(myno, cno)

# this function is shown scissors
def scissor(self):
    cno=randint(0,2)
    myno=1
    #Your choice is [scissors].
    self.show_result(myno,cno)

# this function is shown rock
```

```

def rock(self):
    cno=randint(0,2)
    myno=2
    #Your choice is [rock]
    self.show_result(myno,cno)

# this function is listening the event
def eventListener(self):
    #I make rocks
    turtle.onkeypress(self.rock, 'r')
    turtle.onkeypress(self.rock, 'R')

    #I make sissor
    turtle.onkeypress(self.scissor,'s')
    turtle.onkeypress(self.scissor,'S')

    #I make paper
    turtle.onkeypress(self.paper,'p')
    turtle.onkeypress(self.paper,'P')

    turtle.listen()
def showGameUI(self):
    self.welcome()
    self.eventListener()
    turtle.mainloop()

#create game object
game=GameUI()
#call showGameUI method to start a game
game.showGameUI()

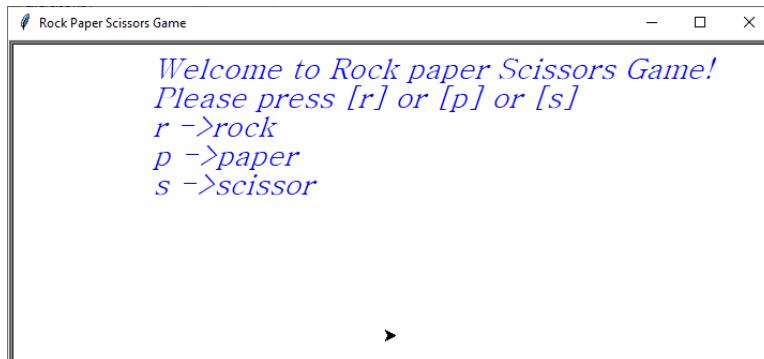
```

29.3. Testing and debugging

Here are some testing cases for the programming:

Test case 1:

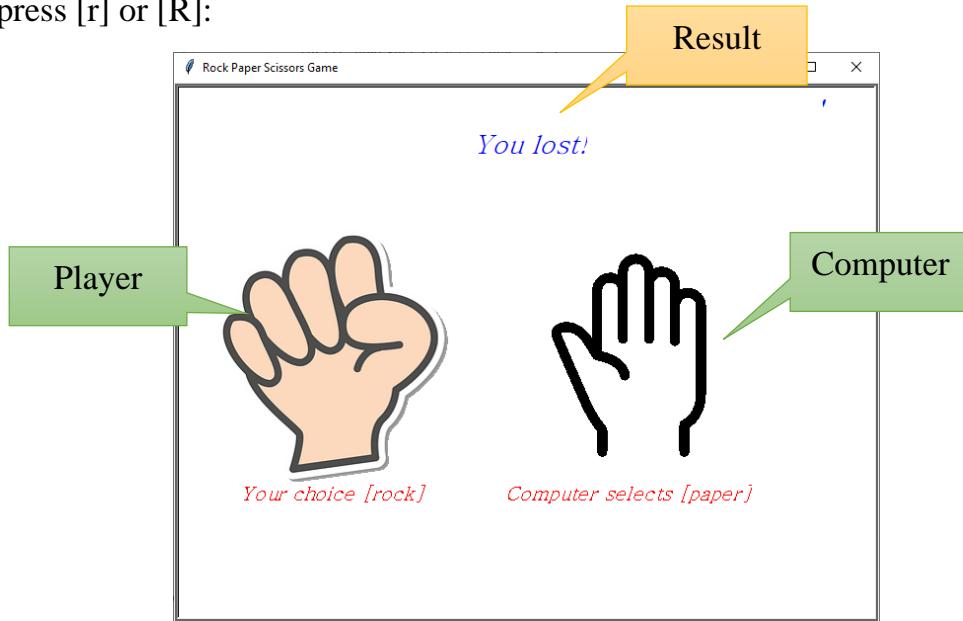
Run game.showGameUI(). The first time, program will show the Welcome Screen and some instructions for player:



Player can press r, R, p, P, s, S to choose rock, paper or scissors. Program will show image for human and computer random image on the screen.

Test case 2:

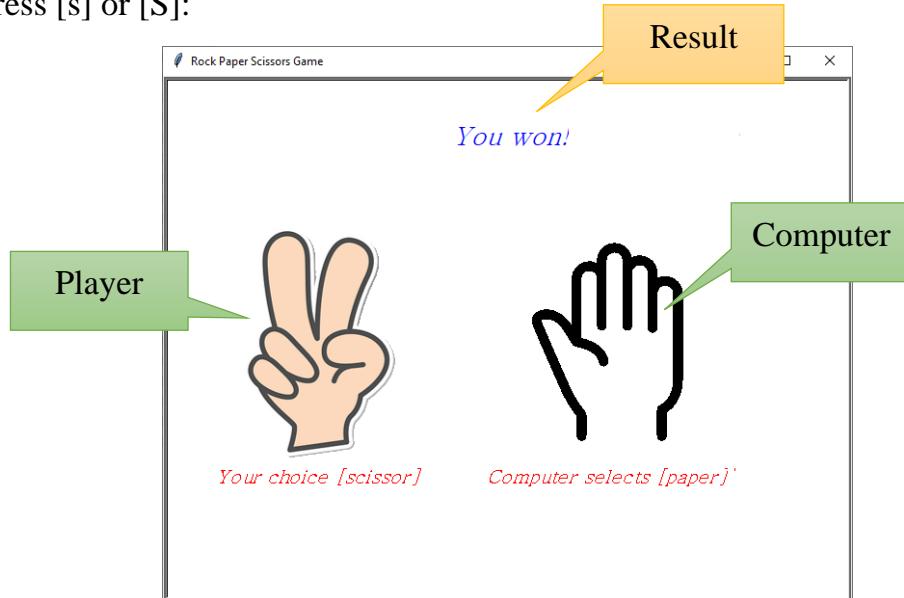
Player press [r] or [R]:



Player press [r] [R]. program will show Rock for player. And now random computer is Paper. In this case, player is lost!

Test case 3:

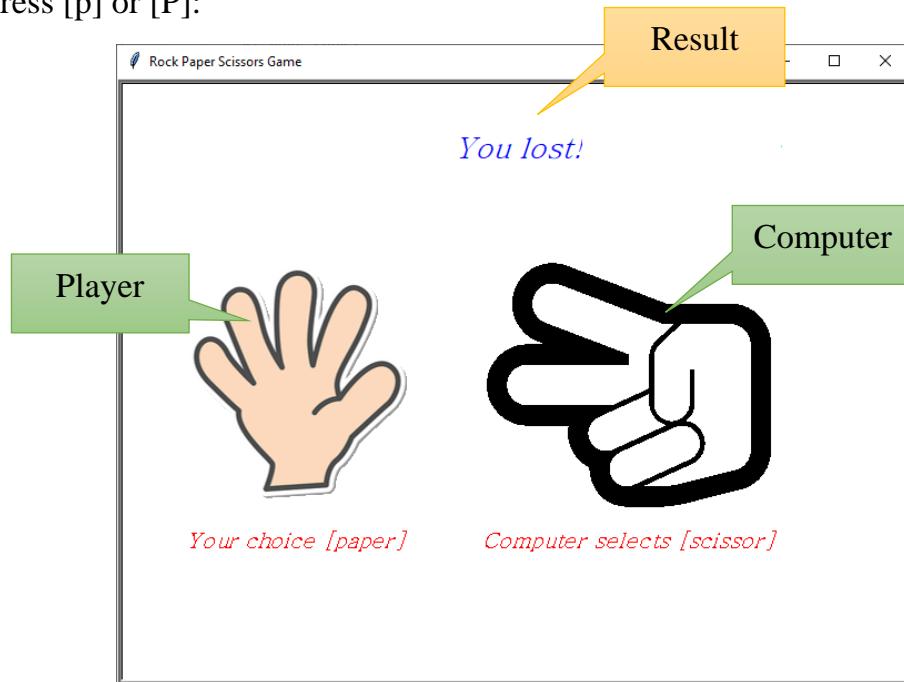
Player press [s] or [S]:



Player press [s] [S]. program will show Scissors for player. And now random computer is Paper. In this case, player is won!

Test case 4:

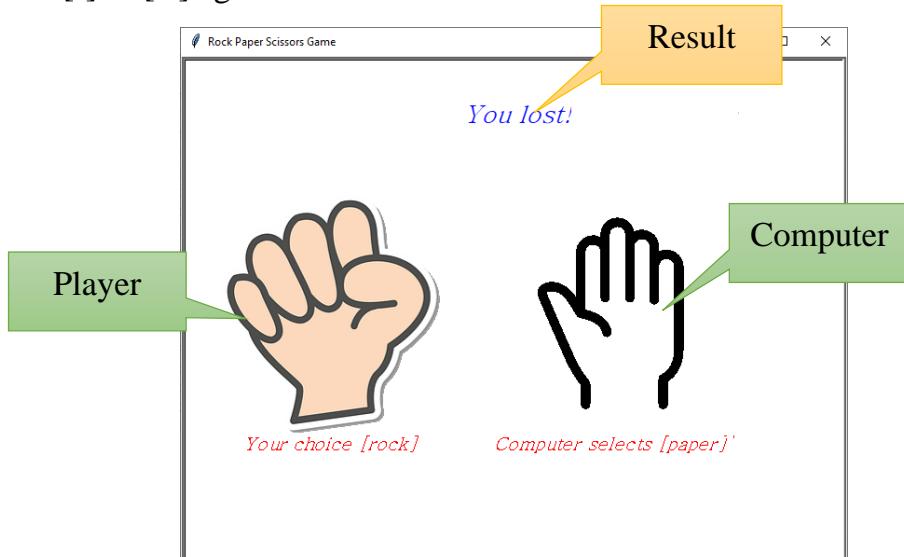
Player press [p] or [P]:



Player press [p] [P]. program will show Paper for player. And now random computer is Scissors. In this case, player is lost!

Test case 5:

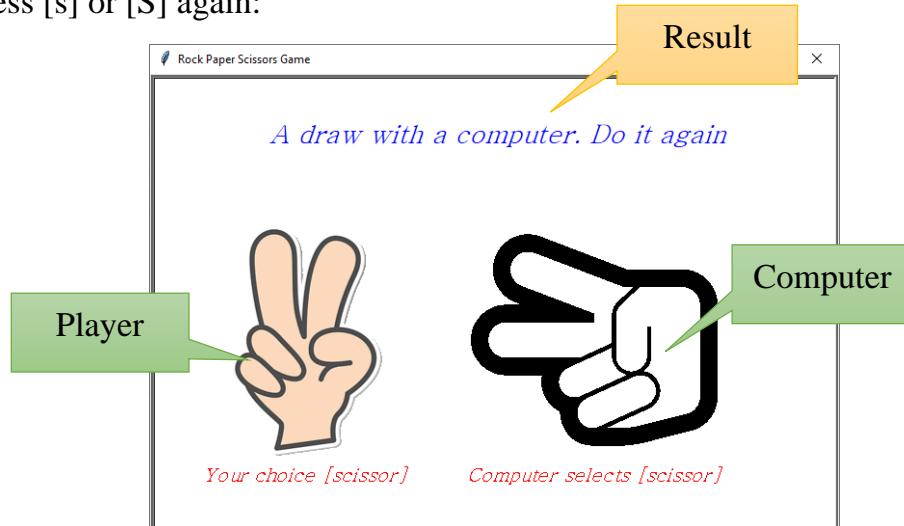
Player press [r] or [R] again:



Player press [r] [R]. program will show Rock for player. And now random computer is Paper. In this case, player is lost!

Test case 6:

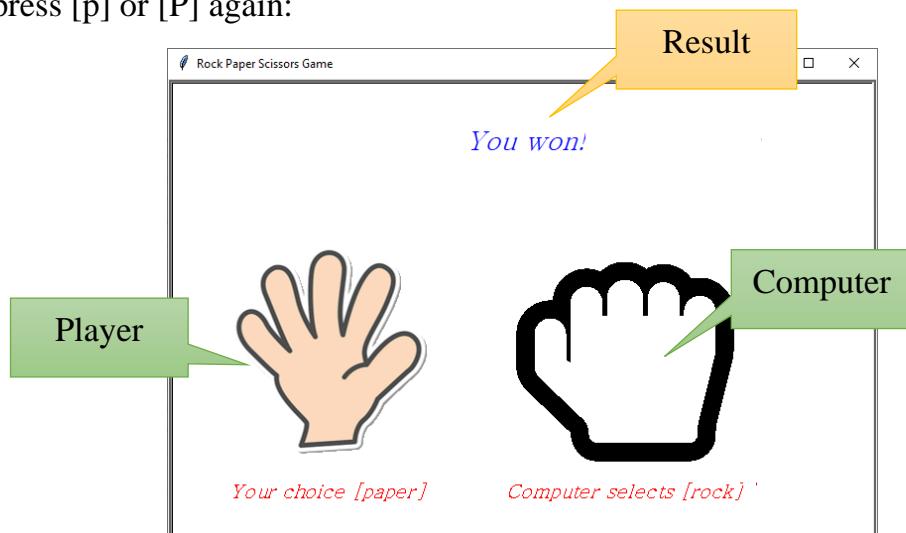
Player press [s] or [S] again:



Player press [s] [S]. program will show Scissors for player. And now random computer is Scissors. In this case, player and computer getting drawing!

Test case 7:

Player press [p] or [P] again:



Player press [p] [P]. program will show Paper for player. And now random computer is Rock. In this case, player is won!

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise25.py

30. Exercise 26 – Memory test game machine

30.1. Learning Objectives

30.1.1. Basic learning objectives

- You can use the sleep() function of the Time module appropriately.
- Textinput() function can be used appropriately
- You can generate random numbers in a specific range and store as many as you want in the list.

30.1.2. In-depth learning objectives

- You can use the write function of the Turtle module appropriately.

30.1.3. Problem situation

After receiving a random number n between 1 and 99 that we want to remember, we present n numbers at intervals of 2 seconds, close the window, and make a computer game in which the suggested numbers are matched in order. I want to use this game to test people's memory.

30.2. Problem analysis

30.2.1. The role of the computer

- How to generate random numbers between 1 and 99
- Think about how to control an action at two-second intervals
- Thinking about how to compare two lists for the same value

30.2.2. User

- Thinking about how to present the numbers
- After presenting a number, think about how to get user input

30.2.3. Input

- number of numbers to test
- The user's memory of numbers presented by the computer

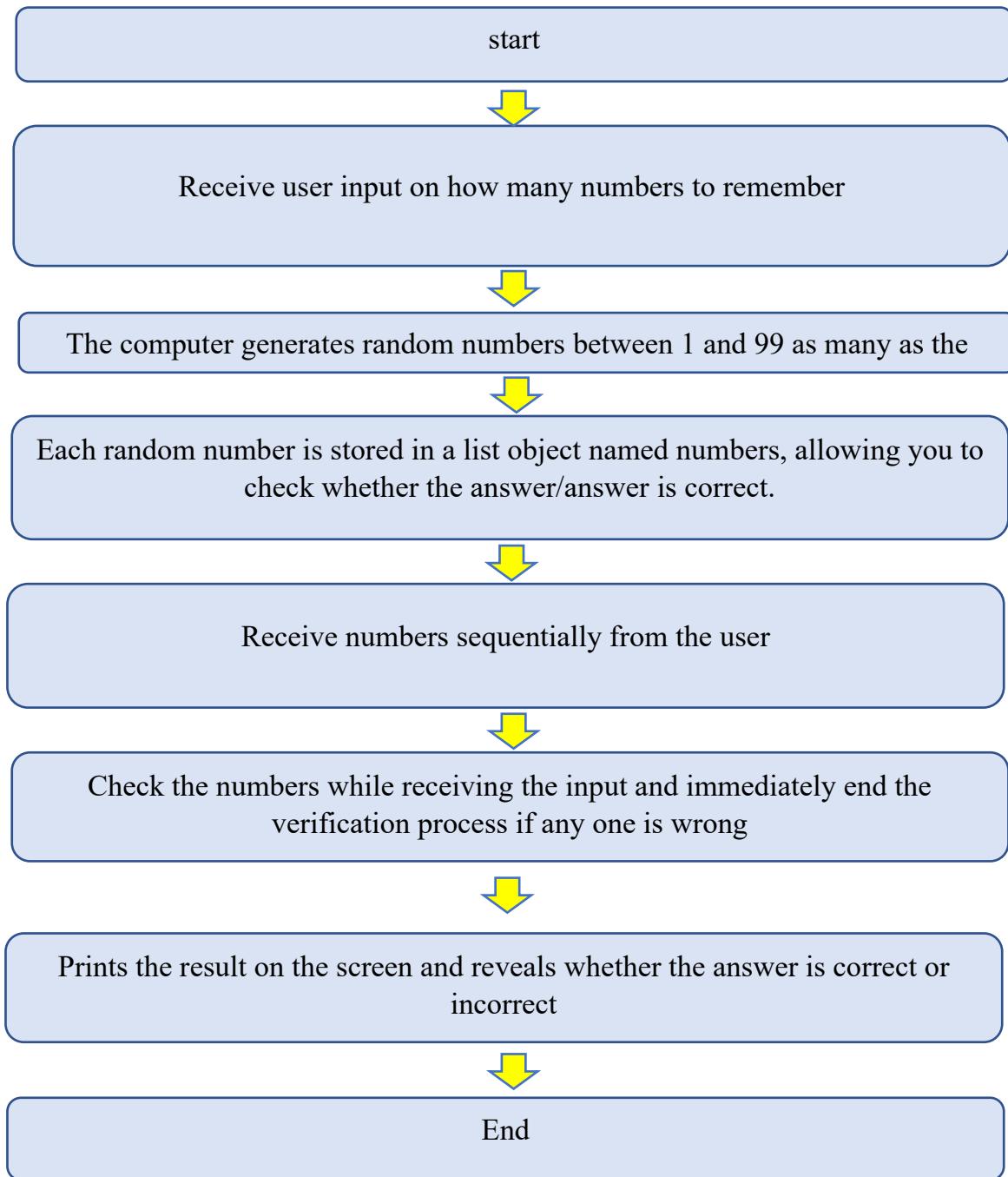
30.2.4. Output

- Numbers entered by the user and numbers presented by the computer
- Whether the answer is correct or incorrect

30.2.5. Data

- A list of turtle shapes

30.3. Full algorithm skeleton



30.4. Programming

30.4.1. Variables

- numbers: A list of computer-generated random numbers.
- qno: number of random numbers
- unumber: the number entered by the user

30.4.2. Some explanations

Methods of importing modules:

```
from turtle import * # Turtle module
from random import randint #Random number related module
from time import sleep #Time training module
```

Receive input with `textinput()` function and set the size of the window

```
qno=int(textinput("","How many numbers do you want to remember?"))

if qno>1 :
    setup(300,200)
```

- Repeat the following as many times as the number of random numbers to generate (qno).
 - Generate a random number between 1 and 99 and output it to (0, 0) of the turtle window
 - Save the generated random number to list numbers
 - Stop execution for a while, delete the turtle window, and initialize the turtle's position

```
for i in range(qno):
    # Clear the screen and set the turtle's position to (0,0)
    reset()
    #hide the shape of the turtle
    ht()
    # Pick up the pen to stop drawing
    pu()
    #turtle 위/자/의 이/동
    goto(-30,0)
    numbers.append(randint(1,99))
    write(numbers[i],font=(" ",32))
    # 2 second pause
    sleep(2)
```

User guessing the numbers:

```
Success=True
for i in range(qno):#Compare 5 numbers
    unumber=int(input(str(i+1)+' digit>>'))
    if unumber != numbers[i] :
        #If not, treat incorrect answer
        print(numbers,"It's wrong answer")
```

```
#The answer is not the same, so it is treated as a failure
Success = False
break
if i == qno - 1: #If everything is correct, the answer is correct
    print(numbers,"That's the right answer.")
```

30.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

from turtle import * # Turtle module
from random import randint #Random number related module
from time import sleep #Time training module

numbers=[] #List to store random numbers

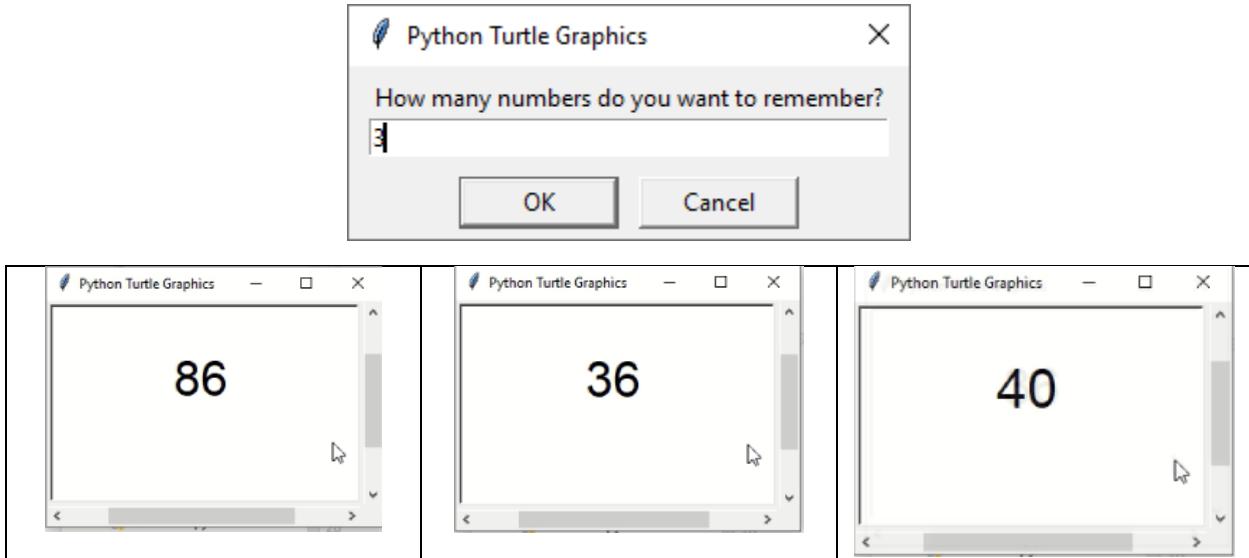
qno=int(textinput("","How many numbers do you want to remember?"))

if qno>1 :
    setup(300,200)

for i in range(qno):
    # Clear the screen and set the turtle's position to (0,0)
    reset()
    #hide the shape of the turtle
    ht()
    # Pick up the pen to stop drawing
    pu()
    #turtle უ/ჩ/ი/ ი/დ/ო
    goto(-30,0)
    numbers.append(randint(1,99))
    write(numbers[i],font=("",32))
    # 2 second pause
    sleep(2)
    #Close the turtle screen
    bye()
    Success=True
    for i in range(qno):#Compare 5 numbers
        unumber=int(input(str(i+1)+' digit>>'))
        if unumber != numbers[i] :
            #If not, treat incorrect answer
            print(numbers,"It's wrong answer")
            #The answer is not the same, so it is treated as a failure
            Success = False
            break
    if i == qno - 1: #If everything is correct, the answer is correct
        print(numbers,"That's the right answer.")
```

30.5. Testing and debugging

Run program and see the results:



Result:

```
C:\Python\Python38\python.exe D:/TranDuyThanh/
1 digit>>86
2 digit>>36
3 digit>>40
[86, 36, 40] That's the right answer.
```

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise26.py

31. Exercise 27 - Memory test game machine (improve)

31.1. Problem situation

Let's modify the program Exercise 26 to memorize 5 out of 52 random alphabets (lowercase and uppercase) and match them.

31.2. Programming-the whole program

These codes I improved from Exercise 26:

Improved the Memory test game machine

1. Study string join method to convert list to string with comma-separated and get ascii lower and upper case list
2. study functools library to listening event and get the value from keyboard
3. Design GUI for Memory Game
Player can play on the GUI, click button to start
And can re-play the game no need to re-run application
4. Allow user choose any number alphabet to test (recommendation 5)

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 26.
#1. Study string join method to convert list to string with comma-separated
# and get ascii lower and upper case list
#2. study functools to listening event and get the value from keyboard
#3. Design GUI for Memory Game
#4. allow user choose any number alphabet to test (recommendation 5)

import string
from turtle import * # Turtle module
from random import randint # Random number related module
from time import sleep # Time training module
import functools
import turtle as turtle

#get ascii lower case list
alphabets_lowercase=list(string.ascii_lowercase)
#get ascii upper case list
alphabets_uppercase=list(string.ascii_uppercase)
#convert to 52 alphabets (Upper + lower)
alphabets=alphabets_lowercase+alphabets_uppercase
#save alphabets that player has to memorize
alphabets_memory=[]
#current position alphabet that player has to answer
position_memory=-1
```

```
#needToRemoveListen remove listening when finish the GAME
needToRemoveListen=False
#cursor, font size
CURSOR_SIZE = 20
FONT_SIZE = 12
FONT = ('Arial', FONT_SIZE, 'bold')

#random_turtle show the random Alphabet that user has to memorize
random_turtle=Turtle()
#turtle_title draw: Welcome title
turtle_title=Turtle()
#alphabets_uppercase_title: draw list of upper class
alphabets_uppercase_title=Turtle()
#alphabets_lowercase_title: draw list of lower class
alphabets_lowercase_title=Turtle()

#this function use to write a string with many parameter
def drawLabel(turtleItem, msg, x, y, color, style, align):
    turtleItem.reset()
    turtleItem.penup()
    turtleItem.setposition(x, y)
    turtleItem.color(color)
    turtleItem.write(msg, move=False, font=style, align=align)
    turtleItem.hideturtle()

#this function process to Start Game
#When user click on the Red Circle
#alphabets will be stored in the alphabets_memory
#and show on the random_turtle
def draw_onclick(x, y):
    qno = int(textinput("", "How many numbers do you want to remember?"))
    style = ('tahoma', 50, 'bold')

    alphabets_memory.clear()
    for i in range(qno):
        position = randint(0, len(alphabets) - 1)
        alphabet = alphabets[position]
        alphabets_memory.append(alphabet)
        drawLabel(random_turtle, alphabet, 0, 10, 'blue', style, "left")
        sleep(2)
    style = ('tahoma', 20, 'bold')
    drawLabel(random_turtle, "Please press key to answer!",
              -150, 10, 'red', style, "left")

    #listening event
    for k in alphabets_lowercase:
        turtle.onkeypress(functools.partial(event_handler, k), key=k)
    for k in alphabets_uppercase:
        turtle.onkeypress(functools.partial(event_handler, k), key=k)
    global needToRemoveListen, position_memory
    needToRemoveListen=False
    position_memory=0
    turtle.listen()

#this function use to draw welcome and alphabet
def drawWelcomeAndAlphabet():
```

```

#draw welcome
style = ('tahoma', 20, 'bold')
drawLabel(turtle_title, "Welcome to Memory Testing!",
-150, 250, 'blue', style, "left")
style = ('tahoma', 12, 'italic')
#draw upper case alphabets
s_uppercase=""
for s in alphabets_uppercase:
    s_uppercase=s_uppercase+s+" "
s_uppercase="Uppercase alphabet:"+s_uppercase
drawLabel(alphabets_uppercase_title, s_uppercase,
-350, 200, 'blue', style, "left")
#draw lower case alphabets
s_lowercase=""
for s in alphabets_lowercase:
    s_lowercase=s_lowercase+s+" "
s_lowercase="Lowercase alphabet:"+s_lowercase
style = ('tahoma', 12, 'italic')
drawLabel(alphabets_lowercase_title, s_lowercase, -350, 150, 'blue', style, "left")
#this function use to draw red Circle button
#player will click on this Button to Start the Game
def drawRedCircleButton():
    button = Turtle()
    button.hideturtle()
    button.shape('circle')
    button.fillcolor('red')
    button.penup()
    button.goto(0, 100)
    button.write("Click red Circle to Start Game!", align='center', font=FONT)
    button.sety(100 + CURSOR_SIZE + FONT_SIZE)
    button.onclick(draw_onclick)
    button.showturtle()
#this function use to process listening player press the Key on Keyboard
#if all Alphabets is right position->show congratulations
#if any an alphabet is wrong->show Condolatory!
def event_handler(key):
    global needToRemoveListen,position_memory
    if needToRemoveListen == True:
        return
    print(key)
    style = ('tahoma', 20, 'bold')
    drawLabel(random_turtle, key, -150, 10, 'blue', style, "left")
    sleep(0.5)
    if key != alphabets_memory[position_memory]:
        needToRemoveListen=True
        style = ('tahoma', 20, 'bold')
        drawLabel(random_turtle, "Condolatory!\nYou are wrong, the memory list:" +
        ','.join(alphabets_memory), -150, 10,'red', style, "left")
        position_memory = position_memory+1;
    if position_memory == len(alphabets_memory) and needToRemoveListen ==False:
        style = ('tahoma', 20, 'bold')
        drawLabel(random_turtle, "Congratulations!\nYou answered correctly:"+
        ','.join(alphabets_memory),-150, 10, 'blue', style, "left")
    turtle.setup(900, 600)
    turtle.title("Memory Testing!")

```

```
drawWelcomeAndAlphabet()  
drawRedCircleButton()  
turtle.mainloop()
```

31.3. Testing and debugging

Here are some testing cases for the programming:

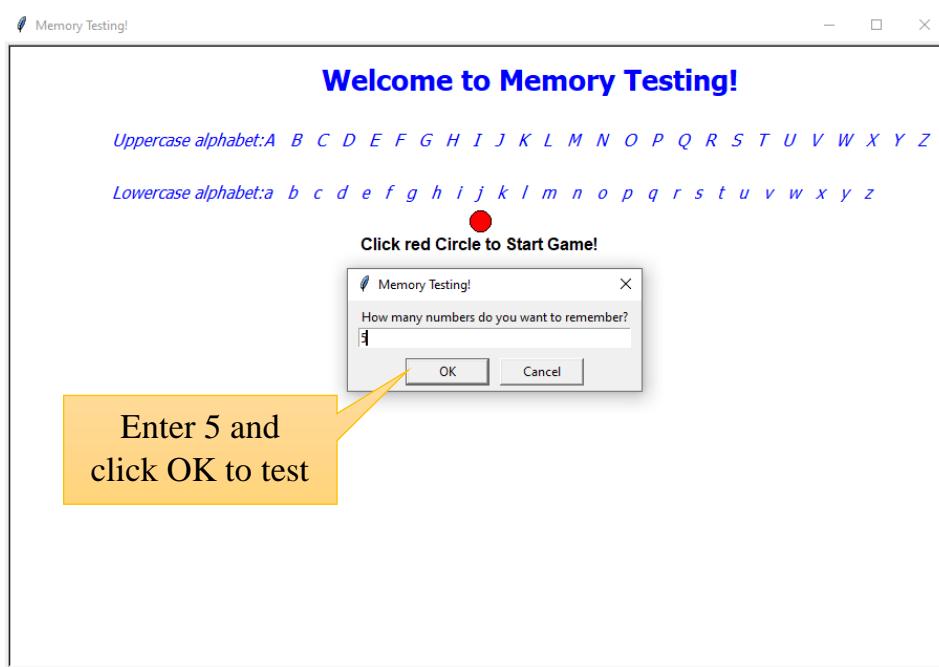
Test case 1:

Run Program, we have the Screen is shown below:

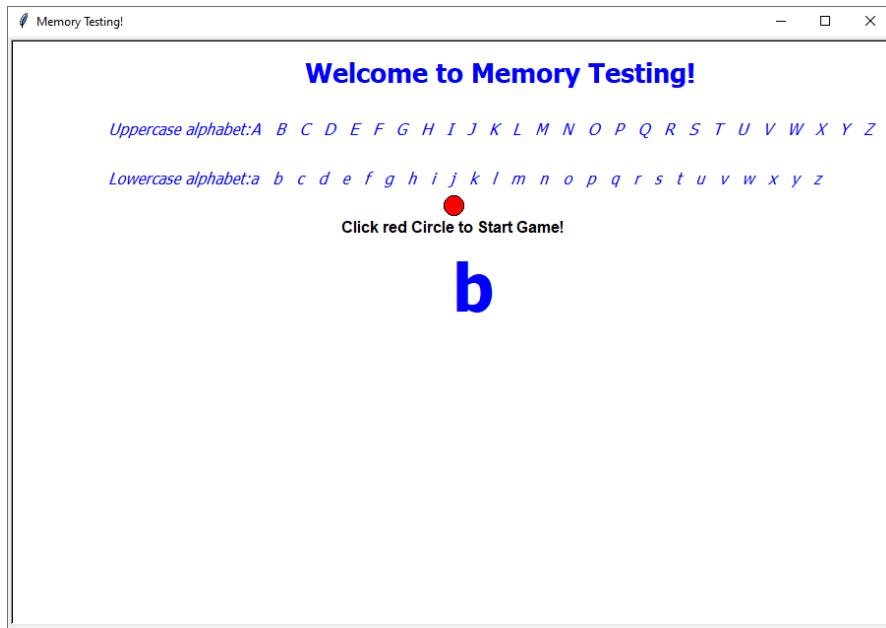


Test case 2:

Now, we click the Red Circle button to start the Game:



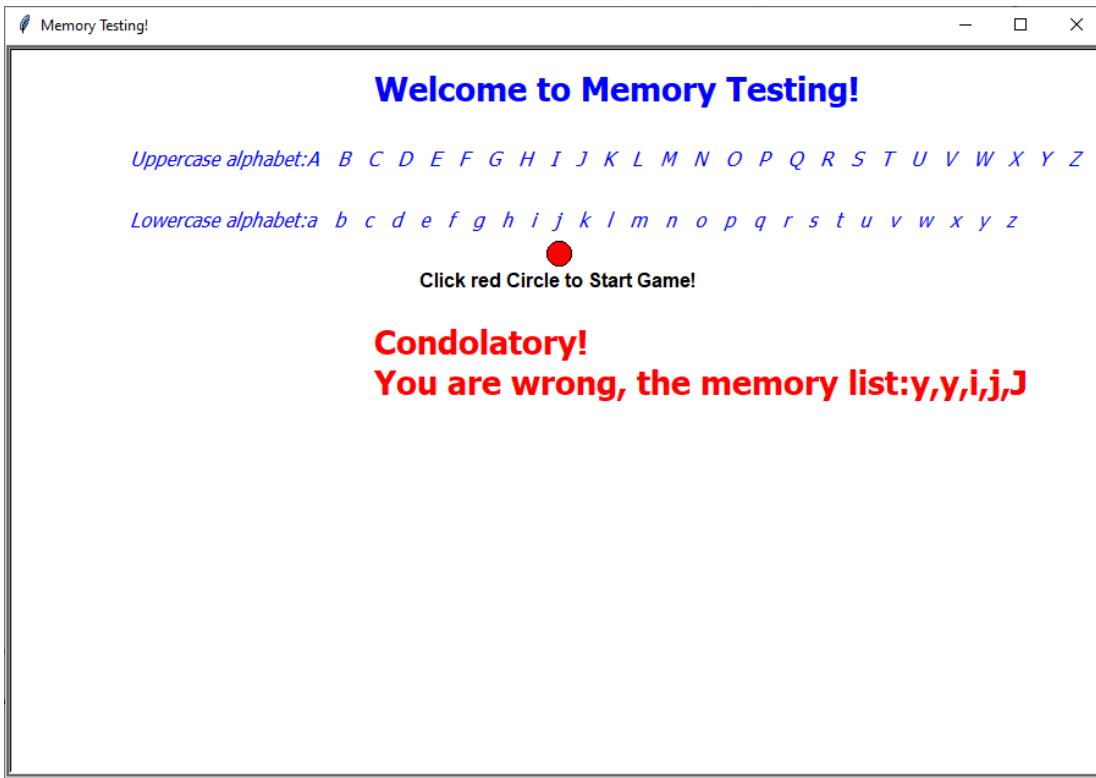
Program will draw step by step with sleep 2 second for each random Alphabet



After draw all step-by-step random alphabet. Program will ask user to play:



Now, We press to start the game. If player press not correct the alphabet, program will show message "...You are wrong..." with the list random of computer:

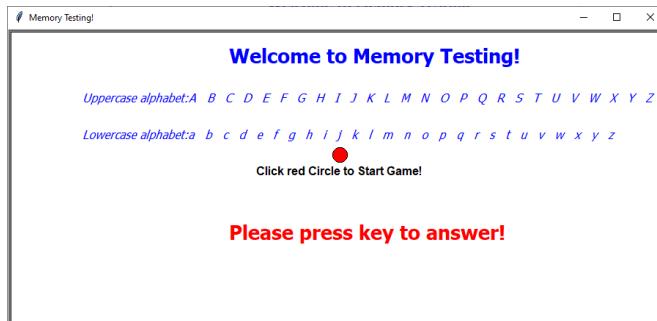


Test case 3:

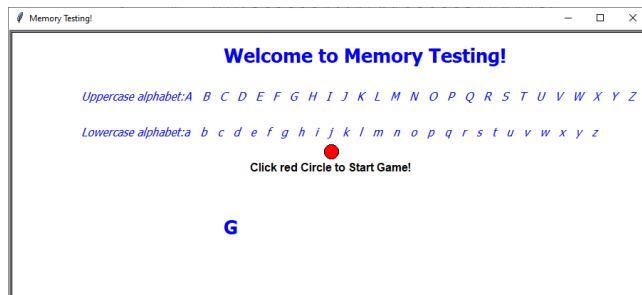
Now, we click the Red Circle button again to start the Game:



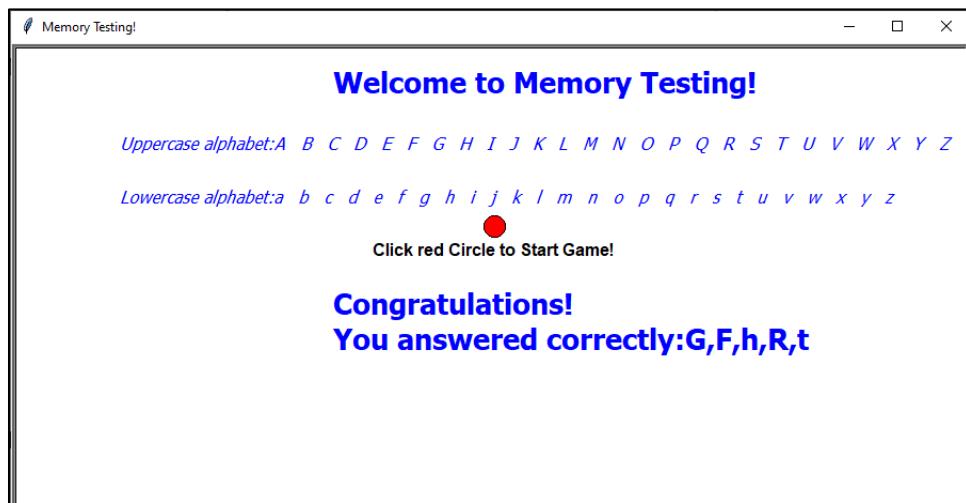
We enter 5 and click ok to answer. In this test case, 5 alphabets are: G,F,h,R,t



Now, We press G, F, h, R, t



Final result, if we press all alphabets are correct:



We can test many cases. By re-click Red Circle button!

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise27.py

32. Exercise 28 – GUI calculator

32.1. Learning Objectives

32.1.1. Basic learning objectives

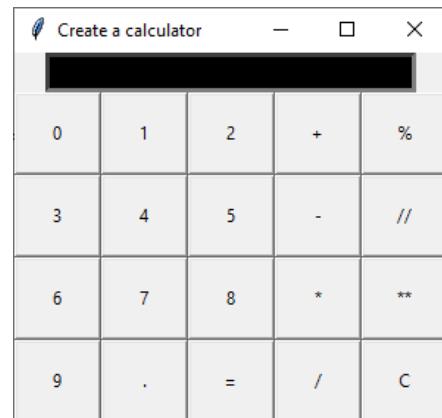
- You can make up a name for Window
- You can create a Button widget object in Tkinter
- Tkinter's Entry Agenda can be created
- You can define functions to handle events on mouse clicks or keystroke events..

32.1.2. In-depth learning objectives

- You can create multiple Button widget objects..

32.1.3. Problem situation

As shown in the picture, let's program to print the calculation result by clicking '=' after clicking the button with the mouse in the black box at the top of the window using the tkinter module. . In addition to the additional arithmetic operators, this calculator includes operators to calculate remainders, operators to calculate exponents, and operators to calculate integer division and real division.



32.2. Problem analysis

32.2.1. The role of the computer

- Let's think about the things a computer has to do when we build a calculator.
 - How to use Tkinter's entry agenda and button widget
 - How to place a widget on a window
 - How to call a handler that handles the button click event when the button click event occurs
 - Definition of event handlers

32.2.2. User

- Size and position the button to make it easier for users to calculate
- Define a function so that once the calculation is complete, the previous inputs can be easily erased.

32.2.3. Input

- Button click

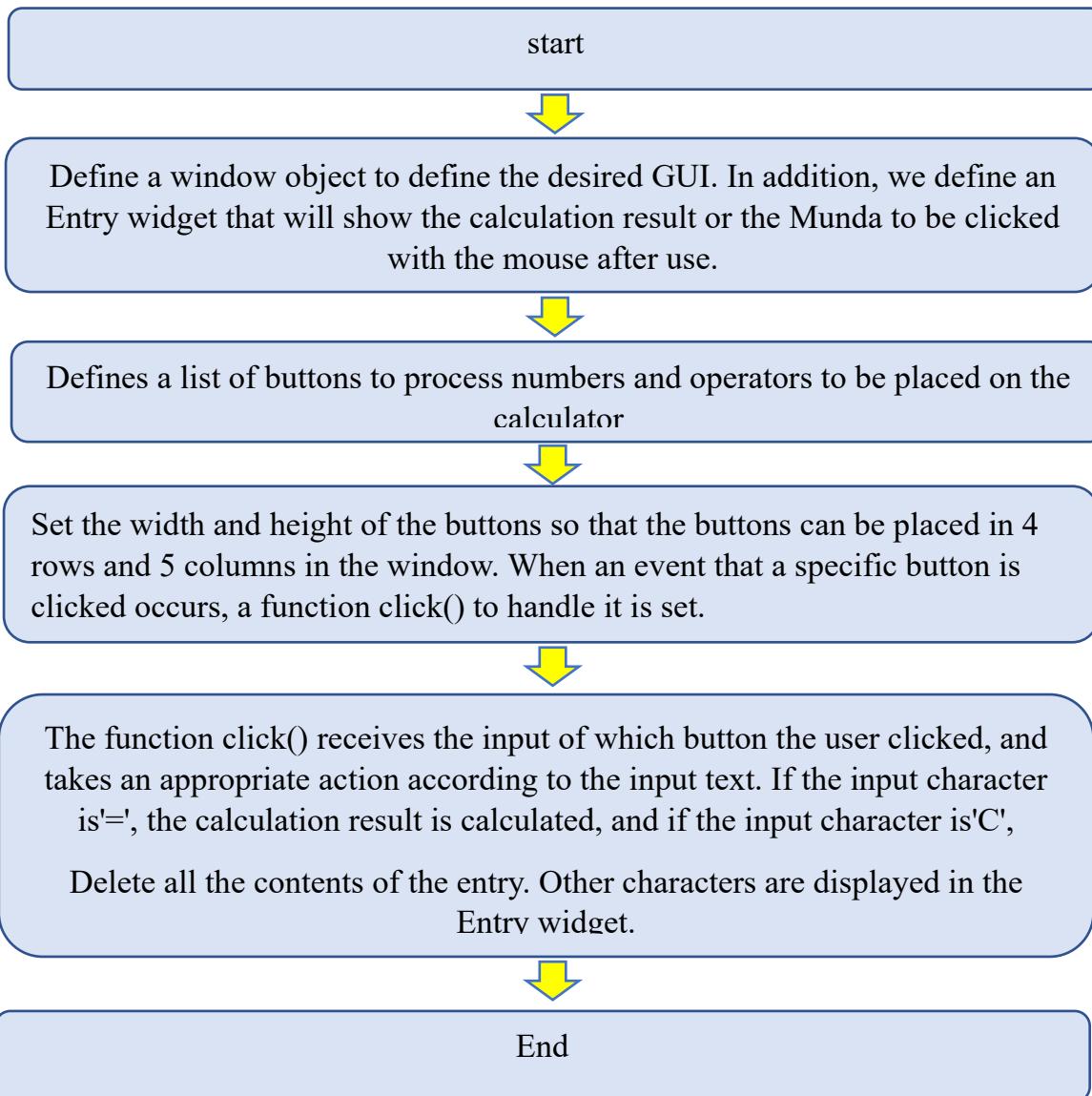
32.2.4. Output

- Text written above the button clicked
- calculation result

32.2.5. Data

- Numbers 0 to 9, four arithmetic operators (+, -, *, /) and other Python operators (%, //, **)

32.3. Full algorithm skeleton



32.4. Programming

32.4.1. Variables

- window: the window object
- e: Entry widget. Show calculation results
- buttons: List of characters to be printed on the button
- row: The row where the specific button will be placed
- col: the column in which the specific button will be placed
- key: The character above the button the user clicked
- b: One Button widget

32.4.2. Some explanations

The tkinter module and defines a window object:

```
from tkinter import *
window = Tk()

window.title("Create a calculator")
```

Define an Entry widget to show the result of the calculation:

```
e=Entry(window,width=40,bg="black",fg="white",bd=5)
e.grid(row=0,column=0,columnspan=5)
```

Grid:

When placing the Entry widget in a window, it will be placed at the bottom.

Button widgets are arranged in a grid shape, 5 per row. Assuming you want to place the columnspan to 5 setting. Since it is the first widget, the row and column values are All set to 0.

Initialization of the button list:

```
buttons=[  
    '0', '1', '2', '+', '%',  
    '3', '4', '5', '-', '//',  
    '6', '7', '8', '*', "***",  
    '9', '.', '=', '/', 'C']
```

Button widget initialization:

```
row = 1
col = 0
for char in buttons:
    b=Button(window, text=char, width=7, height=3, command=click)
    b.grid(row=row, column=col)
    col += 1
    if col >4:
        row +=1
        col=0
```

- The Button widget is defined by dividing the window into a 4-row, 5-column grid.
- Buttons are arranged from row 1. It starts to be arranged from the 0th column, and when the value of the column col variable becomes 5, This means that one row is full of 5 buttons, so we set the variable row to 1 to go to the next row. increment, and the variable col is initialized to 0.
- When an event that a button is clicked occurs, the function that handles it, click, is assigned as the value of the command option.

When an event of clicking a specific button occurs, click(), a function that handles this event:

```
for char in buttons:
    def click(key = char):
        if key == '=':
            result=eval(e.get())
            s=str(result)
            e.delete(0,END)
            e.insert(0,s)
        elif key =='C':
            e.delete(0,END)
        else:
            e.insert(END,key)
```

- Function click() is defined in a loop to accept which button was clicked as a parameter.
- Determine which button was clicked and what the corresponding character is with the parameter key.
- If the key value is ‘ = ’, define the strings entered so far as parameters of the eval() function and calculate the result value.
- If the key value is ‘C’, the contents of the Entry widget are deleted with the delete() function.
- Other characters are stored in the Entry widget by using the method **insert ()** of the object variable e in turn.

32.4.3.Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
from tkinter import *
window=Tk()

window.title("Create a calculator")
e=Entry(window,width=40,bg="black",fg="white",bd=5)
```

```

e.grid(row=0,column=0,columnspan=5)
buttons=[

'0', '1', '2', '+', '%',
'3', '4', '5', '-', '/',
'6', '7', '8', '*', "**",
'9', '.', '=', '/', 'C']

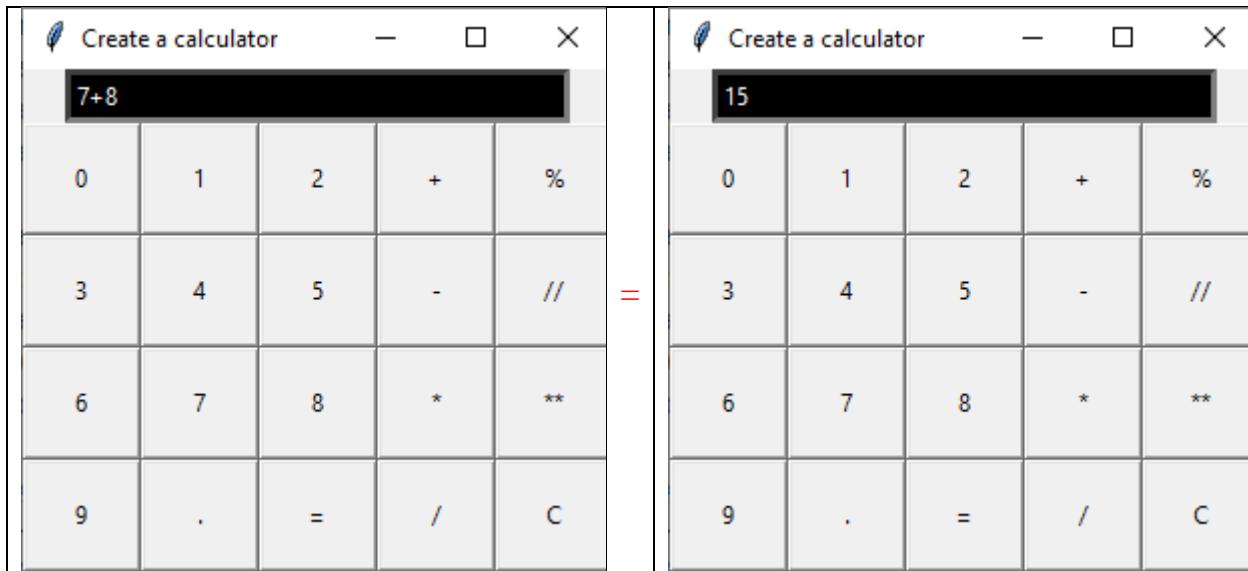
row = 1
col = 0
for char in buttons:
    def click(key = char):
        if key == '=':
            result=eval(e.get())
            s=str(result)
            e.delete(0,END)
            e.insert(0,s)
        elif key =='C':
            e.delete(0,END)
        else:
            e.insert(END,key)
    b=Button(window, text=char, width=7, height=3, command=click)
    b.grid(row=row,column=col)
    col +=1
if col >4:
    row +=1
    col=0
window.mainloop()

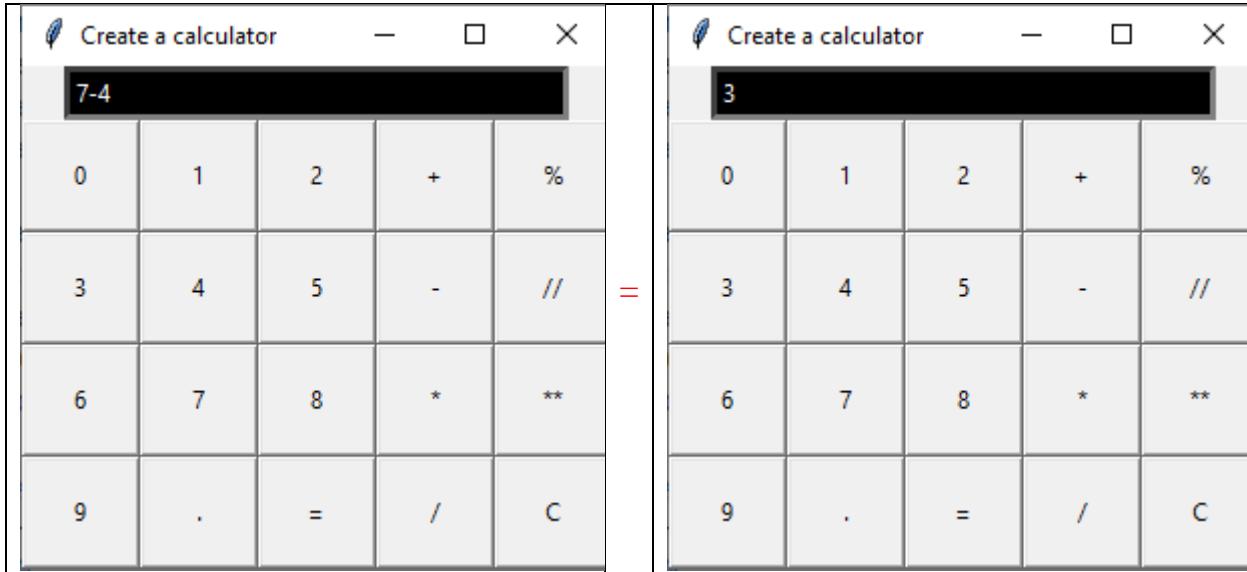
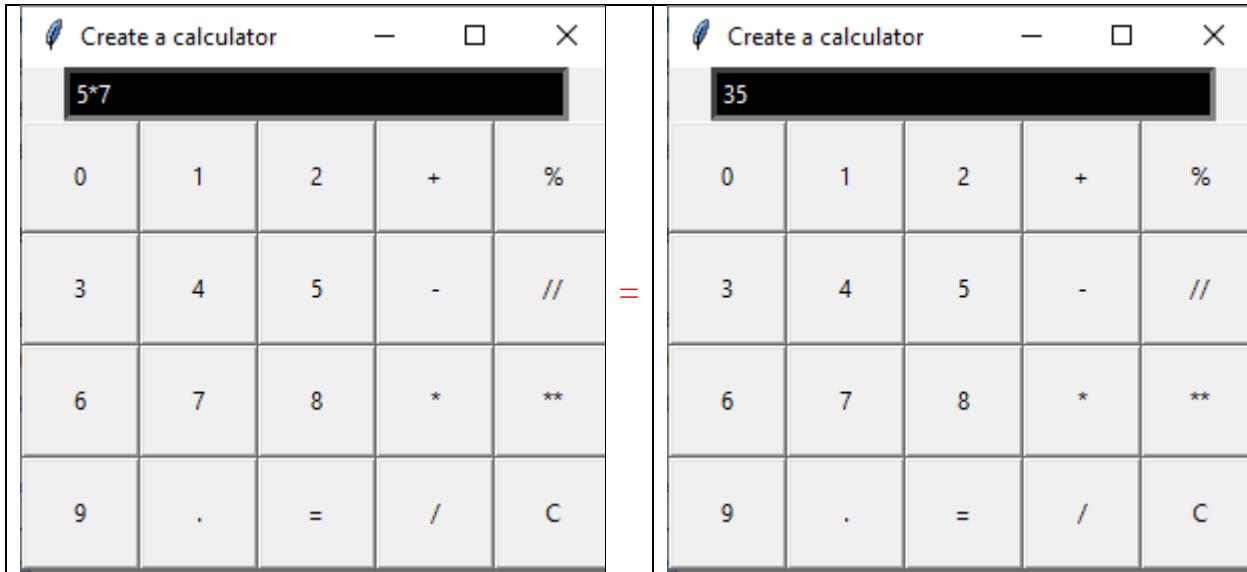
```

32.5. Testing and debugging

Run program and see the results:

Test case 1: +



Test case 2: -**Test case 3: ***

You should test complex expression.

I put my code in github, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise28.py

33. Exercise 29 - GUI calculator (improve)

33.1. Problem situation

Modify exercise 28, how the program will change if we switch to the Label widget instead of the Entry widget?

33.2. Programming-the whole program

I modified code, I used Label instead Entry, add new labelConfig and center functions for **calculator GUI improved**

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 28
#Use label instead Entry
#create a new function labelConfig(string,append=False)
# string: text to display
# append false if user wants to clear old data and display new text
# true if user wants to append the new text
#create a new function center(window) to set center screen of the desktop
from tkinter import *
window = Tk()
window.title("Create a calculator")
lformula=Label(window, text="", anchor='w', width=40, bg="black", fg="white", bd=5)
lformula.grid(row=0, column=0, columnspan=5)
buttons=[
    '0', '1', '2', '+', '%',
    '3', '4', '5', '-', '/',
    '6', '7', '8', '*', "**",
    '9', '.', '=', '/', 'C']
row = 1
col = 0
#this function use to update text for label
#string: text to display
#append false if user wants to clear old data and display new text
# true if user wants to append the new text
def labelConfig(string,append=False):
    if append:
        text = lformula.cget("text") + string
        lformula.configure(text=text)
    else:
        lformula.configure(text=string)
for char in buttons:
    def click(key = char):
        if key == '=':
            result=eval(lformula.cget("text"))
            s=str(result)
```

```

labelConfig("=", True)
labelConfig(s, True)
elif key =='C':
    #reset text for Label
    labelConfig("", False)
else:
    #concat string for label
    labelConfig(key, True)
b=Button(window, text=char, width=7, height=3, command=click)
b.grid(row=row,column=col)
col +=1
if col >4:
    row +=1
    col=0
#this function use to set the screen is center of the desktop
def center(window):
    window.update_idletasks()
    width = window.winfo_width()
    frm_width = window.winfo_rootx() - window.winfo_x()
    win_width = width + 2 * frm_width
    height = window.winfo_height()
    titlebar_height = window.winfo_rooty() - window.winfo_y()
    win_height = height + titlebar_height + frm_width
    x = window.winfo_screenwidth() // 2 - win_width // 2
    y = window.winfo_screenheight() // 2 - win_height // 2
    window.geometry('{0}x{1}+{2}+{3}'.format(width, height, x, y))
    window.deiconify()
center(window)
window.mainloop()

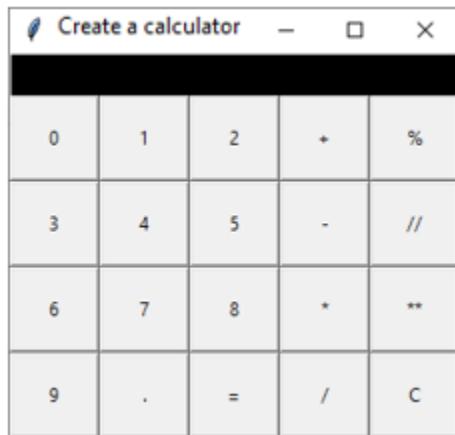
```

33.3. Testing and debugging

Here are some testing cases for the programming:

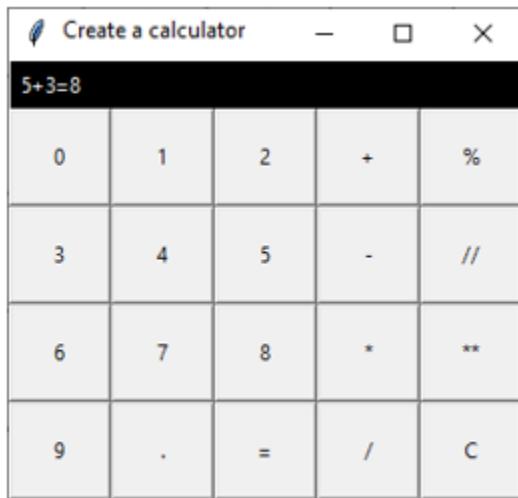
Test case 1:

Run the software, it is placed at center of the desktop screen:



Test case 2:

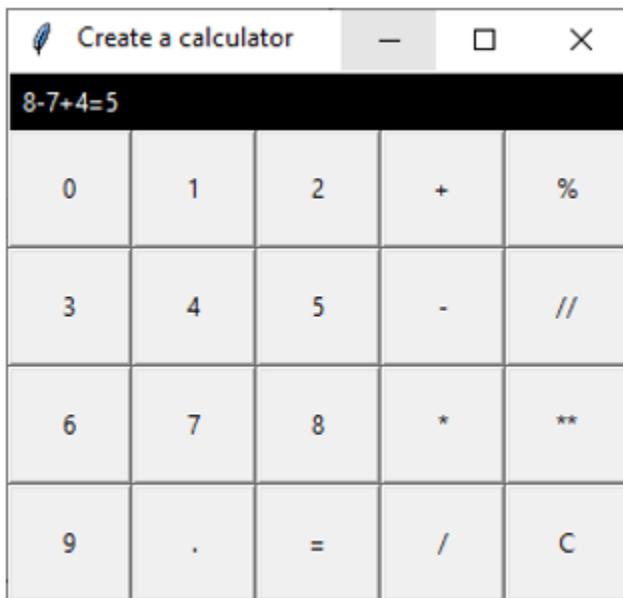
Test for '+':



In this case, we click 5, +, 3 and =. The result is 8 as shown above.

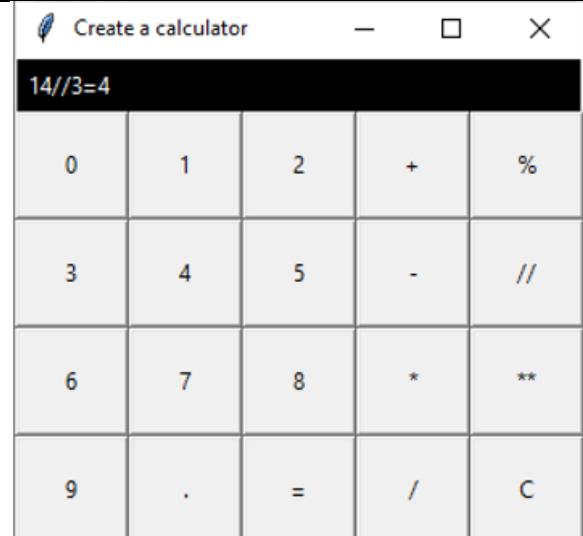
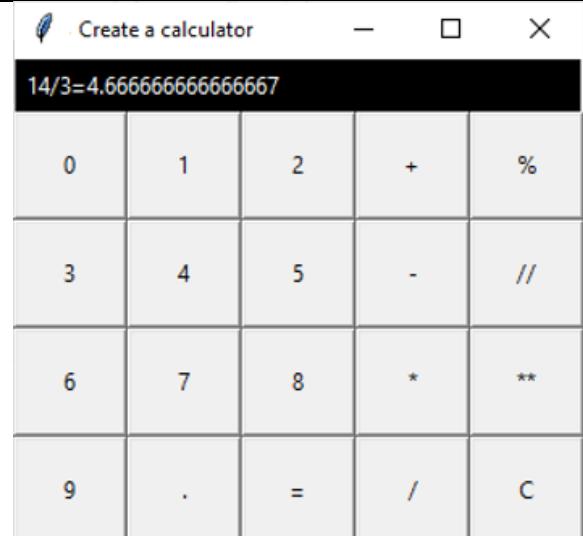
Test case 3:

Click 'C' and test for '-' and '+':



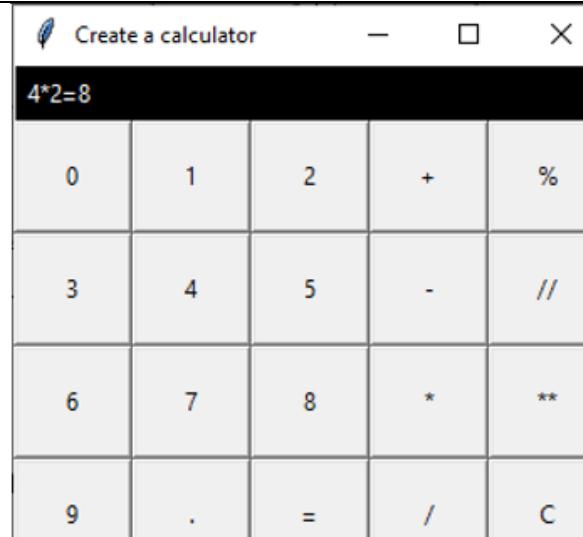
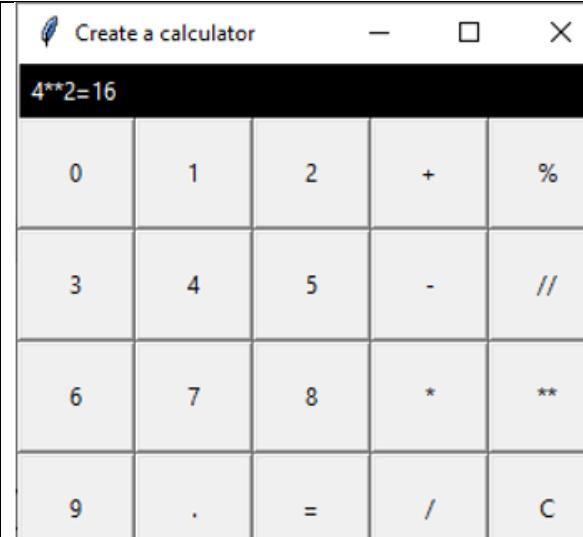
Test case 4:

Click 'C' and test for '// and '/':

	
The result is round	The result is not round

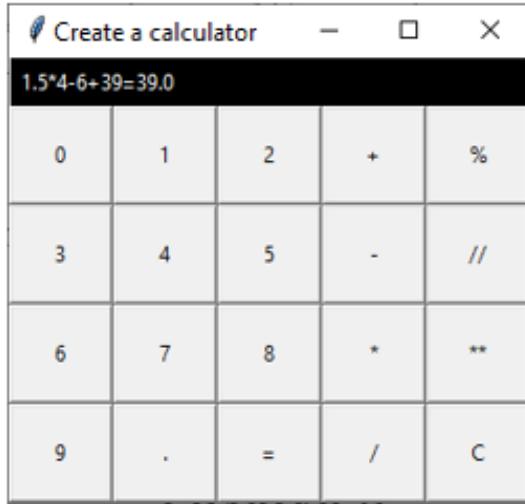
Test case 5:

Click ‘C’ and test for ‘*’ and ‘**’:

	
The result is 8, $4*2=8$	The result is 16, $4^2=16$

Test case 6:

Click ‘C’ and test for decimal number and complex expression



We can test with another expressions.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise29.py

34. Exercise 30 – Quiz Game

34.1. Learning Objectives

34.1.1. Basic learning objectives

- A different string can be presented continuously using the config() function of the Label widget.
- Global variables can be used to use variables defined outside the function inside the function.
- You can process images using the PhotoImage widget.

34.1.2. In-depth learning objectives

- You can use the RadioButton widget

34.1.3. Problem situation

She is studying cultural areas covered in social studies. In order to study with friends, it is intended to be programming so that a computer can produce a quiz problem of guessing cultural areas. In other words, we try to make a quiz program that answers the questions correctly when the contents of the question and the image of the cultural area are presented. At this time, the content of the problem is stored in the file, and the number is uncertain because the content of the file can be updated. The subject or content of the problem can be changed in the future depending on which problem is contained in the file.

34.2. Problem analysis

34.2.1. The role of the computer

- Presenting images and contents related to images using GUI
- File processing
- Using Label, Entry, and Button widgets
- Keystroke event handling

34.2.2. User

- Designing screens such as images, questions, fields for inputting correct answers, buttons, etc.
- Thinking about how to handle the correct answer
- After solving all the problems stored in the file, think about how to deal with the screen...

34.2.3. Input

- [Enter] key to confirm the correct answer and correct answer

- Click the button to view the next issue

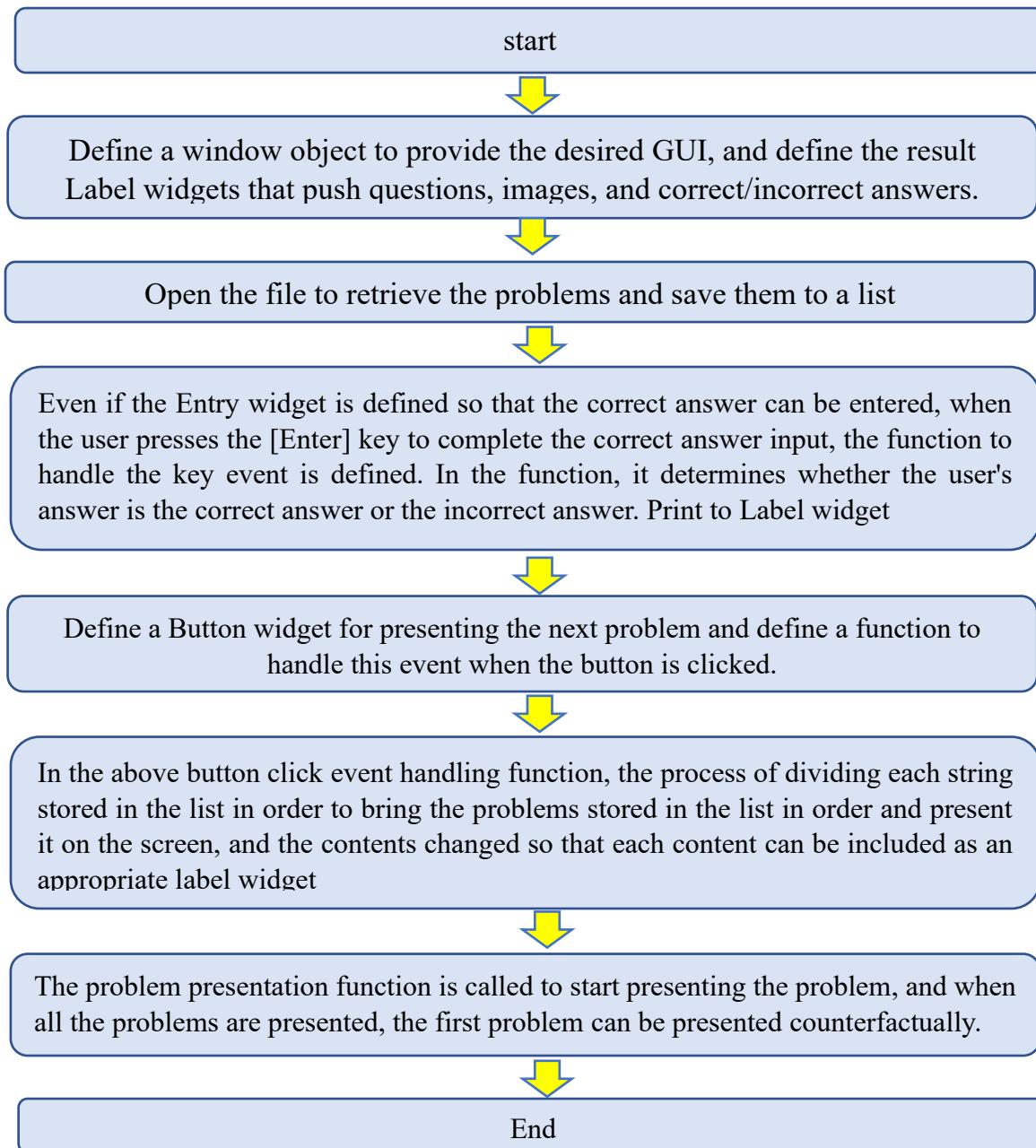
34.2.4. Output

- Problem and problem image, next problem check button
- Whether the answer is correct/wrong

34.2.5. Data

- The file storing the problem

34.3. Full algorithm skeleton



34.4. Programming

34.4.1. Variables

- window: the window object
- qlabel: problem label widget variable
- ilabel: Image label widget variable
- rlabel: Result label widget variable indicating whether the correct answer is correct or not
- file: file object
- p: a list containing the string containing the problem as an element
- answer: the correct answer string
- e: Entry widget variable that handles user input

34.4.2. Some explanations

Includes tkinter for the GUI and defines the Label widget variables to define the quiz question, image, and answer fields.

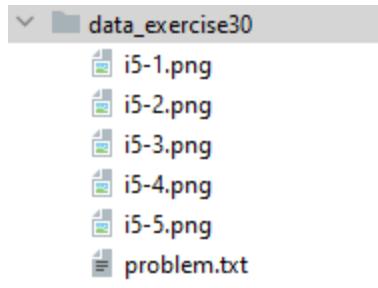
```
from tkinter import *
window=Tk()
window.title("Take the quiz")
window.geometry("400x650+10+10")
#question label
qlabel=Label(window,width=100,text="")
qlabel.pack()
#image label
ilabel=Label(window)
ilabel.pack()
#Write the correct answer and press [Enter].
Label(window,text="Write the correct answer and press [Enter].").pack()
#Correct/Incorrect result label
rlabel=Label(window)
rlabel.pack()
```

Open file containing quiz questions and answers:

Data for exercise 30.

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise30

```
file=open("data_exercise30/problem.txt","r",encoding="utf8")
p=file.readlines()
file.close()
i=-1
answer=""
```



- Open the text file 'problem.txt' containing the problem.
- By using the file object's `readlines()` function, each line of the file is read and stored as an element of the list.
- Initialize the index variable to be used in the program and the variable answer indicating the correct answer.
- In the problem.txt file, four lines of strings in the format of
“Problem Content: Problem Answer: Image File Name” are stored.

Contents of the **problem.txt** file:

```
Which cultural area corresponds to the following picture?:Mongolian steppe:data_exercise30/i5-1.png
Which cultural area corresponds to the following picture?:Japanese geisha:data_exercise30/i5-2.png
Which cultural area corresponds to the following picture?:Traditional Korean clothes:data_exercise30/i5-3.png
Which cultural area corresponds to the following picture?:Da Nang Golden Bridge:data_exercise30/i5-4.png
Which cultural area corresponds to the following picture?:Great Wall:data_exercise30/i5-5.png
```

❖ Create a button to solve the next problem

- When the button is clicked, the function `getQuestion()` to be executed is specified.
- Called once to perform the first problem without clicking a button.

```
Button(window,text="next problem",command=getQuestion).pack()
getQuestion()
window.mainloop()
```

❖ GetQuestion() function to print the problem

- Variables `i`, `answer`, and `e` are treated as global variables so that they can be used throughout the program.
- Since only 4 questions are stored in the `problem.txt` input file, the number of problems is limited to 4.
- When the next question button is clicked, the `getQuestion()` function is called, so the value of the index variable `i` indicating the question order is increased by 1.

- If all the questions are answered and i is 4, i is reinitialized to 0.
- Deletes all the strings stored in the entry object variable e containing the previous problem, and also deletes all result labels.
- Using the label's config() method.

```
global i,answer,e
i +=1
# Repeat because there are 4 problems
if i >=4 : i =0
e.delete(0,len(e.get()))
rlabel.config(text="")
```

- The string element (p[i]) containing the current problem is taken from the list of all problems p, the newline character ('\n') at the end of the string is first removed, and then replaced with the aQuestion variable.
- The aQuestion string variable is split into a list of strings called Q using the split() function. ':' is used as a separator to separate information.
- Q contains the problem in the first list element, the answer in the second list element, and the image filename for the problem in the third list element.

```
aQuestion =p[i].strip()
Q=aQuestion.split(":")
```

Q[0]

Q[1]

Q[2]

Which cultural area corresponds to the following picture?:Mongolian steppe:data_exercise30/i5-1.png
 Which cultural area corresponds to the following picture?:Japanese geisha:data_exercise30/i5-2.png
 Which cultural area corresponds to the following picture?:Traditional Korean clothes:data_exercise30/i5-3.png
 Which cultural area corresponds to the following picture?:Da Nang Golden Bridge:data_exercise30/i5-4.png
 Which cultural area corresponds to the following picture?:Great Wall:data_exercise30/i5-5.png

- The question Q[0] is replaced with the text of qlabel, and the answer Q[1] is replaced with the global variable answer.
- The image uses the PhotoImage(file=file name) widget to create an image object img, and then uses the ilabel.config(image=img) method and ilabel.image = img to process it as the image of the label.

```
#Q[0]: - the problem
#Q[1]: - Answer
#Q[2]: - image file
qlabel.config(text=Q[0])
answer=Q[1]
img=PhotoImage(file=Q[2])
ilabel.config(image=img)
ilabel.image=img
```

- ❖ Checkanswer() function to check correct answers
- The user's answer contained in the Entry widget and the value of the answer variable, which is the correct answer to the problem obtained from getQuestion(), are the same to check whether the answer is correct.

```
def checkanswer(event):
    global answer, e
    if answer == e.get() :
        rlabel.config(text="Correct answer.")
    else:
        rlabel.config(text="Wrong answer.")
```

34.4.3.Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
from tkinter import *
window=Tk()
window.title("Take the quiz")
window.geometry("400x650+10+10")
#question label
qlabel=Label(window,width=100,text="")
qlabel.pack()
#image label
ilabel=Label(window)
ilabel.pack()
#Write the correct answer and press [Enter].
Label(window,text="Write the correct answer and press [Enter].").pack()
#Correct/Incorrect result label
rlabel=Label(window)
rlabel.pack()
file=open("data_exercise30/problem.txt","r",encoding="utf8")
p=file.readlines()
file.close()
i=-1
answer=""

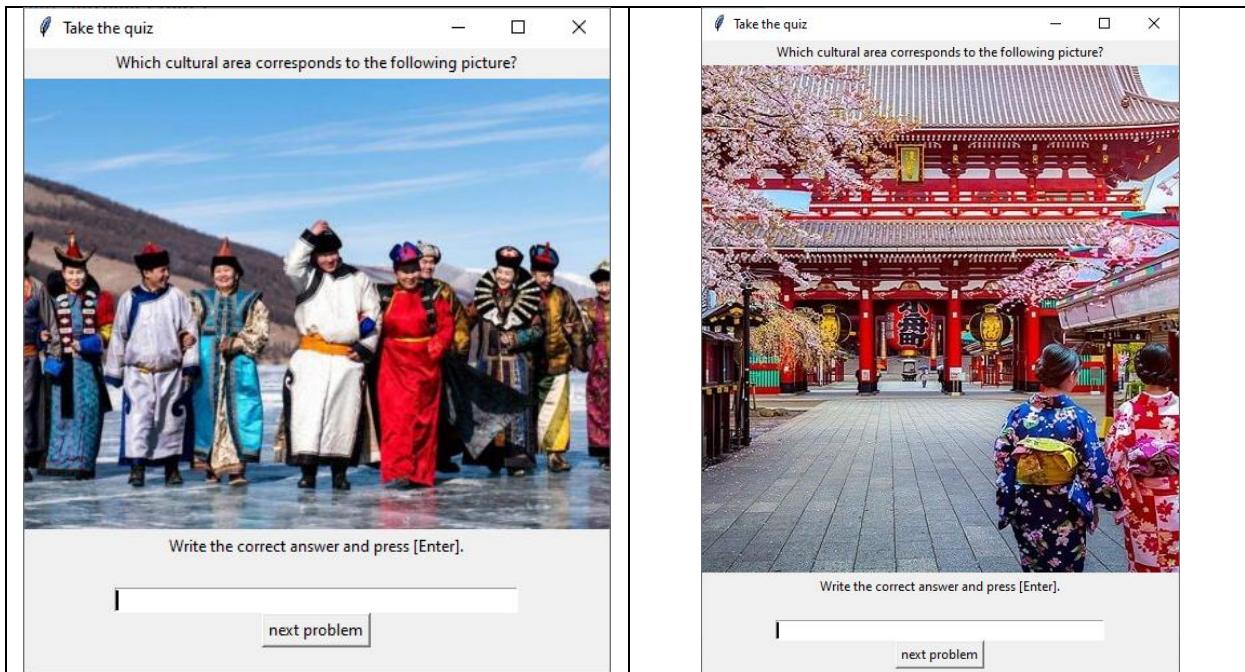
def checkanswer(event):
    global answer, e
    if answer == e.get() :
        rlabel.config(text="Correct answer.")
    else:
        rlabel.config(text="Wrong answer.")

#User answer input window
e=Entry(window,width=50)
e.bind("<Return>",checkanswer)
```

```
e.pack()
def getQuestion():
    global i,answer,e
    i +=1
    # Repeat because there are 4 problems
    if i >=4 : i =0
    e.delete(0,len(e.get()))
    rlabel.config(text="")
    aQuestion =p[i].strip()
    Q=aQuestion.split(":")
    #Q[0]: - the problem
    #Q[1]: - Answer
    #Q[2]: - image file
    qlabel.config(text=Q[0])
    answer=Q[1]
    img=PhotoImage(file=Q[2])
    ilabel.config(image=img)
    ilabel.image=img
Button(window,text="next problem",command=getQuestion).pack()
getQuestion()
window.mainloop()
```

34.5. Testing and debugging

Run program and see the results:



<p>Take the quiz</p> <p>— □ ×</p> <p>Which cultural area corresponds to the following picture?</p>  <p>Write the correct answer and press [Enter].</p> <input type="text"/> <p>next problem</p>	<p>Take the quiz</p> <p>— □ ×</p> <p>Which cultural area corresponds to the following picture?</p>  <p>Write the correct answer and press [Enter].</p> <input type="text"/> <p>next problem</p>
--	---

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise30.py

35. Exercise 31 - Quiz Game (improve)

35.1. Problem situation

Modify exercise 28 to a multiple-choice problem type that can answer the answer using the Radiobutton widget in the process of answering the problem.

35.2. Programming-the whole program

These codes I improved from Exercise 30, re-design GUI for quiz game

1. Add 4 radio button to show sample question (4 question)
2. Add more data for problem.txt
3. Create a FileFactory class, use to read a text file and return a List string
4. Create a Question class, it is a model class for a question
 - includes attributes: problemContent(string), problemAnser(string), image (string)
 - isCorrect method to check user correct nor not
5. Create a ListQuestion to store list of question from file problem.txt
 - 5.1 read all data from probem.txt and store to a set object
 - 5.2 add isEmpty method to check list question is empty or not
 - 5.3 add getQuestion method to get question object at the current postion
 - 5.4 add sample method to get random and shuffle the question
 - 5.5 add sizeOfList method to return size of the element
6. Create a QuizeUI class to design the GUI
 - all the UI is designed in this class
 - process show question when user click next question
 - show the result when user answer the question
 - add center place for the screen of the desktop

Link for data:

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise31

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni

#Description:
#These codes I improved from Exercise 30
#re-design GUI for quiz game
#1. Add 4 radio button to show sample question (4 question)
#2. Add more data for problem.txt
#3. Create a FileFactory class, use to read a text file and return a List string
#4. Create a Question class, it is a model class for a question
```

```
# includes attributes: problemContent(string), problemAnser(string), image (string)
# isCorrect method to check user correct nor not
#5. Create a ListQuestion to store list of question from file problem.txt
# 5.1 read all data from probem.txt and store to a set object
# 5.2 add isEmpty method to check list question is empty or not
# 5.3 add getQuestion method to get question object at the current postion
# 5.4 add sample method to get random and shuffle the question
# 5.5 add sizeOfList method to return size of the element
#6. Create a QuizeUI class to design the GUI
# all the UI is designed in this class
# process show question when user click next question
# show the result when user answer the question
# add center place for the screen of the desktop
import random
from tkinter import *
#This class use to text file factory
class FileFactory:
    questionBank="data_exercise31/problem.txt"
    encoding="utf8"
    mode="r"
    #this function use to read the content of file
    #return a List String
    def readData(self):
        file = open(self.questionBank, self.mode, encoding=self.encoding)
        datas = file.readlines()
        file.close()
        return datas
    #this class use to build a model of question
    #each line in the file, we will build a model for question object
    class Question:
        problemContent=None
        problemAnser=None
        image=None
        def __init__(self,problemContent,problemAnser,image):
            self.problemContent=problemContent
            self.problemAnser=problemAnser
            self.image=image
        def isCorrect(self,answer):
            return self.problemAnser==answer
    #This class use to store all question model object
    #check empty, get sample , getQuestion current postion
    class ListQuestion:
        questions=set()
        def __init__(self):
            #constructor read all data in file and store to a set
            datas=FileFactory().readData()
            for item in datas:
                arrDataOfItem =item.strip().split(":")
                if len(arrDataOfItem) == 3:
                    question =Question(arrDataOfItem[0],arrDataOfItem[1],arrDataOfItem[2])
                    self.questions.add(question)
        #this function check the listquestion is empty or not
        def isEmpty(self):
            return len(self.questions)<=0
        #this function return question at index position
```

```
def getQuestion(self,index):
    if self.isEmpty():
        return None
    if index<0 or index>=len(self.questions):
        return None
    return list(self.questions)[index]
#function to get sample 4 question
#and shuttle the question
def sample(self,index):
    currentQuestion = self.getQuestion(index)

    currentSet=set()
    currentSet.add(currentQuestion)
    remainList=self.questions-currentSet

    samples=random.sample(remainList,3)
    samples.append(currentQuestion)
    random.shuffle(samples)
    return samples
#this function return size of the list
def sizeOfList(self):
    return len(self.questions)
#this class use to design GUI for the Quiz game
# all the UI is designed in this class
# process show question when user click next question
# show the result when user answer the question
# add center place for the screen of the desktop
class QuizeUI:
    window = None
    qlabel = None
    ilabel = None
    rlabel = None
    radVars =None
    questionBanks = None
    currentPosition=-1
    currentQuestion=None
    #4 radio buttons
    rad1 = None
    rad2 = None
    rad3 = None
    rad4 = None
    def __init__(self):
        self.window = Tk()
        # question label
        self qlabel = Label(self.window, width=100, text="",fg="blue")
        self qlabel.pack()
        # image label
        self ilabel = Label(self.window)
        self ilabel.pack()
        Label(self.window, text="Choose the correct answer and click [Next Question]").pack()
        # Correct/Incorrect result label
        self rlabel = Label(self.window, text="")
        self rlabel.pack()
        frame = Frame(self.window)
        self.radVars = StringVar(frame, "<none>")
```

```

self.rad1 = Radiobutton(frame, text="", value="", variable=self.radVars, command=self.selection)
self.rad1.grid(row=0, column=0)
self.rad2 = Radiobutton(frame, text="", value="", variable=self.radVars, command=self.selection)
self.rad2.grid(row=0, column=1)
self.rad3 = Radiobutton(frame, text="", value="", variable=self.radVars, command=self.selection)
self.rad3.grid(row=0, column=2)
self.rad4 = Radiobutton(frame, text="", value="", variable=self.radVars, command=self.selection)
self.rad4.grid(row=0, column=3)
frame.pack()
#Next problem
Button(self.window, text="Next Question",command=self.checkAnswerAndMoveNextQuestion).pack()

self.questionBanks = ListQuestion()
if self.questionBanks.isEmpty() == False:
    self.currentPosition = 0
    self.showQuestion()
#this function use to show the value of radiobutton when user selection
def selection(self):
    s = "You selected the option [" + str(self.radVars.get())+"]"
    self.rlabel.config(text=s,fg='blue')
#This function use to show the next question and
def showQuestion(self):
    self.currentQuestion=self.questionBanks.getQuestion(self.currentPosition)
    self qlabel.config(text=self.currentQuestion.problemContent)
    img = PhotoImage(file=self.currentQuestion.image).subsample(2, 2)
    self.ilabel.config(image=img)
    self.ilabel.image = img
    randomAnswers= self.questionBanks.sample(self.currentPosition)
    self.rad1.config(text=randomAnswers[0].problemAnser,value=randomAnswers[0].problemAnser)
    self.rad2.config(text=randomAnswers[1].problemAnser,value=randomAnswers[1].problemAnser)
    self.rad3.config(text=randomAnswers[2].problemAnser,value=randomAnswers[2].problemAnser)
    self.rad4.config(text=randomAnswers[3].problemAnser,value=randomAnswers[3].problemAnser)
    self.currentPosition +=1

# this function use to the result of the current question
def checkAnswerAndMoveNextQuestion(self):
    if self.currentQuestion == None:
        return
    if self.currentQuestion.isCorrect(str(self.radVars.get())) == True:
        self.rlabel.config(text="You are right", fg='blue')
    else:
        self.rlabel.config(text="You are wrong!", fg='red')
    if self.currentPosition >= self.questionBanks.sizeOfList():
        self.currentPosition=0
        self.radVars.set("<none>")
        self.showQuestion()
#this function use to set the screen is center of the desktop
def center(self,win):
    win.update_idletasks()
    width = win.winfo_width()
    frm_width = win.winfo_rootx() - win.winfo_x()
    win_width = width + 2 * frm_width
    height = win.winfo_height()
    titlebar_height = win.winfo_rooty() - win.winfo_y()
    win_height = height + titlebar_height + frm_width

```

```

x = win.winfo_screenwidth() // 2 - win_width // 2
y = win.winfo_screenheight() // 2 - win_height // 2
win.geometry('{0x0}+{0}+{0}'.format(width, height, x, y))
win.deiconify()

#this function use to show the GUI of the Quiz
def showUI(self):
    # Take the quiz
    self.window.title("Take the quiz- Improved")
    self.window.geometry("550x400")
    self.center(self.window)
    self.window.mainloop()

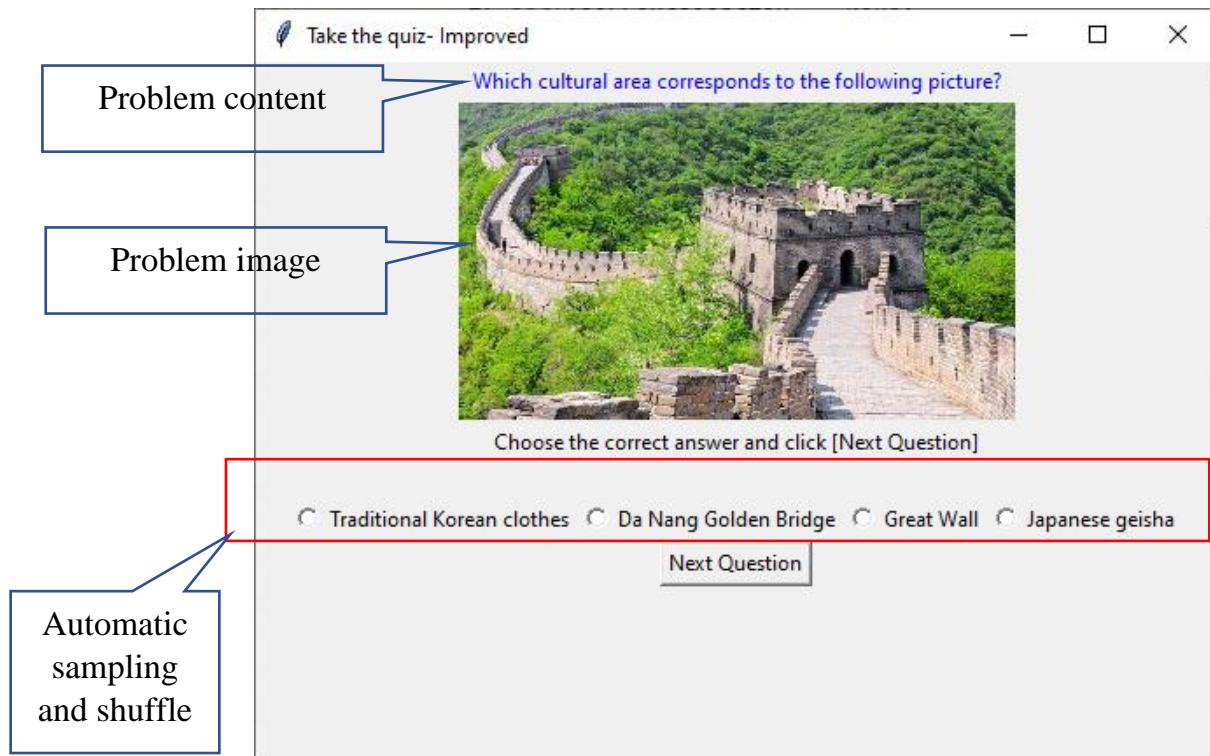
#call showUI() to start the program
QuizeUI().showUI()

```

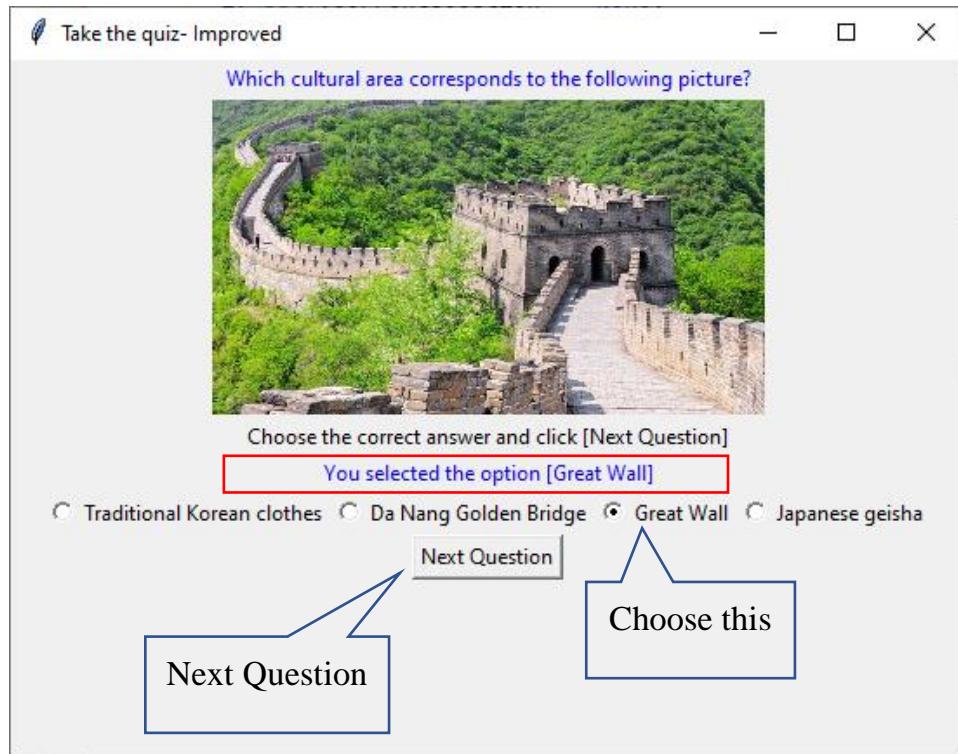
35.3. Testing and debugging

Here are some testing cases for the programming:

Run program to start:



If user choose the radio, the content of radio will be shown on the GUI:

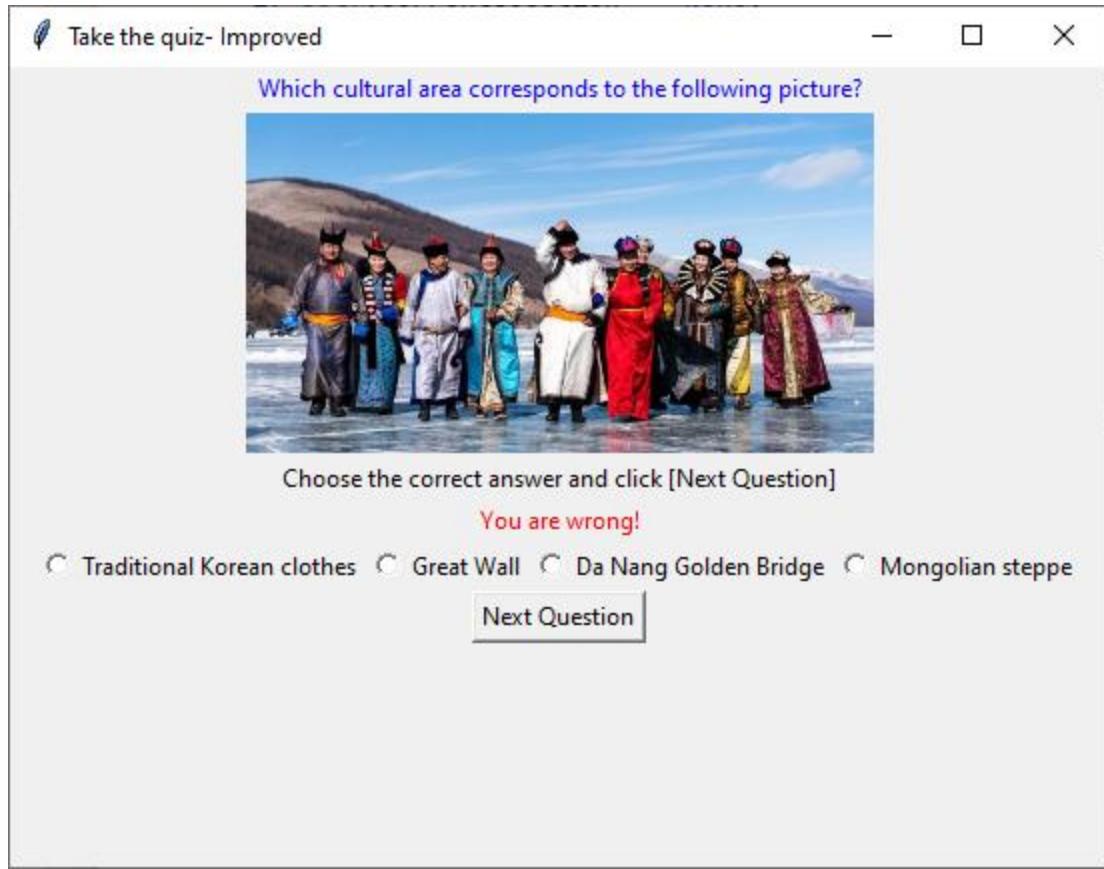


And user clicks Next Question button:

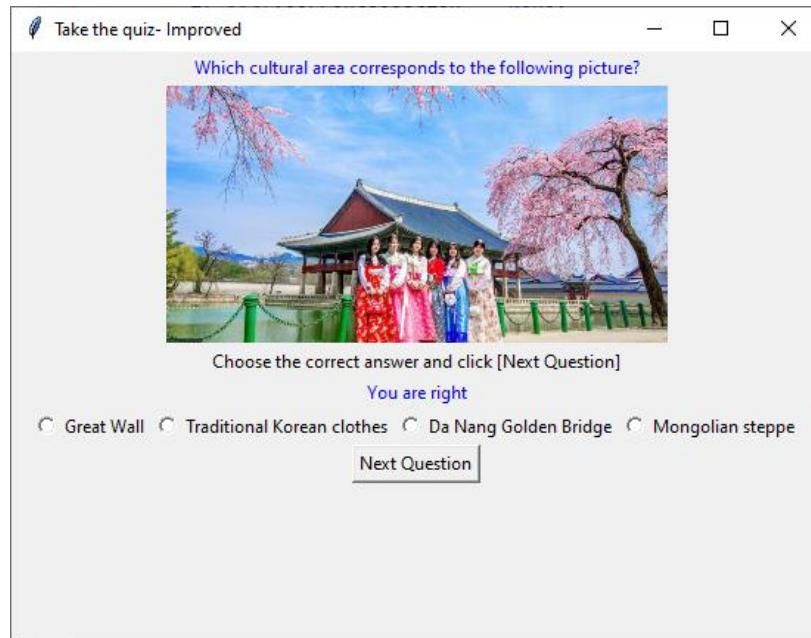
If user answers but not right:



In case, user choose “Mongolian steppe” (it is wrong):



Now Clicking Next Question to see the result and next question:



We can test with many test cases. The program will repeat the list of the question.

Take the quiz- Improved

Which cultural area corresponds to the following picture?



Choose the correct answer and click [Next Question]

You are right

Da Nang Golden Bridge Traditional Korean clothes Mongolian steppe Japanese geisha

[Next Question](#)

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise31.py

36. Exercise 32 – Visualize fine dust data

36.1. Learning Objectives

36.1.1. Basic learning objectives

- Matplotlib, bar(), subplot()

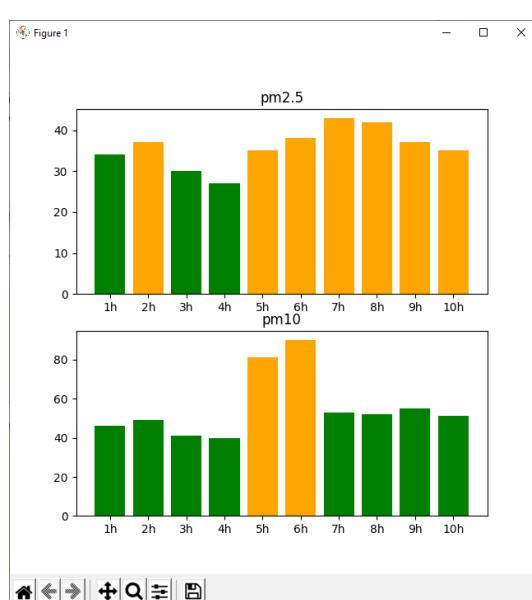
36.1.2. In-depth learning objectives

- Pandas, DataFrame()

36.1.3. Problem situation

```
hour=['1h','2h','3h','4h','5h','6h','7h','8h','9h','10h']
pm25=np.array([34,37,30,27,35,38,43,42,37,35])
pm10=np.array([46,49,41,40,81,90,53,52,55,51]).
```

How to show data chart:



36.2. Problem analysis

36.2.1. Input

- Provides fine dust data as initial values

36.2.2. Output

- ultrafine dust bar graph
- fine dust bar graph

36.2.3. Problem decomposition

- Displaying fine dust and ultrafine dust data
- Representing fine dust as a bar graph
- Representing ultrafine dust as a bar graph

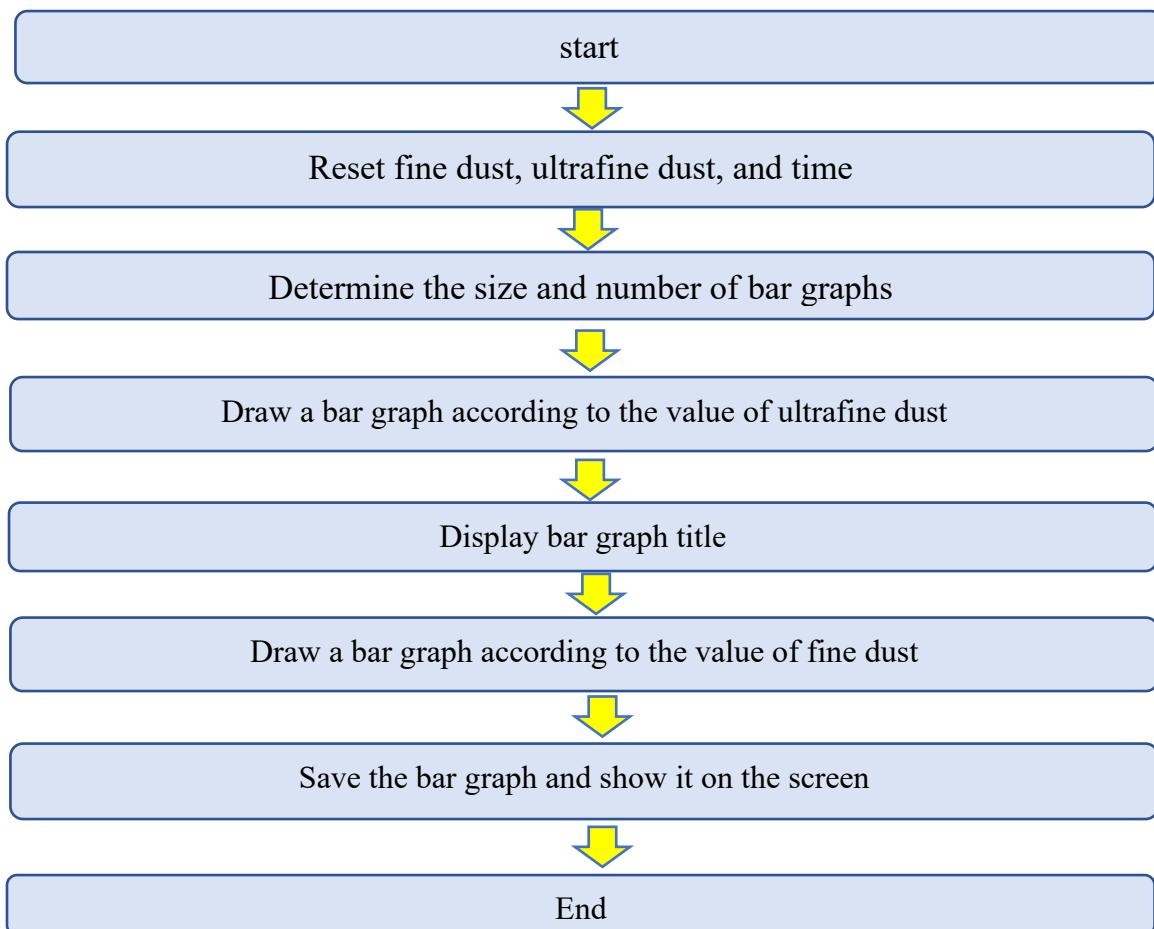
36.2.4. Thinking of users

- Consider which color to use
- Consider how to display the horizontal and vertical axes

36.2.5. Data

- Array of integers to represent ultrafine dust
- integer array to represent fine dust
- A list of strings to represent the time from 1:00 to 10:00

36.3. Full algorithm skeleton



36.4. Programming

36.4.1. Variables

- hour: String list from 1 o'clock to 10 o'clock
- pm25: Integer array of ultrafine dust over time
- pm10: Integer array of fine dust over time

36.4.2. Some explanations

Import packages for drawing numpy arrays and bar graphs

Initializing list and array to store time, fine dust, and ultrafine dust data

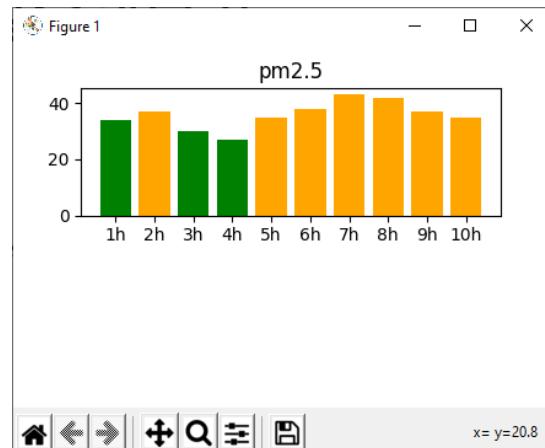
```
import numpy as np
import matplotlib.pyplot as plt

hour=['1h','2h','3h','4h','5h','6h','7h','8h','9h','10h']
pm25=np.array([34,37,30,27,35,38,43,42,37,35])
pm10=np.array([46,49,41,40,81,90,53,52,55,51])
```

- The program below determines the size of the bar graph and the layout of the bar graph.
- Specify the color of the bar graph according to the fine dust value
- Display legend and set bar graph title

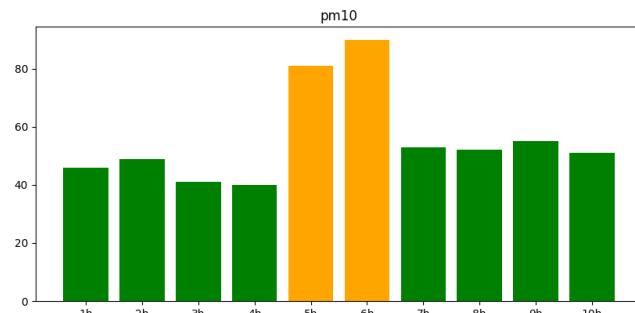
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style

hour=['1h','2h','3h','4h','5h','6h','7h','8h','9h','10h']
pm25=np.array([34,37,30,27,35,38,43,42,37,35])
pm10=np.array([46,49,41,40,81,90,53,52,55,51])
plt.figure(figsize=(10,10))
plt.subplot(2,1,1)
for i in range(10):
    if pm25[i]<15:
        plt.bar(hour[i],pm25[i],color='blue')
    elif 15<=pm25[i]<35:
        plt.bar(hour[i],pm25[i],color='green')
    elif 35<=pm25[i]<75:
        plt.bar(hour[i],pm25[i],color='orange')
    elif pm25[i]>=75:
        plt.bar(hour[i],pm25[i],color='red')
plt.title('pm2.5')
plt.show()
```



- The following program sets the layout of the bar graph below the bar graph drawn earlier.
- Specify the color of the bar graph according to the fine dust value
- Decide on a bar graph title and display it on the screen

```
plt.subplot(2,1,2)
for i in range(10):
    if pm10[i]<30:
        plt.bar(hour[i],pm10[i],color='blue')
    elif 30<=pm10[i]<80:
        plt.bar(hour[i],pm10[i],color='green')
    elif 80<=pm10[i]<150:
        plt.bar(hour[i],pm10[i],color='orange')
    elif pm10[i]>=150:
        plt.bar(hour[i],pm10[i],color='red')
plt.title('pm10')
plt.show()
```



36.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import numpy as np
import matplotlib.pyplot as plt

hour=['1h','2h','3h','4h','5h','6h','7h','8h','9h','10h']
pm25=np.array([34,37,30,27,35,38,43,42,37,35])
pm10=np.array([46,49,41,40,81,90,53,52,55,51])
#Set the picture horizontal and vertical size
plt.figure(figsize=(10,10))
#1 of 2 rows
plt.subplot(2,1,1)

for i in range(10):

    if pm25[i]<15:
        plt.bar(hour[i],pm25[i],color='blue')
    elif 15<=pm25[i]<35:
        plt.bar(hour[i],pm25[i],color='green')
    elif 35<=pm25[i]<75:
        plt.bar(hour[i],pm25[i],color='orange')
    elif pm25[i]>=75:
        plt.bar(hour[i],pm25[i],color='red')

plt.title('pm2.5')
#2 of 2 rows
```

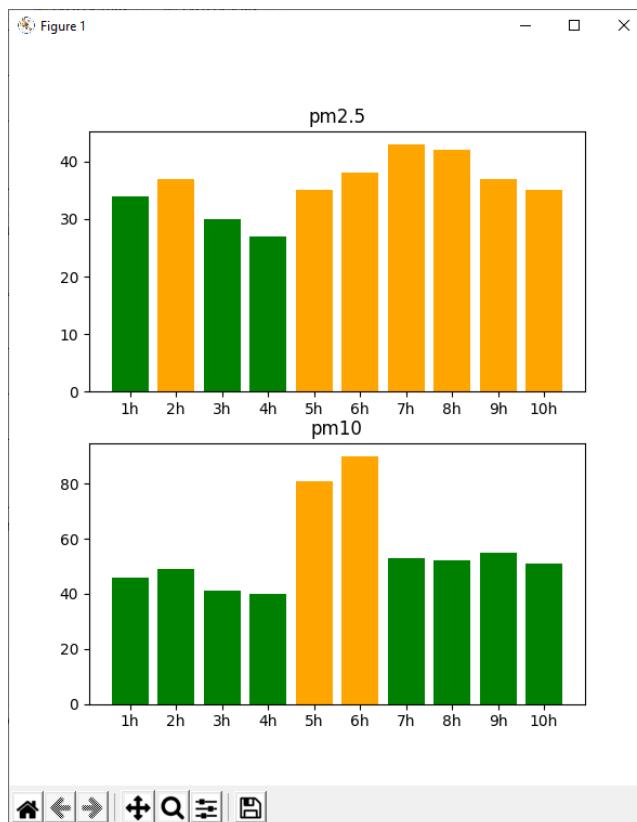
```
plt.subplot(2,1,2)

for i in range(10):
    if pm10[i]<30:
        plt.bar(hour[i],pm10[i],color='blue')
    elif 30<=pm10[i]<80:
        plt.bar(hour[i],pm10[i],color='green')
    elif 80<=pm10[i]<150:
        plt.bar(hour[i],pm10[i],color='orange')
    elif pm10[i]>=150:
        plt.bar(hour[i],pm10[i],color='red')

plt.title('pm10')
plt.show()
```

36.5. Testing and debugging

Run program and see the results:



I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise32.py

37. Exercise 33 - Visualize fine dust data (improve)

37.1. Problem situation

Let's visualize the ultrafine dust and fine dust data using Pandas' DataFrame object. Also, let's modify it so that it can visualize and show the sum of the ultrafine dust and fine dust data.

37.2. Programming-the whole program

These codes I improved from Exercise 32. Visualize the ultrafine dust and fine dust data using Pandas' DataFrame object. Also, it so that it can visualize and show the sum of the ultrafine dust and fine dust data.

1. I changed the value of ultrafine dust and fine dust to get full of colors list
2. Write 2 methods mapping Color for ultrafine dust and fine dust
3. Write getSumOfDust function reuse both ultrafine dust and fine dust
4. Write showNotation function for plot
5. Use DataFrame of Pandas object and combine with pyplot to show the chart

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 32.
#visualize the ultrafine dust and fine dust data using Pandas' DataFrame object.
# Also,it so that it can visualize and show the sum of the ultrafine dust and fine dust data.
# 1. I changed the value of ultrafine dust and fine dust to get full of colors list
# 2. Write 2 methods mapping Color for ultrafine dust and fine dust
# 3. Write getSumOfDust function reuse both ultrafine dust and fine dust
# 4. Write showNotation function for plot
# 5. Use DataFrame of Pandas object and combine with pyplot to show the chart
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#This function use to mapping color for pm25
def mappingColorPm25(value):
    if value < 15:
        return 'blue'
    elif 15 <= value < 35:
        return 'green'
    elif 35 <= value < 75:
        return 'orange'
    elif value >= 75:
        return 'red'
```

```

#This function use to mapping color for Pm10
def mappingColorPm10(value):
    if value < 30:
        return 'blue'
    elif 30 <= value < 80:
        return 'green'
    elif 80 <= value < 150:
        return 'orange'
    elif value >= 150:
        return 'red'

#this function use to get sum of value for pm25 or pm10
def getSumOfDust(data):
    sum=0
    for x in data:
        sum = sum + x
    return sum

#this function use to show the notation for pm25 or pm10
def showNotation(ax):
    for i in ax.patches:
        ax.text(i.get_x() + 0.1,
                i.get_height() + .5,
                i.get_height(),
                fontsize=8,
                color='dimgrey')

#set the subplots and some configuration
fig, (ax1, ax2) = plt.subplots(2, 1)
fig.subplots_adjust(hspace=0.5)
fig.set_size_inches(8, 8)
#hour to show the chart
hour=['1h','2h','3h','4h','5h','6h','7h','8h','9h','10h']
#pm25 value
#pm25=np.array([34,37,30,14,35,75,43,42,37,35])
pm25=np.array([14,30,34,35,35,37,37,42,43,75])
#pm25=np.array([34,37,30,27,35,38,43,42,37,35])
#call DataFrame of Pandas object for pm25
dfpm25 = pd.DataFrame({'Ultrafine Dust Data':hour, 'pm25':pm25})
#create color mapping list for pm25
pm25colors = [mappingColorPm25(x) for x in dfpm25["pm25"]]
#get the sum of pm25
sumpm25=getSumOfDust(dfpm25["pm25"])
#draw bar chart
axpm25 = dfpm25.plot.bar(x='Ultrafine Dust Data', y='pm25',
                           rot=0,color=pm25colors,ax=ax1,legend=False)
#set title for pm25 char
axpm25.set_title("PM25-Sum Of Ultrafine Dust Data is "+str(sumpm25))
#show notation for pm25
showNotation(axpm25)
#pm10 value
#pm10=np.array([46,29,41,40,81,90,53,52,151,51])
pm10=np.array([29,40,41,46,51,52,53,81,90,151])
#pm10=np.array([46,49,41,40,81,90,53,52,55,51])
#call DataFrame of Pandas object for pm10
dfpm10 = pd.DataFrame({'Fine Dust Data':hour, 'pm10':pm10})
#create color mapping list for pm10
pm10colors = [mappingColorPm10(x) for x in dfpm10["pm10"]]

```

```
#get the sum of pm10
sumpm10=getSumOfDust(dfpm10["pm10"])
#draw bar chart
axpm10 = dfpm10.plot.bar(x='Fine Dust Data', y='pm10', rot=0,color=pm10colors,ax=ax2,legend=False)
#set title for pm10 char
axpm10.set_title("PM10-Sum Of Fine Dust Data is "+str(sumpm10))
#show notation for pm10
showNotation(axpm10)
#Show the chart
plt.show()
```

37.3. Testing and debugging

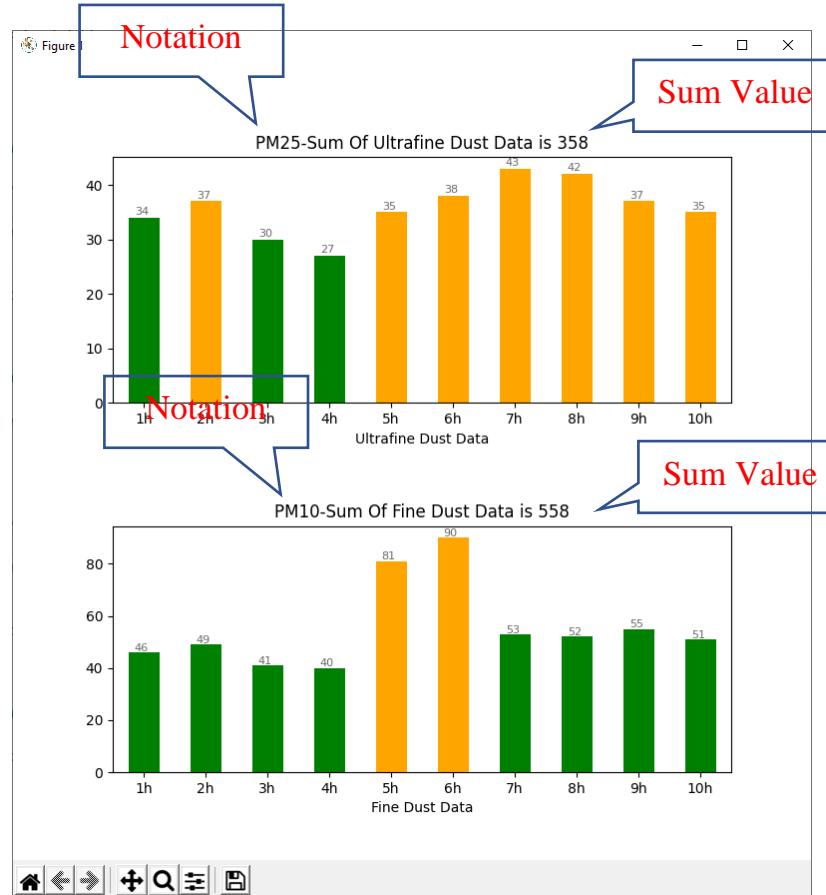
Here are some testing cases for the programming:

Test case 1:

The default value from Professor document:

```
pm25=np.array([34, 37, 30, 27, 35, 38, 43, 42, 37, 35])
pm10=np.array([46, 49, 41, 40, 81, 90, 53, 52, 55, 51])
```

The result: The chart show is the same Professor document. And also, we have sum and notation:



Test case 2:

Change the array of value:

```
pm25=np.array([34, 37, 30, 14, 35, 75, 43, 42, 37, 35])
pm10=np.array([46, 29, 41, 40, 81, 90, 53, 52, 151, 51])
```

The result: The chart show is changed with full color list mapping. And also, we have sum and notation are changed.



Test case 3:

Change the array of value, make sorting array:

```
pm25=np.array([14, 30, 34, 35, 35, 37, 37, 42, 43, 75])
pm10=np.array([29, 40, 41, 46, 51, 52, 53, 81, 90, 151])
```

The result: The chart show is changed with full color list mapping. And also, we have sum and notation are changed.



I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise33.py

38. Exercise 34 – Monte Carlo simulation

38.1. Learning Objectives

38.1.1. Basic learning objectives

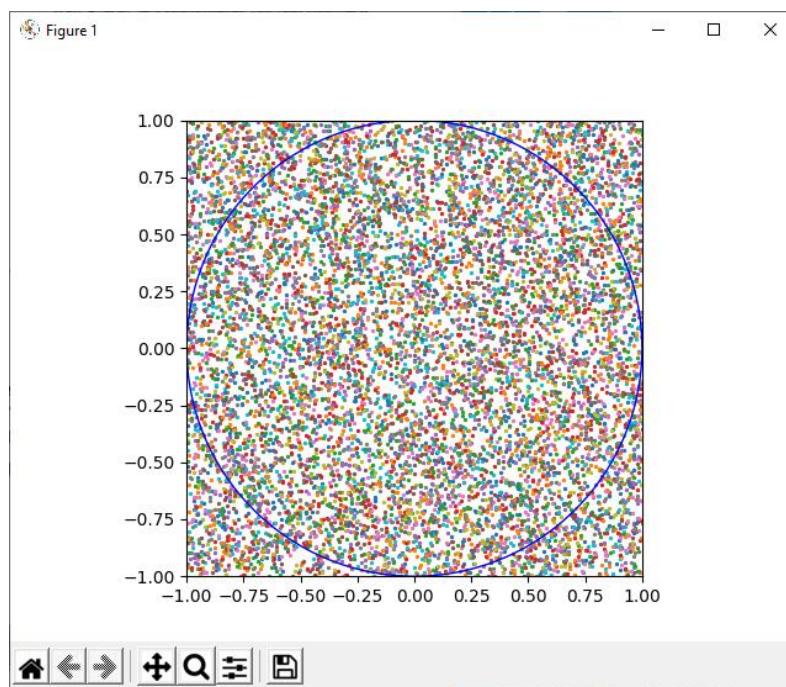
- Know what a Monte Carlo simulation is
- It is possible to write a program that finds the percentage of a certain percentage through Monte Carlo simulation
- You can visualize it using the Scatter() method.
- Understand and be able to use the Uniform() method

38.1.2. In-depth learning objectives

- A Monte Carlo simulation can be applied to find the area

38.1.3. Problem situation

How to calculate PI value by Monte Carlo simulation?



38.2. Problem analysis

38.2.1. Input

- Number of simulations

38.2.2. Output

- Circumference value obtained by calculation

- Rectangle, circle, point

38.2.3. Problem decomposition

- Circle drawing
- Draw dots
- Find the number of points in a circle

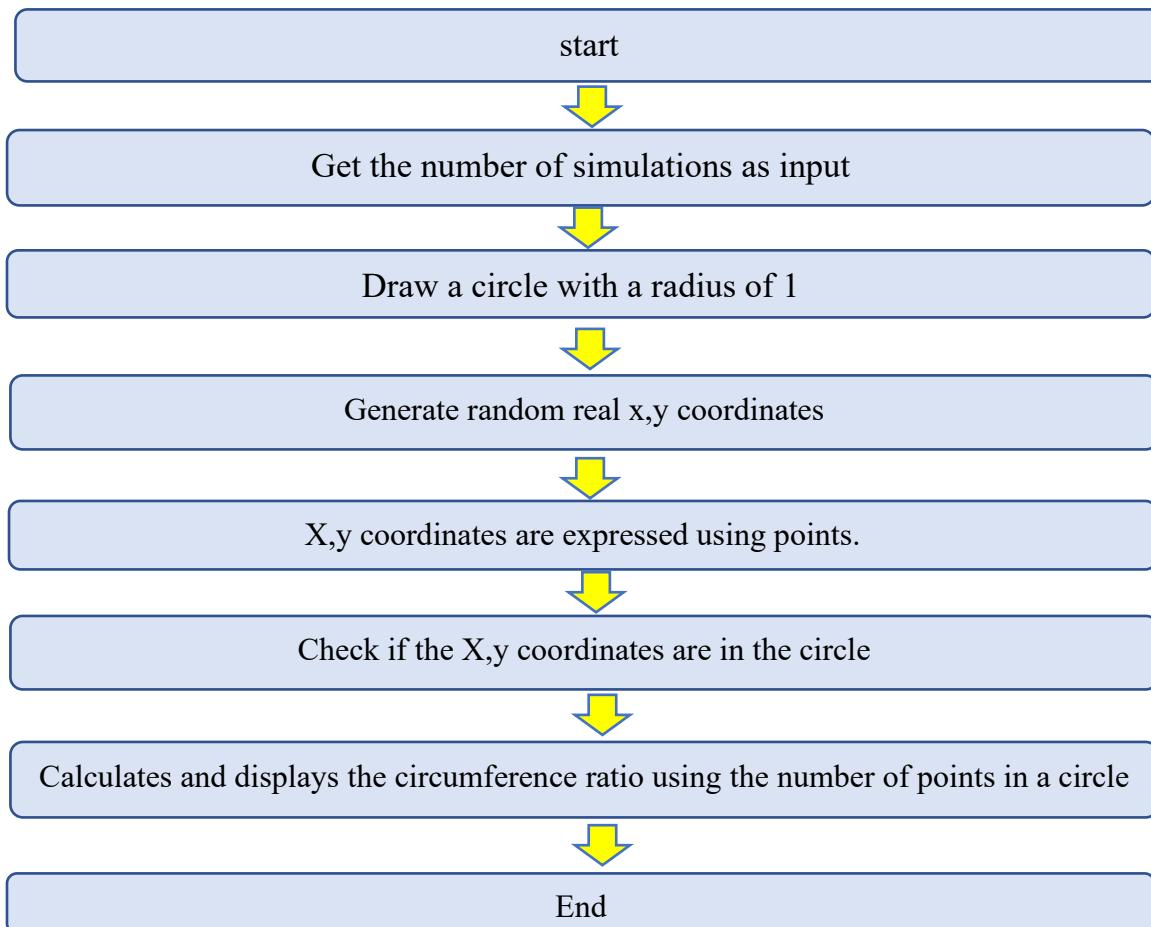
38.2.4. Thinking of users

- Visualize simulation results to users
- Lets you choose the number of simulations you want
- Show points inside and outside the circle

38.2.5. Data

- real x,y coordinates
- number of points in a circle
- Total number of simulations

38.3. Full algorithm skeleton



38.4. Programming

38.4.1. Variables

- inCircle: the number of points in the circle
- simCount: number of simulations
- x,y: x,y coordinate values

38.4.2. Some explanations

- A simulation method to obtain probabilistic values through iterative calculations using random numbers
- Monte Carlo simulation is widely used to probabilistically solve complex problems in numerical analysis, statistical mechanics, and science and engineering.
- Even in programs such as AlphaGo, an artificial intelligence Go program, the probability of winning and losing is calculated by playing countless Go games through Monte Carlo simulation.
- Monte Carlo simulation is mathematically based on the concept of statistics and probability fields that the characteristics of a sample randomly selected from a large population become closer to the characteristics of the population as the number increases.
- For example, the probability of a characteristic investigation with dice is obtained as close to 1/6 as the number of simulations increases.

- Import packages for random numbers and visualization
- Initialize the number of points in a circle to 0
- The more simulations you get, the closer you get to a more accurate value of perimeter, but the more simulations you run, the longer it takes

```
import random
import matplotlib.pyplot as plt

incircle=0

simCount=int(input("Maximum number of simulations?"))
```

- The following program creates a circle with origin (0, 0) and radius length 1.
- The inside of the circle is not filled, the circle is drawn in blue
- Use the add_patch() method to display the circle object on the graph.
- Use the set_aspect() method to set the ratio of the x-axis and y-axis equally

```
circle_center=(0,0)
circle_radius=1

c=plt.Circle(circle_center,circle_radius,ec='b',fill=False)

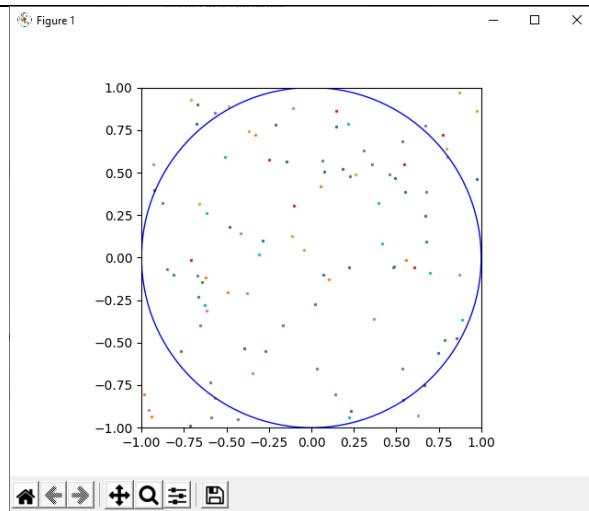
a=plt.axes(xlim=(-1,1),ylim=(-1,1))
a.add_patch(c)
a.set_aspect('equal')
```

- The following program creates x and y coordinates using a for loop.
- Display the X,y coordinates as a point of size 2 using the scatter() method.
- Check whether the Y coordinate value is within a circle with a radius of 1 using a conditional statement
- Calculate the circumference ratio by dividing the number of points in the circle by the total number of simulations

```
for i in range(simCount):
    # Create real number x coordinates from -1 to 1
    x=random.uniform(-1.0,1.0)
    # Create real y coordinates from -1 to 1
    y=random.uniform(-1.0,1.0)
    plt.scatter(x,y,s=2)

    dot_value=x*x+y*y
    # Test where the (x,y) coordinates are in the circle
    if dot_value<=1:
        incircle =incircle+1

print("Pi=",4*incircle/simCount)
plt.show()
```



- Line 6 - import the random module to perform the Monte Carlo simulation
- 11th line - the number of simulations entered by the user
- Lines 13 to 15 draw a circle with a radius of 1

```

6   import random
7   import matplotlib.pyplot as plt
8
9   incircle=0
10
11  simCount=int(input("Maximum number of simulations?"))
12
13  circle_center=(0,0)
14  circle_radius=1
15
16  c=plt.Circle(circle_center,circle_radius,ec='b',fill=False)
17
18  a=plt.axes(xlim=(-1,1),ylim=(-1,1))
19  a.add_patch(c)
20  a.set_aspect('equal')

```

- Lines 22 to 26 - Use random numbers to obtain x and y coordinate values and display them as small circles using the scatter() method.
- Lines 29 to 32 - Check whether the coordinates obtained using random numbers are in the circle, and if it is in the circle, increase the incircle value by 1.
- Line 34 - Calculate the percentage of circumference and print it

```

22  for i in range(simCount):
23      # Create real number x coordinates from -1 to 1
24      x=random.uniform(-1.0,1.0)
25      # Create real y coordinates from -1 to 1
26      y=random.uniform(-1.0,1.0)
27      plt.scatter(x,y,s=2)
28
29      dot_value=x*x+y*y
30      # Test where the (x,y) coordinates are in the circle
31      if dot_value<=1:
32          incircle =incircle+1
33
34  print("Pi=",4*incircle/simCount)
35  plt.show()

```

38.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import random
import matplotlib.pyplot as plt

incircle=0

simCount=int(input("Maximum number of simulations?"))

circle_center=(0,0)
circle_radius=1

c=plt.Circle(circle_center,circle_radius,ec='b',fill=False)

a=plt.axes(xlim=(-1,1),ylim=(-1,1))
a.add_patch(c)
a.set_aspect('equal')

for i in range(simCount):
    # Create real number x coordinates from -1 to 1
    x=random.uniform(-1.0,1.0)
    # Create real y coordinates from -1 to 1
    y=random.uniform(-1.0,1.0)
    plt.scatter(x,y,s=2)

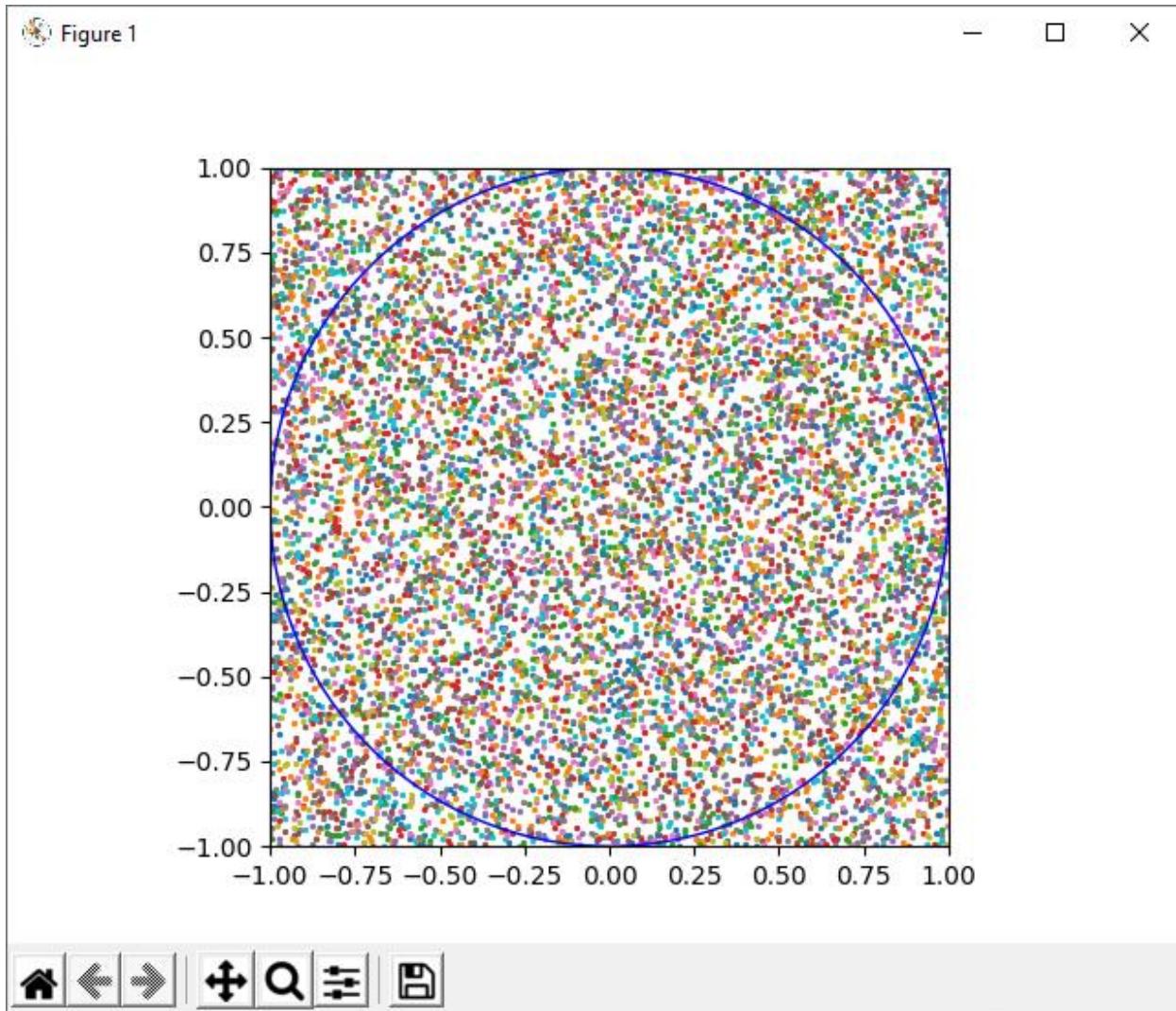
    dot_value=x*x+y*y
    # Test where the (x,y) coordinates are in the circle
    if dot_value<=1:
        incircle =incircle+1

print("Pi=",4*incircle/simCount)
plt.show()
```

38.5. Testing and debugging

Run program and see the results:

Maximum number of simulations?**10000**



39. Exercise 35 - Monte Carlo simulation (improve)

39.1. Problem situation

It can be seen that the value of pi can be obtained as a probabilistic approximation using the Monte Carlo simulation. Let's write a program that calculates the ratio of the area size of a circle with a circle radius of 1 to a circle with a circle radius of 2 using Monte Carlo simulation probabilistically.

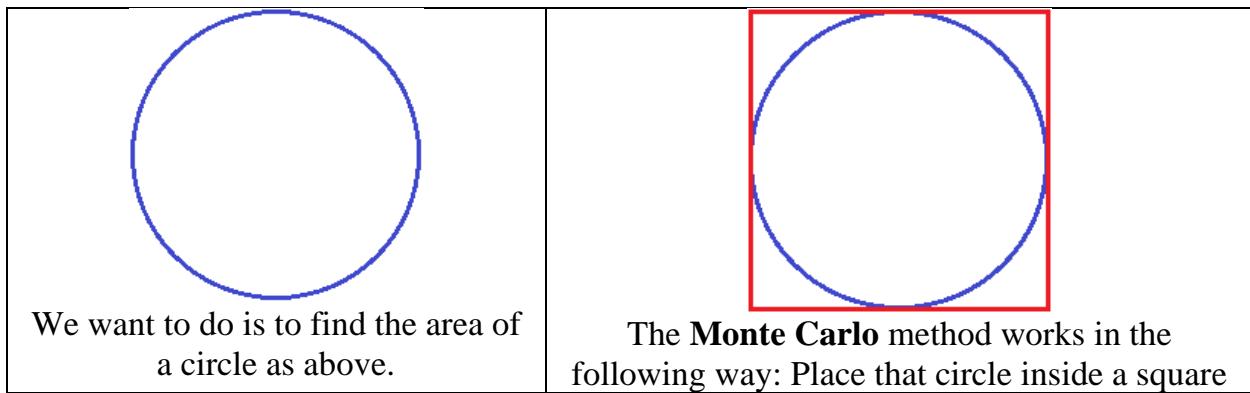
39.2. Programming-the whole program

These codes I improved from Exercise 34.

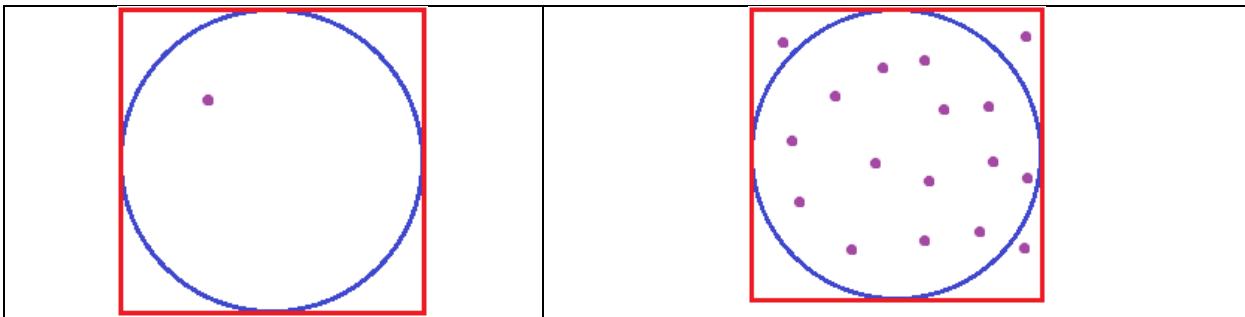
It can be seen that the value of pi can be obtained as a probabilistic approximation using the Monte Carlo simulation. Let's write a program that calculates the ratio of the area size of a circle with a circle radius of 1 to a circle with a circle radius of 2 using Monte Carlo simulation probabilistically.

1. Use subplot to display 2 graphic of Circle
2. Write isPointInCircle method to check random x,y is inside the circle
3. Write approximateCircleArea method to calculate CircleArea by Monte Carlo sampling
4. Test many case to compare the Approximate area

I research on the internet and some papers, So I explain the Monte Carlo method of finding the area of a circle below, [reference website link](#):

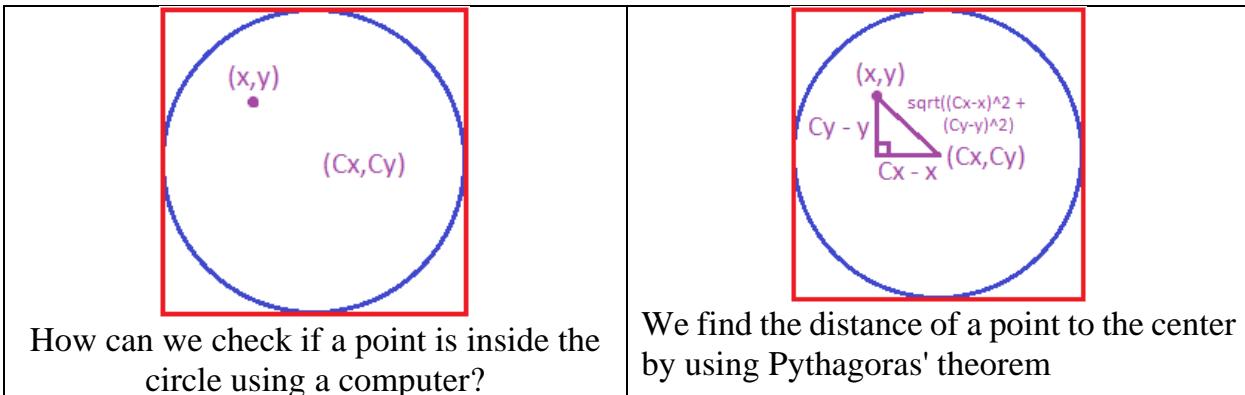


Keep in mind that the area of a square is easy to find. Now throw a point inside the square at random, which would result in something like this:



For example, in the above diagram there are 17 points in total but only 14 of the points lie inside the circle. So, we can say that approximately $14/17$ of the area inside the square is taken by the circle.

So, in the above example, the length of the side of the square is about 6cm, so the area of the square is $6 \times 6 = 36\text{cm}^2$, so the area of the circle is approximately $14/17 \times 36 = 29.64\text{cm}^2$.



All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 34
# It can be seen that the value of pi can be obtained as a probabilistic approximation
# using the Monte Carlo simulation. Let's write a program that calculates the ratio of
# the area size of a circle with a circle radius of 1 to a circle with a circle radius
# of 2 using Monte Carlo simulation probabilistically.
# 1.use subplot to display 2 graphic of Circle
# 2.Write isPointInCircle method to check random x,y is inside the circle
# 3.Write approximateCircleArea method to calculate CircleArea by Monte Carlo sampling
# 4.Test many case to comparing the Approximate area
import math
import random
import matplotlib.pyplot as plt

#function use to check the random x, y is inside the circle
```

```

def isPointInCircle(x, y, Cx, Cy, radius):
    return math.sqrt((x - Cx) ** 2 + (y - Cy) ** 2) <= radius
#function use to calculate area of circle use to sampling mento carlo
#ax: pilot for the circle
#thick: size of point drawing on the chart
#radius: radius for circle
#simCount: Maximum number of simulations
def approximateCircleArea(ax,thick, radius, simCount):
    squareSide = radius * 2
    Cx = radius
    Cy = radius
    pointsInside = 0
    for i in range(simCount):
        x = random.random() * squareSide
        y = random.random() * squareSide
        ax.scatter(x, y, s=thick)
        if (isPointInCircle(x, y, Cx, Cy, radius)):
            pointsInside = pointsInside + 1
    return pointsInside / simCount * squareSide ** 2

plt.figure(figsize=(5,3)) #그림 가로, 세로 크기 설정
#create subplot for Circle with radius =1
ax1=plt.subplot(1,2,1)
ax1.set_xlim([0, 2])
ax1.set_ylim([0, 2])
ax1.set_title("Circle with Radius = 1")
circle_center1=(1,1)
circle_radius1=1
#draw circle 1
c1=plt.Circle(circle_center1,circle_radius1,ec='b',fill=False)
paths1=ax1.add_patch(c1)
ax1.set_aspect('equal')
#create subplot for Circle with radius =2
ax2=plt.subplot(1,2,2)
ax2.set_xlim([0, 4])
ax2.set_ylim([0, 4])
ax2.set_title("Circle with Radius = 2")
circle_center2=(2,2)
circle_radius2=2
#draw circle 2
c2=plt.Circle(circle_center2,circle_radius2,ec='r',fill=False)

paths2=ax2.add_patch(c2)
ax2.set_aspect('equal')

simCount=int(input("Maximum number of simulations?"))
area1=approximateCircleArea(ax1,2,circle_radius1,simCount)
area2=approximateCircleArea(ax2,2,circle_radius2,simCount)
print("Area 1 =",area1)
print("Area 2 =",area2)
ax1.set_title("Area =" +str(area1))
ax2.set_title("Area =" +str(area2))
plt.show()

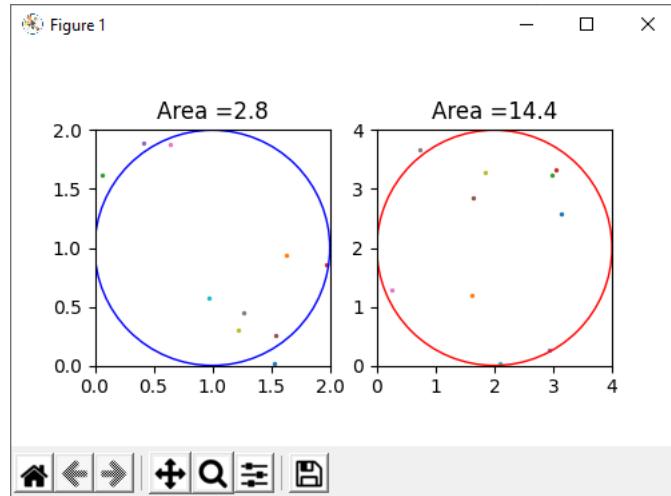
```

39.3. Testing and debugging

Here are some testing cases for the programming:

Test case 1:

SimCount =10

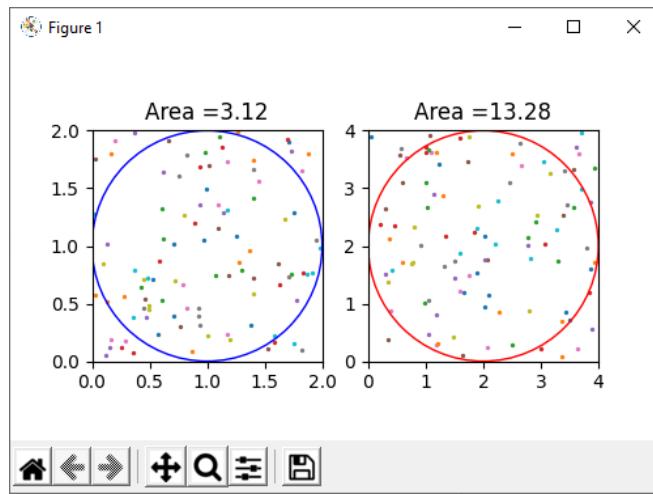


When Simcount =10, Area of circle with radius 1 is 2.8

Area of circle with radius 2 is 14.4

Test case 2:

SimCount =100

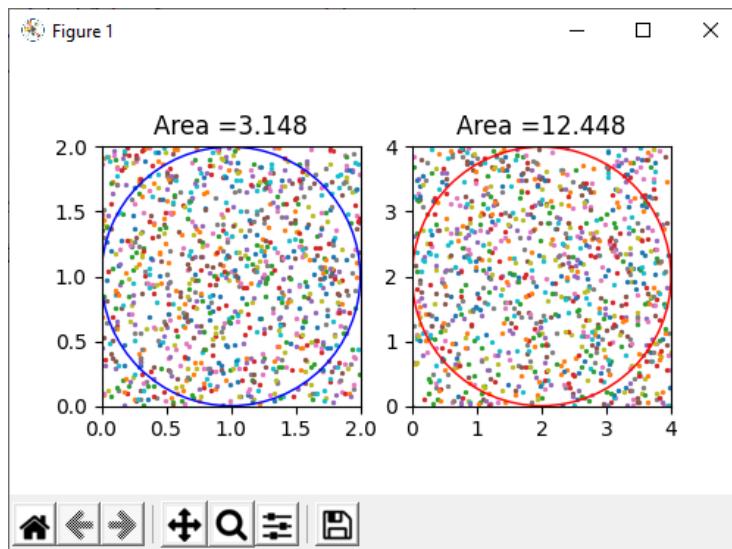


When Simcount =100, Area of circle with radius 1 is 3.12

Area of circle with radius 2 is 13.28

Test case 3:

SimCount =1000

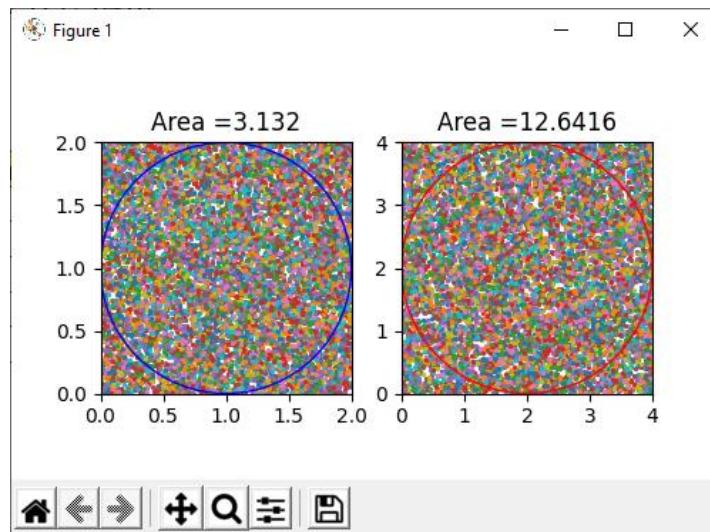


When Simcount =1000, Area of circle with radius 1 is 3.148

Area of circle with radius 2 is 12.448

Test case 4:

SimCount =10000



When Simcount =10000, Area of circle with radius 1 is 3.132

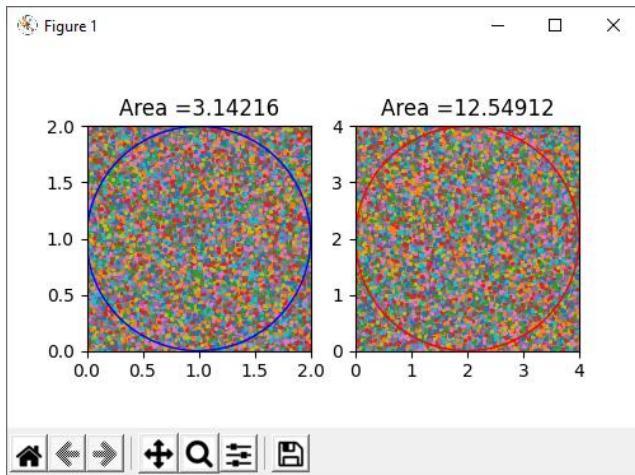
Area of circle with radius 2 is 12.6416

Test case 5:

SimCount =100000

When Simcount =100000, Area of circle with radius 1 is 3.14216

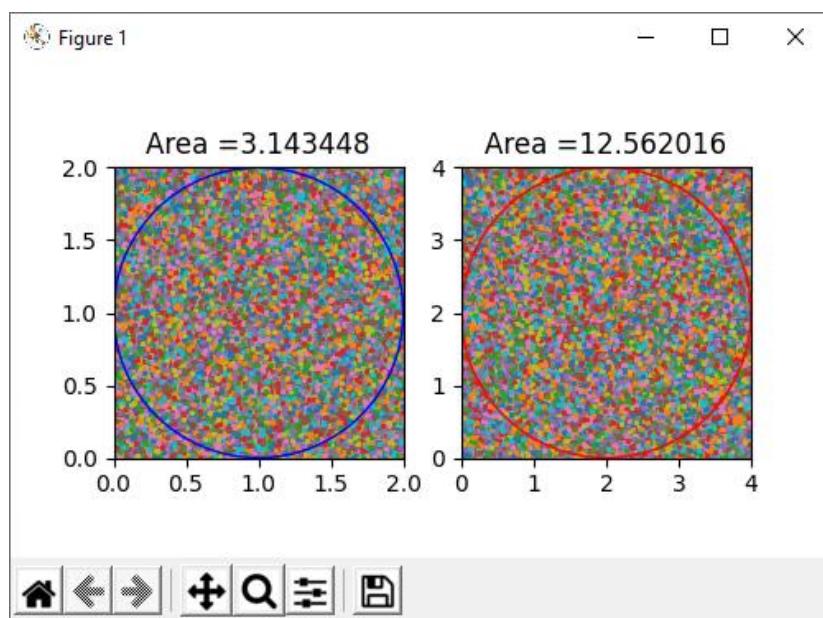
Area of circle with radius 2 is 12.54912

**Test case 6:**

SimCount =500000

When Simcount =500000, Area of circle with radius 1 is 3.143448

Area of circle with radius 2 is 12.562016



I report 6 test cases in the table below:

Test Case	SimCount	Approximate area 1 (Radius is 1)	Approximate area 2 (Radius is 2)
1	10	2.8	14.4
2	100	3.12	13.28
3	1000	3.148	12.448
4	10000	3.132	12.6416
5	100000	3.14216	12.54912
6	500000	3.143448	12.562016

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise35.py

40. Exercise 36 – Numeric Recognizer – Machine learning

40.1. Learning Objectives

40.1.1. Basic learning objectives

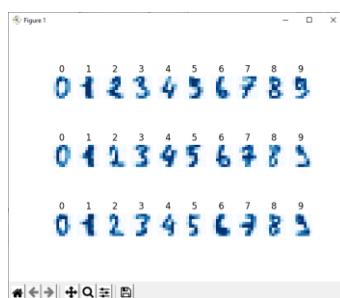
- You can write simple machine learning programs using scikit-learn.
- You can write a number recognition program using scikit-learn
- Understand the difference between training and testing in machine learning

40.1.2. In-depth learning objectives

- You can write a simple deep learning program using Keras.
- Character recognition programs can be written by applying deep learning techniques

40.1.3. Problem situation

How to recognize the number?



40.2. Problem analysis

40.2.1. Input

- Images

40.2.2. Output

- Numeric Recognizer

40.2.3. Problem decomposition

- Learning to Recognize Numbers
- Calculate the number recognition rate
- Recognize numbers written by users

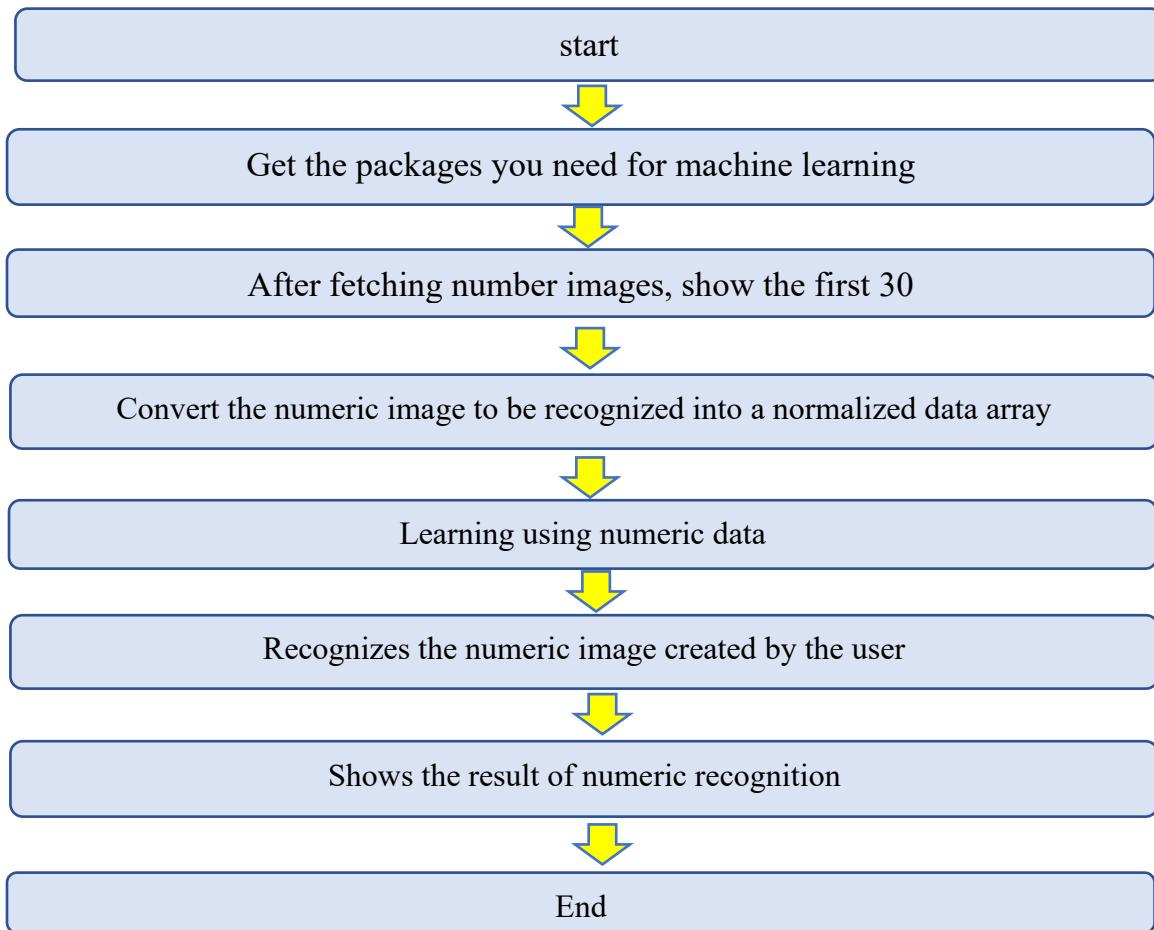
40.2.4. Thinking of users

- Consider a user interface for digit recognition
- Shows numeric image data type
- Notifies the result of numeric recognition

40.2.5.Data

- data array for numeric images
- recognition number

40.3. Full algorithm skeleton



40.4. Programming

40.4.1. Variables

- digitImage, data: a numeric array converted from a numeric image
- n: predictive digit integer
- digits: numeric data set

40.4.2. Some explanations

- Get numeric image data using `sklearn.datasets.load_digits()`
- Use the `train_test_split()` function to separate training data and test data
- Use `imshow()` to display the first 30 number images among the number images.

```
import sklearn.datasets
import matplotlib.pyplot as plt
import sklearn.svm
from PIL import Image
import numpy as np
from sklearn.model_selection import train_test_split
digits=sklearn.datasets.load_digits()
# If the value is set to 0, it is always separated in the same form.
x_train, x_test, y_train, y_test = train_test_split(digits.data,
                                                    digits.target, test_size=0.25, random_state=0)
for i in range(30):
    plt.subplot(3,10,i+1)
    plt.axis("off")
    plt.title(digits.target[i])
    plt.imshow(digits.images[i], cmap="Blues")
plt.show()
```

- Take an image in png format and convert it to a NumPy numeric array
- Customize the size and color depth to the image data set you want to compare
- Since the original image has values from 0 to 255, convert it to values from 0 to 16

```
def conv_image_to_data(filename):
    blackImage=Image.open(filename).convert('L') #gray scale
    blackImage=blackImage.resize((8,8))
    digitImage=np.asarray(blackImage, dtype=float)

    digitImage=16*np.divide(blackImage, 256)
    digitImage=np.floor(16-digitImage)
    digitImage=digitImage.flatten()
    plt.imshow(blackImage, cmap="Blues")
    plt.show()
    print(digitImage)

    return digitImage
```

- The following program is a part of learning using a learning model
- Use test data to see how accurate your predictions are
- Then, recognize what number the image of the number you have written is

```
learning_model=sklearn.svm.SVC(gamma=0.001)

learning_model.fit(x_train,y_train)
score = learning_model.score(x_test,y_test)
print("score=",score)

data=conv_image_to_data("data_exercise36\digit-7.png")

n=learning_model.predict([data])
print("What is the predicted number?",n)
```

Here is the link of images to test:

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise36

40.4.3. Programming-the whole program

The whole program is shown as below.

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
import sklearn.datasets
import matplotlib.pyplot as plt
import sklearn.svm
from PIL import Image
import numpy as np
from sklearn.model_selection import train_test_split

digits=sklearn.datasets.load_digits()
# If the value is set to 0, it is always separated in the same form.
x_train,x_test,y_train,y_test=train_test_split(digits.data,
                                               digits.target,test_size=0.25,random_state=0)
for i in range(30):
    plt.subplot(3,10,i+1)
    plt.axis("off")
    plt.title(digits.target[i])
    plt.imshow(digits.images[i],cmap="Blues")
plt.show()

def conv_image_to_data(filename):
    blackImage=Image.open(filename).convert('L')#gray scale
    blackImage=blackImage.resize((8,8))
    digitImage=np.asarray(blackImage,dtype=float)

    digitImage=16*np.divide(blackImage,256)
```

```

digitImage=np.floor(16-digitImage)
digitImage=digitImage.flatten()

plt.imshow(blackImage,cmap="Blues")
plt.show()
print(digitImage)

return digitImage

learning_model=sklearn.svm.SVC(gamma=0.001)

learning_model.fit(x_train,y_train)
score = learning_model.score(x_test,y_test)
print("score=",score)

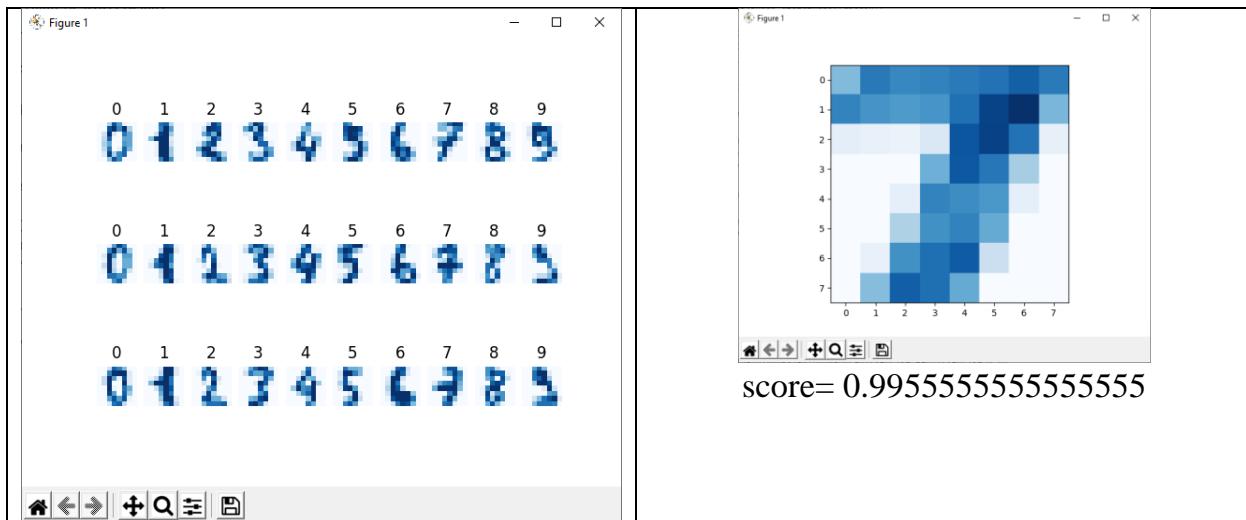
data=conv_image_to_data("data_exercise36\digit-7.png")

n=learning_model.predict([data])
print("What is the predicted number?",n)

```

40.5. Testing and debugging

Run program and see the results:



You should test with another images.

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise36.py

41. Exercise 37 - Numeric Recognizer – Deep learning

41.1. Problem situation

Let's write a number recognition program in exercise 36 using Keras. However, a deep learning model consisting of two layers is used.

41.2. Programming-the whole program

These codes I improved from Exercise 36.

I do Number recognition program using Keras (deep learning)

In this practice, I use and update some functions:

1. Using TensorFlow backend
2. Using MNIST dataset
3. Build models, layers using Keras library
4. Write function to convert physical image to array vector
5. Run model and predict function to recognize the number
6. Use pyplot to show processing and predict result
7. Do 10 test cases to recognize number (0 → 9)

Link for images to test:

https://github.com/thanhtd32/advancedpython_part1/tree/main/data_exercise37

All codes:

```
#Coder: Tran Duy Thanh
#Email: thanhtd@uel.edu.vn
#Phone: 0987773061
#Blog for self-study: https://duythanhcse.wordpress.com
#Facebook for solving coding problem: https://www.facebook.com/groups/communityuni
#Description:
#These codes I improved from Exercise 36
# Number recognition program using Keras (deep learning)
# In this practice, I use and update some functions:
# 1. Using TensorFlow backend
# 2. Using MNIST dataset
# 3. Build model using Keras
# 4. Write function to convert physical image to array vector
# 5. Run model and predict function to regcognize the number
# 6. Use pyplot to show processing and predict result
# 7. Test some cases to recognize number
#Step 1. Import some necessarily library
from PIL import Image
import numpy as np
```

```
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.datasets import mnist
import cv2

# Step 2. Load MNIST Dataset
#Load data from the MNIST dataset, including 60,000 training sets and 10,000 test sets.
# Then divide the training set to 2 parts: 50,000 for training set and 10,000 data for validation set.
(X_train, y_train), (X_test, y_test) = mnist.load_data()
X_val, y_val = X_train[50000:60000,:], y_train[50000:60000]
X_train, y_train = X_train[:50000,:], y_train[:50000]
#print to see data
print(X_train.shape)

# Step 3. Reshape the data to the correct size required by keras
# Input data for the Convolutional Neural Network model is a 4-dimensional tensor (N, W, H, D),
#in this MNIST dataset is a grayscale image so W = H = 28, D = 1,
#N is the number of images for each training session.
#Because the image data above has the size of (N, 28, 28) ie (N, W, H)
#so need to reshape to N 28 28 * 1 size to match the size required by keras.
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1)
X_val = X_val.reshape(X_val.shape[0], 28, 28, 1)
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1)

# Step 4. One hot encoding label (Y)
#This step converts the one-hot encoding label Y of the example
#image number 5 into a vector [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
Y_train = np_utils.to_categorical(y_train, 10)
Y_val = np_utils.to_categorical(y_val, 10)
Y_test = np_utils.to_categorical(y_test, 10)
#print result to see
print('Original data y', y_train[0])
print('y after one-hot encoding', Y_train[0])

# Step 5. Model definition
#1.Model = Sequential() to tell keras that we will layer layers on top of each other to create models.
# Example input -> CONV -> POOL -> CONV -> POOL -> FLATTEN -> FC -> OUTPUT
#2.In the first layer, We need to specify the input_shape of the image, input_shape = (W, H, D),
# we use grayscale image size (28,28) so input_shape = (28, 28, 1)
#3.When adding a Convolutional Layer, we need to specify the parameters: K (number of layers),
# kernel size (W, H), activation function to use.
# structure: model.add(Conv2D(K, (W, H), activation='function_name_activation'))
#4.When adding a Maxpooling Layer, specify the size of the kernel, model.add(MaxPooling2D(pool_size=(W, H)))
#5.Flatten step from tensor to vector just add flatten layer.
#6.To add Fully Connected Layer (FC) need to specify the number of nodes in the layer and the activation
# function used in the layer, structure: model.add(Dense(number_node_activation='activation_function_name'))
model = Sequential()
# Add Convolutional layer with 32 kernels, kernel size 3*3
# use sigmoid function as activation and specify input_shape for first layer
model.add(Conv2D(32, (3, 3), activation='sigmoid', input_shape=(28, 28, 1)))

# Add Convolutional layer
model.add(Conv2D(32, (3, 3), activation='sigmoid'))
```

```
# Add Max pooling layer
model.add(MaxPooling2D(pool_size=(2, 2)))

# Flatten layer convert from tensor to vector
model.add(Flatten())

# Add Fully Connected layer with 128 nodes and use sigmoid . function
model.add(Dense(128, activation='sigmoid'))

# Output layer with 10 nodes and use softmax function to convert to probability.
model.add(Dense(10, activation='softmax'))

# Step 6. Compile model, specify which loss_function to use, method
# is used to optimize the loss function.
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

# Step 7. Train model with data
H = model.fit(X_train, Y_train, validation_data=(X_val, Y_val),
               batch_size=32, epochs=10, verbose=1)

# Step 8. Plot loss, accuracy of training set and validation set
fig = plt.figure()
numOfEpoch = 10
plt.plot(np.arange(0, numOfEpoch), H.history['loss'], label='training loss')
plt.plot(np.arange(0, numOfEpoch), H.history['val_loss'], label='validation loss')
plt.plot(np.arange(0, numOfEpoch), H.history['accuracy'], label='accuracy')
plt.plot(np.arange(0, numOfEpoch), H.history['val_accuracy'], label='validation accuracy')
plt.title('Accuracy and Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss|Accuracy')
plt.legend()

# Step 9. Model evaluation with test set data
score = model.evaluate(X_test, Y_test, verbose=0)
#print result to see
print(score)
#[0.033531852066516876, 0.9901999831199646]
# We will use the evaluation result of the mode with the test set to make the final result of the model.
# That is, our model predicts digits with 99% accuracy with the MNIST dataset.
# That is, predicting about 100 images will be wrong by 1 image.

# Step 10. Number prediction
# create conv_image_to_data function to convert image file to array 28x28
def conv_image_to_data(filename):
    img_array = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
    img_pil = Image.fromarray(img_array)
    img_28x28 = np.array(img_pil.resize((28, 28), Image.ANTIALIAS))

    img_array = (img_28x28.flatten())
    img_array = img_array.reshape(-1,1).T
    return img_array

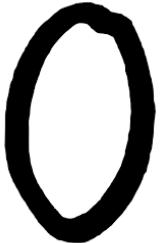
#call conv_image_to_data to test image digit-5.png
data=conv_image_to_data("data_exercise37\digit-1.png")
#show image into chart
plt.imshow(data.reshape(28,28), cmap='Blues')
#call predict function
```

```
y_predict = model.predict(data.reshape(1,28,28,1))
#get number recognition and show into title of Chart
plt.title('Predict number is '+str(np.argmax(y_predict)))
plt.show()
```

41.3. Testing and debugging

Here are some testing cases for the programming:

Here the report for 10 test cases in the table, the details for each test case are next pages:

Test Case	Test Digit	Test Loss	Test Accuracy	Predict
1		~0.0374	~0.9869	5 (correct)
2		~0.0326	~0.9904	7 (correct)
3		~0.0356	~0.9884	2 (correct)
4		~0.0378	~0.990	0(correct)
5		~0.0306	~0.9903	9(correct)

6	1	~ 0.03287	~ 0.9912	1(correct)
7	8	~0.0363	~ 0.989	3 (wrong)
8	3	~ 0.0355	~ 0.9903	3(correct)
9	4	~0.0421	~0.9874	7 (wrong)
10	6	~0.0364	~0.9888	8 (wrong)

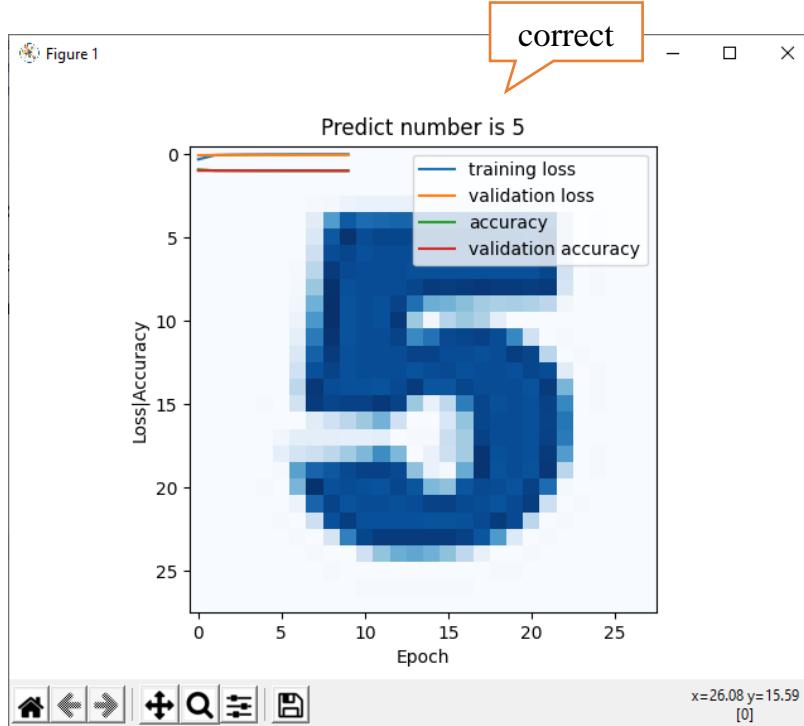
The details for each test case are next pages.

Test case 1:

Digit number =5

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-5.png



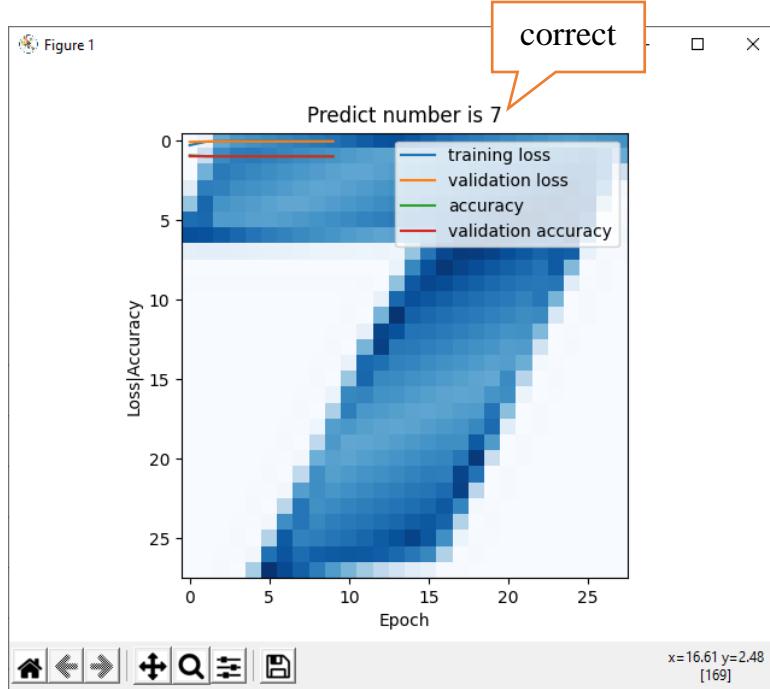
```
2021-06-11 19:22:25.984522: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1a9d28d84a0 initialized for platform Host (this is the host)
2021-06-11 19:22:25.984703: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.2548 - accuracy: 0.9282 - val_loss: 0.0735 - val_accuracy: 0.9792
Epoch 2/10
1563/1563 [=====] - 41s 27ms/step - loss: 0.0572 - accuracy: 0.9836 - val_loss: 0.0585 - val_accuracy: 0.9830
Epoch 3/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0371 - accuracy: 0.9896 - val_loss: 0.0457 - val_accuracy: 0.9859
Epoch 4/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0253 - accuracy: 0.9926 - val_loss: 0.0448 - val_accuracy: 0.9857
Epoch 5/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0161 - accuracy: 0.9959 - val_loss: 0.0432 - val_accuracy: 0.9882
Epoch 6/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0103 - accuracy: 0.9972 - val_loss: 0.0437 - val_accuracy: 0.9882
Epoch 7/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0079 - accuracy: 0.9979 - val_loss: 0.0401 - val_accuracy: 0.9880
Epoch 8/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0052 - accuracy: 0.9989 - val_loss: 0.0422 - val_accuracy: 0.9886
Epoch 9/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0044 - accuracy: 0.9990 - val_loss: 0.0482 - val_accuracy: 0.9870
Epoch 10/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0038 - accuracy: 0.9990 - val_loss: 0.0451 - val_accuracy: 0.9883
[0.037427082657814026, 0.9868999719619751]
```

Test case 2:

Digit number =7

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-7.png

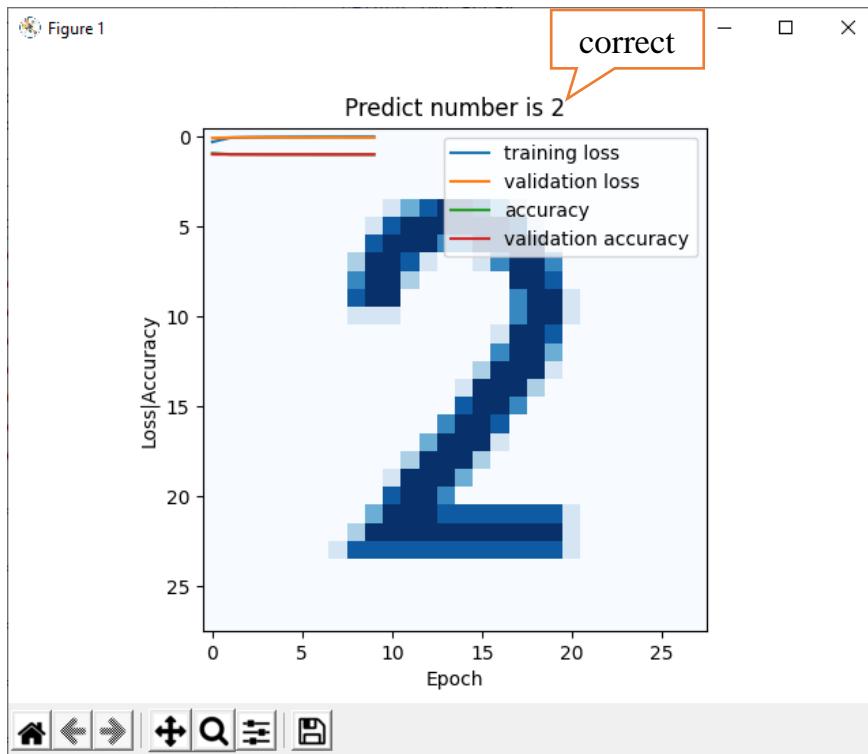


```
2021-06-11 19:34:59.550156: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x265190b5b70 initialized for platform Host (this
2021-06-11 19:34:59.550326: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.2855 - accuracy: 0.9203 - val_loss: 0.0819 - val_accuracy: 0.9789
Epoch 2/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0617 - accuracy: 0.9821 - val_loss: 0.0553 - val_accuracy: 0.9842
Epoch 3/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0383 - accuracy: 0.9886 - val_loss: 0.0480 - val_accuracy: 0.9854
Epoch 4/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0254 - accuracy: 0.9927 - val_loss: 0.0391 - val_accuracy: 0.9889
Epoch 5/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0180 - accuracy: 0.9948 - val_loss: 0.0461 - val_accuracy: 0.9859
Epoch 6/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0122 - accuracy: 0.9969 - val_loss: 0.0435 - val_accuracy: 0.9877
Epoch 7/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0087 - accuracy: 0.9978 - val_loss: 0.0404 - val_accuracy: 0.9890
Epoch 8/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0071 - accuracy: 0.9981 - val_loss: 0.0456 - val_accuracy: 0.9873
Epoch 9/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0043 - accuracy: 0.9990 - val_loss: 0.0437 - val_accuracy: 0.9885
Epoch 10/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0054 - accuracy: 0.9984 - val_loss: 0.0401 - val_accuracy: 0.9903
[0.032649289816617966, 0.9904000163078308]
```

Test case 3:

Digit number =2

Link of this number:

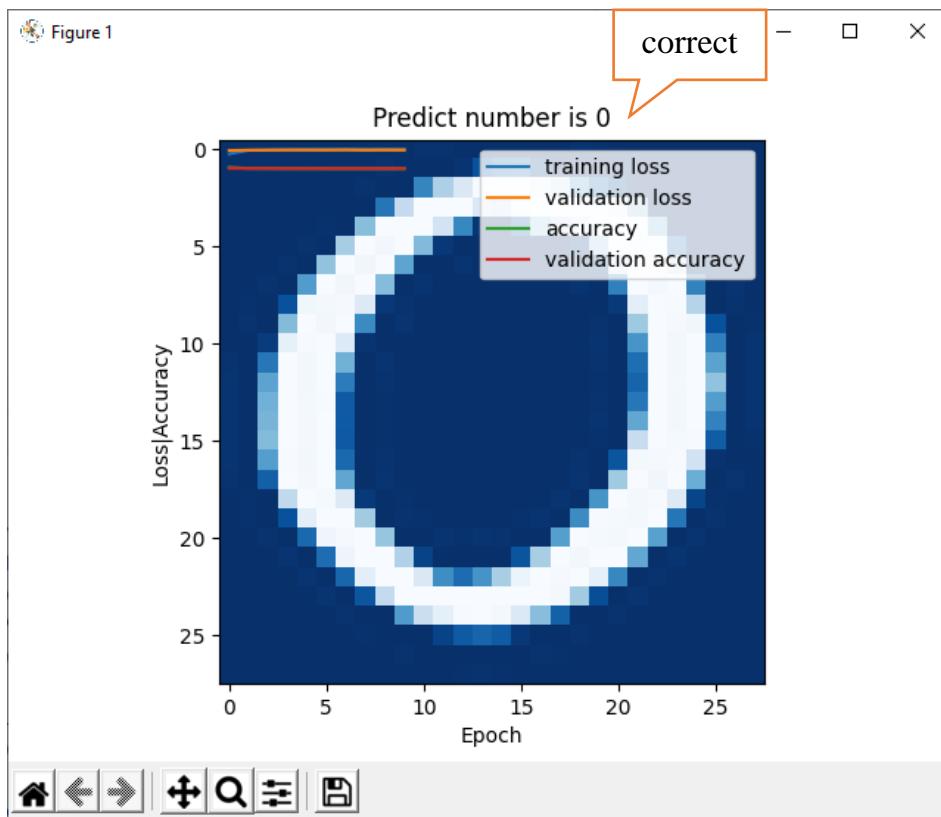
https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-2.png

```
2021-06-11 16:57:09.553910: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.2918 - accuracy: 0.9154 - val_loss: 0.0726 - val_accuracy: 0.9813
Epoch 2/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.0568 - accuracy: 0.9832 - val_loss: 0.0481 - val_accuracy: 0.9862
Epoch 3/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0352 - accuracy: 0.9897 - val_loss: 0.0426 - val_accuracy: 0.9880
Epoch 4/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0221 - accuracy: 0.9939 - val_loss: 0.0397 - val_accuracy: 0.9883
Epoch 5/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0155 - accuracy: 0.9956 - val_loss: 0.0395 - val_accuracy: 0.9881
Epoch 6/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0095 - accuracy: 0.9977 - val_loss: 0.0489 - val_accuracy: 0.9864
Epoch 7/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0069 - accuracy: 0.9982 - val_loss: 0.0425 - val_accuracy: 0.9877
Epoch 8/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0049 - accuracy: 0.9990 - val_loss: 0.0421 - val_accuracy: 0.9893
Epoch 9/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0029 - accuracy: 0.9992 - val_loss: 0.0441 - val_accuracy: 0.9889
Epoch 10/10
1563/1563 [=====] - 40s 25ms/step - loss: 0.0038 - accuracy: 0.9990 - val_loss: 0.0458 - val_accuracy: 0.9887
[0.03563575819134712, 0.9883999824523926]
```

Test case 4:

Digit number =0

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-0.png

```

↑ 2021-06-11 19:45:39.892834: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1625445d430 initialized for platform Host (this
↓ 2021-06-11 19:45:39.893028: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.2463 - accuracy: 0.9308 - val_loss: 0.0644 - val_accuracy: 0.9832
Epoch 2/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0555 - accuracy: 0.9836 - val_loss: 0.0482 - val_accuracy: 0.9859
Epoch 3/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0346 - accuracy: 0.9898 - val_loss: 0.0461 - val_accuracy: 0.9865
Epoch 4/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.0233 - accuracy: 0.9933 - val_loss: 0.0399 - val_accuracy: 0.9881
Epoch 5/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0164 - accuracy: 0.9954 - val_loss: 0.0403 - val_accuracy: 0.9876
Epoch 6/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0106 - accuracy: 0.9972 - val_loss: 0.0443 - val_accuracy: 0.9875
Epoch 7/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0073 - accuracy: 0.9984 - val_loss: 0.0368 - val_accuracy: 0.9895
Epoch 8/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0053 - accuracy: 0.9986 - val_loss: 0.0496 - val_accuracy: 0.9868
Epoch 9/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0038 - accuracy: 0.9991 - val_loss: 0.0419 - val_accuracy: 0.9884
Epoch 10/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0036 - accuracy: 0.9992 - val_loss: 0.0461 - val_accuracy: 0.9877
[0.037846799939870834, 0.989799976348877]

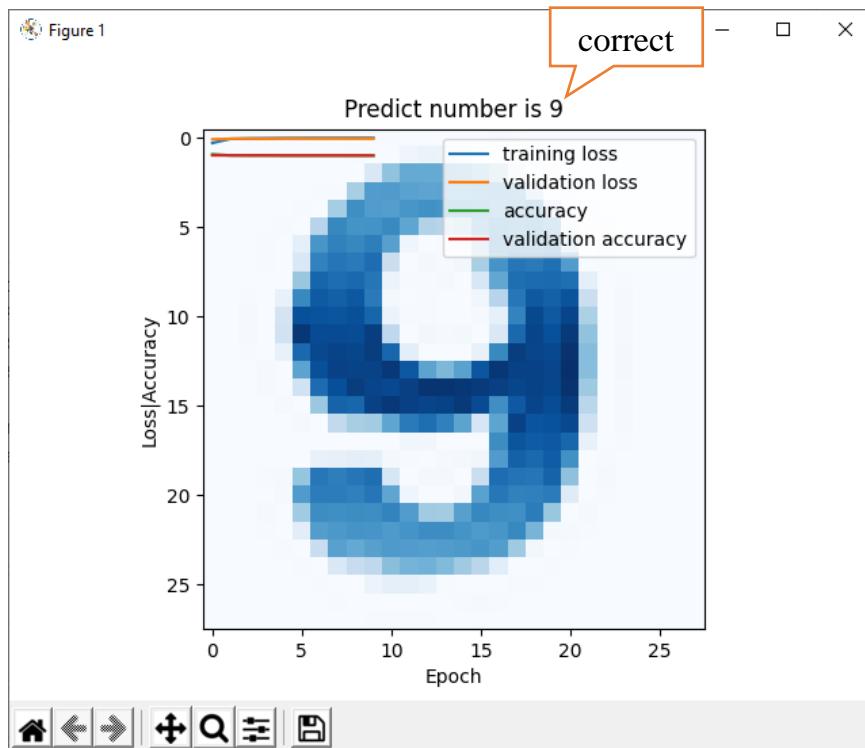
```

Test case 5:

Digit number =9

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-9.png

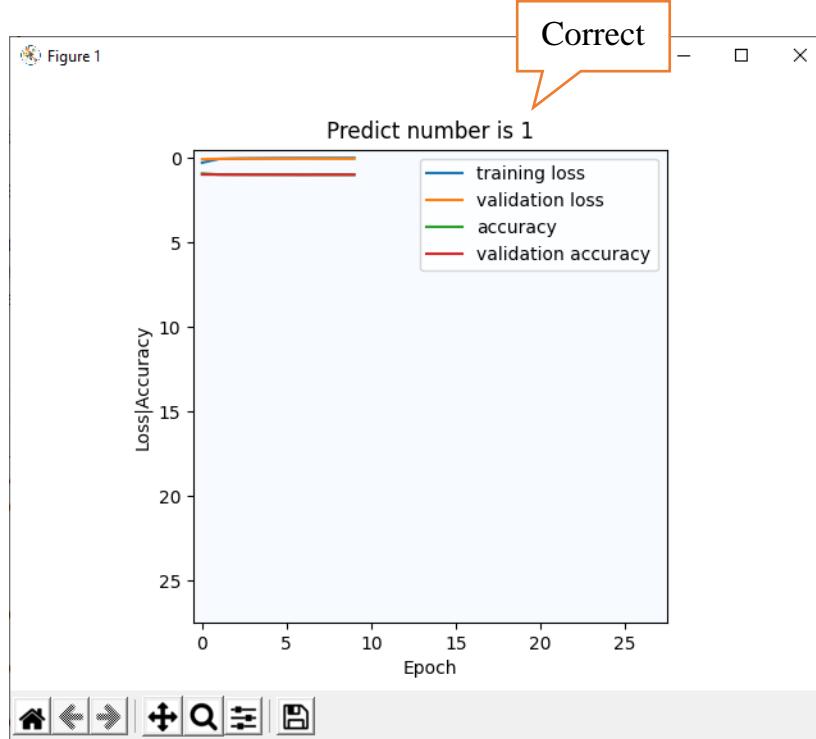


```
2021-06-11 17:15:38.084807: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x2556c9a1070 initialized for platform Host (this
2021-06-11 17:15:38.084964: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.2907 - accuracy: 0.9174 - val_loss: 0.0816 - val_accuracy: 0.9789
Epoch 2/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0586 - accuracy: 0.9834 - val_loss: 0.0547 - val_accuracy: 0.9844
Epoch 3/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0375 - accuracy: 0.9890 - val_loss: 0.0497 - val_accuracy: 0.9858
Epoch 4/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0255 - accuracy: 0.9927 - val_loss: 0.0578 - val_accuracy: 0.9818
Epoch 5/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0170 - accuracy: 0.9954 - val_loss: 0.0425 - val_accuracy: 0.9873
Epoch 6/10
1563/1563 [=====] - 37s 24ms/step - loss: 0.0119 - accuracy: 0.9968 - val_loss: 0.0421 - val_accuracy: 0.9885
Epoch 7/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0087 - accuracy: 0.9977 - val_loss: 0.0477 - val_accuracy: 0.9866
Epoch 8/10
1563/1563 [=====] - 37s 24ms/step - loss: 0.0069 - accuracy: 0.9982 - val_loss: 0.0452 - val_accuracy: 0.9882
Epoch 9/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0051 - accuracy: 0.9988 - val_loss: 0.0473 - val_accuracy: 0.9875
Epoch 10/10
1563/1563 [=====] - 38s 24ms/step - loss: 0.0042 - accuracy: 0.9990 - val_loss: 0.0431 - val_accuracy: 0.9890
[0.03064233623445034, 0.9902999997138977]
```

Test case 6:

Digit number =1

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-1.png

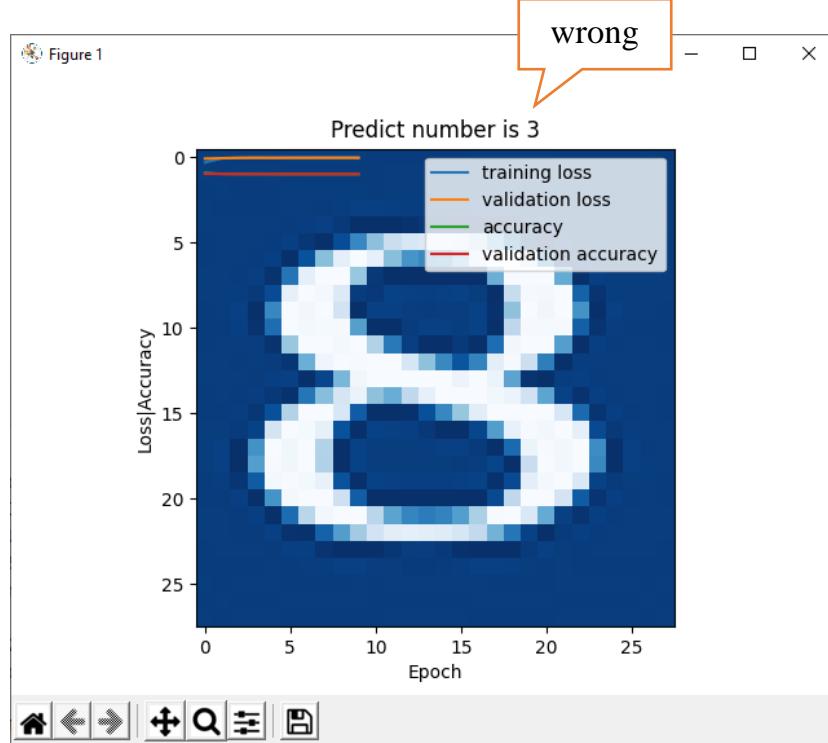
```
2021-06-11 23:06:02.072459: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x153a6962c70 initialized for platform Host (th
2021-06-11 23:06:02.072655: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 50s 32ms/step - loss: 0.2831 - accuracy: 0.9206 - val_loss: 0.0779 - val_accuracy: 0.9795
Epoch 2/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0603 - accuracy: 0.9833 - val_loss: 0.0561 - val_accuracy: 0.9836
Epoch 3/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0388 - accuracy: 0.9886 - val_loss: 0.0479 - val_accuracy: 0.9859
Epoch 4/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0265 - accuracy: 0.9927 - val_loss: 0.0453 - val_accuracy: 0.9864
Epoch 5/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0184 - accuracy: 0.9947 - val_loss: 0.0438 - val_accuracy: 0.9866
Epoch 6/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0117 - accuracy: 0.9971 - val_loss: 0.0451 - val_accuracy: 0.9871
Epoch 7/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0071 - accuracy: 0.9985 - val_loss: 0.0393 - val_accuracy: 0.9897
Epoch 8/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0063 - accuracy: 0.9985 - val_loss: 0.0452 - val_accuracy: 0.9874
Epoch 9/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0048 - accuracy: 0.9986 - val_loss: 0.0462 - val_accuracy: 0.9872
Epoch 10/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0028 - accuracy: 0.9994 - val_loss: 0.0454 - val_accuracy: 0.9888
[0.03286777809262276, 0.9912999868392944]
```

Test case 7:

Digit number =8

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-8.png



```

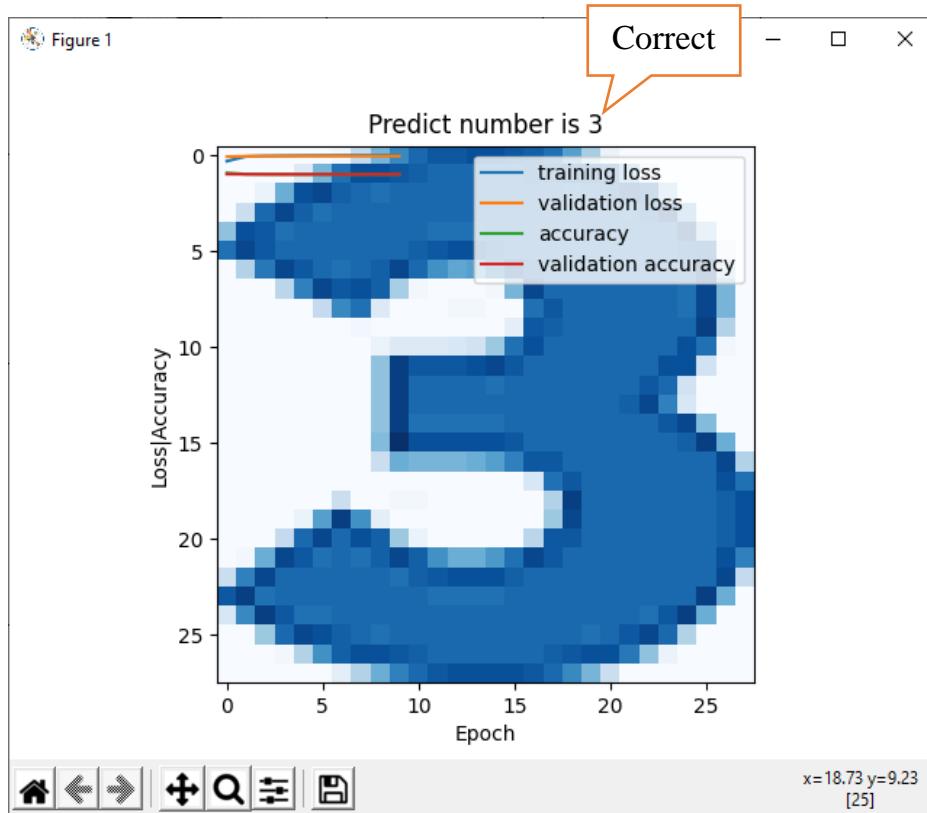
2021-06-11 19:56:57.051425: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x235f3bf2e40 initialized for platform Host (this
2021-06-11 19:56:57.051651: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.3109 - accuracy: 0.9158 - val_loss: 0.0859 - val_accuracy: 0.9763
Epoch 2/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0627 - accuracy: 0.9820 - val_loss: 0.0613 - val_accuracy: 0.9833
Epoch 3/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0385 - accuracy: 0.9890 - val_loss: 0.0446 - val_accuracy: 0.9877
Epoch 4/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0256 - accuracy: 0.9930 - val_loss: 0.0396 - val_accuracy: 0.9888
Epoch 5/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0176 - accuracy: 0.9950 - val_loss: 0.0463 - val_accuracy: 0.9875
Epoch 6/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0122 - accuracy: 0.9967 - val_loss: 0.0437 - val_accuracy: 0.9868
Epoch 7/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0085 - accuracy: 0.9976 - val_loss: 0.0449 - val_accuracy: 0.9875
Epoch 8/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0066 - accuracy: 0.9984 - val_loss: 0.0429 - val_accuracy: 0.9890
Epoch 9/10
1563/1563 [=====] - 40s 26ms/step - loss: 0.0047 - accuracy: 0.9989 - val_loss: 0.0445 - val_accuracy: 0.9879
Epoch 10/10
1563/1563 [=====] - 41s 26ms/step - loss: 0.0042 - accuracy: 0.9990 - val_loss: 0.0473 - val_accuracy: 0.9887
[0.03633527085185051, 0.9886999726295471]

```

Test case 8:

Digit number =3

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-3.png

```

T 2021-06-11 20:29:37.757361: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x259b206a750 initialized for platform Host (this
↓ 2021-06-11 20:29:37.757537: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.3033 - accuracy: 0.9162 - val_loss: 0.0708 - val_accuracy: 0.9803
Epoch 2/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0564 - accuracy: 0.9842 - val_loss: 0.0491 - val_accuracy: 0.9862
Epoch 3/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0345 - accuracy: 0.9899 - val_loss: 0.0410 - val_accuracy: 0.9886
Epoch 4/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0231 - accuracy: 0.9934 - val_loss: 0.0367 - val_accuracy: 0.9889
Epoch 5/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0149 - accuracy: 0.9959 - val_loss: 0.0374 - val_accuracy: 0.9894
Epoch 6/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.0104 - accuracy: 0.9973 - val_loss: 0.0386 - val_accuracy: 0.9872
Epoch 7/10
1563/1563 [=====] - 45s 29ms/step - loss: 0.0070 - accuracy: 0.9982 - val_loss: 0.0363 - val_accuracy: 0.9897
Epoch 8/10
1563/1563 [=====] - 46s 29ms/step - loss: 0.0046 - accuracy: 0.9990 - val_loss: 0.0390 - val_accuracy: 0.9899
Epoch 9/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0034 - accuracy: 0.9994 - val_loss: 0.0413 - val_accuracy: 0.9894
Epoch 10/10
1563/1563 [=====] - 43s 27ms/step - loss: 0.0029 - accuracy: 0.9994 - val_loss: 0.0373 - val_accuracy: 0.9905
[0.035501670092344284, 0.9902999997138977]

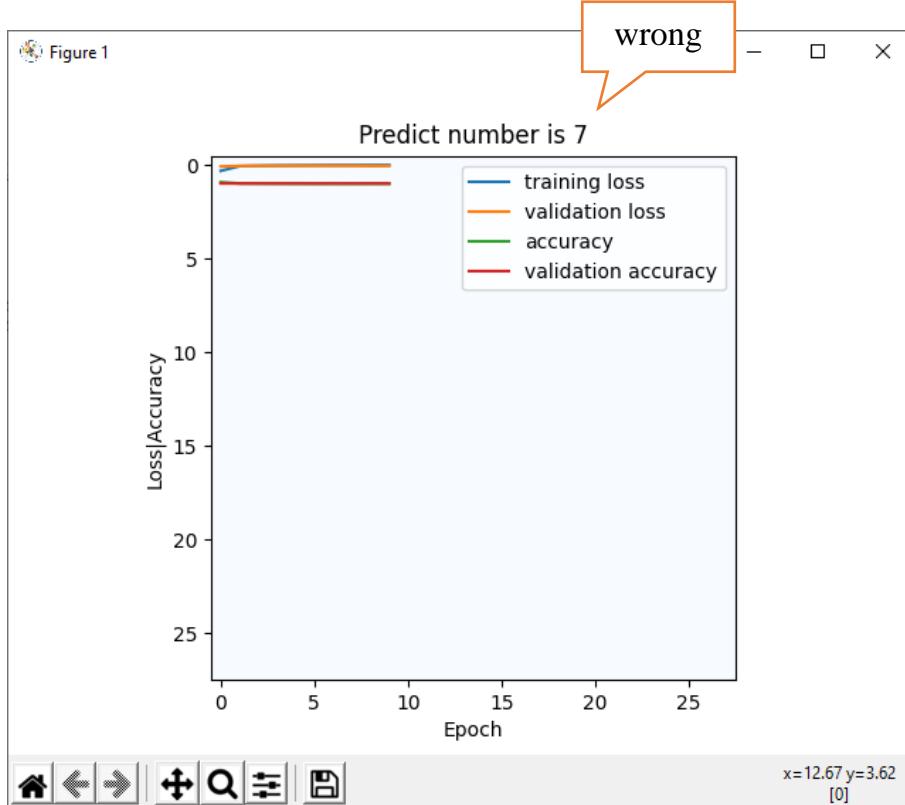
```

Test case 9:

Digit number =4

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-4.png



```

2021-06-11 20:39:23.601558: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x2865a44f7e0 initialized for platform Host (ti
2021-06-11 20:39:23.601734: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Epoch 1/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.3123 - accuracy: 0.9124 - val_loss: 0.0758 - val_accuracy: 0.9809
Epoch 2/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0595 - accuracy: 0.9829 - val_loss: 0.0473 - val_accuracy: 0.9863
Epoch 3/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0376 - accuracy: 0.9892 - val_loss: 0.0443 - val_accuracy: 0.9863
Epoch 4/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0264 - accuracy: 0.9925 - val_loss: 0.0404 - val_accuracy: 0.9878
Epoch 5/10
1563/1563 [=====] - 43s 28ms/step - loss: 0.0172 - accuracy: 0.9954 - val_loss: 0.0413 - val_accuracy: 0.9879
Epoch 6/10
1563/1563 [=====] - 44s 28ms/step - loss: 0.0122 - accuracy: 0.9966 - val_loss: 0.0380 - val_accuracy: 0.9899
Epoch 7/10
1563/1563 [=====] - 42s 27ms/step - loss: 0.0076 - accuracy: 0.9979 - val_loss: 0.0413 - val_accuracy: 0.9876
Epoch 8/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0051 - accuracy: 0.9988 - val_loss: 0.0409 - val_accuracy: 0.9890
Epoch 9/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0044 - accuracy: 0.9990 - val_loss: 0.0539 - val_accuracy: 0.9863
Epoch 10/10
1563/1563 [=====] - 39s 25ms/step - loss: 0.0030 - accuracy: 0.9994 - val_loss: 0.0466 - val_accuracy: 0.9873
[0.04210694879293442, 0.9873999953269958]

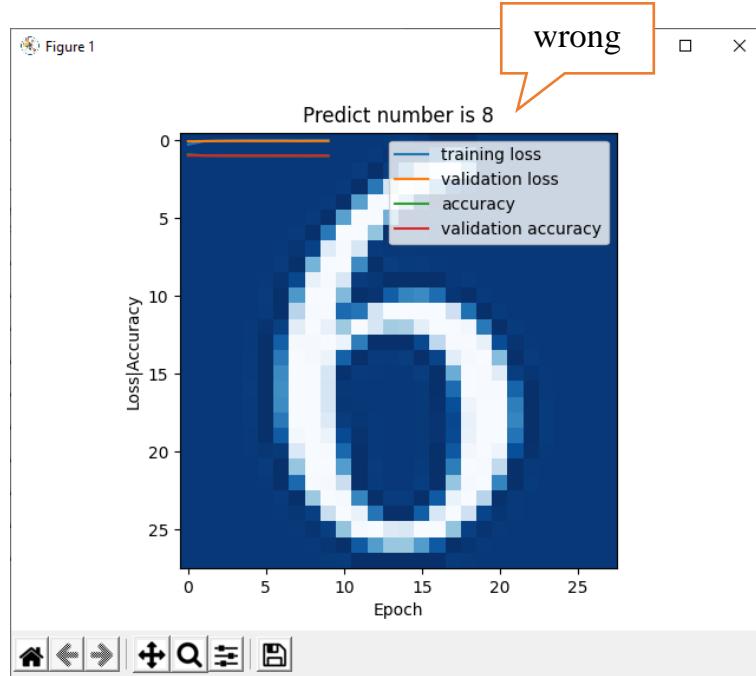
```

Test case 10:

Digit number =6

Link of this number:

https://github.com/thanhtd32/advancedpython_part1/blob/main/data_exercise37/digit-6.png



```

T | 2021-06-11 20:48:09.737972: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x18882d57da0 initialized for platform Host (t
D | 2021-06-11 20:48:09.738141: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
P | Epoch 1/10
E | 1563/1563 [=====] - 45s 29ms/step - loss: 0.2786 - accuracy: 0.9189 - val_loss: 0.0710 - val_accuracy: 0.9813
E | Epoch 2/10
E | 1563/1563 [=====] - 43s 28ms/step - loss: 0.0595 - accuracy: 0.9825 - val_loss: 0.0581 - val_accuracy: 0.9843
E | Epoch 3/10
E | 1563/1563 [=====] - 45s 29ms/step - loss: 0.0377 - accuracy: 0.9892 - val_loss: 0.0457 - val_accuracy: 0.9857
E | Epoch 4/10
E | 1563/1563 [=====] - 44s 28ms/step - loss: 0.0247 - accuracy: 0.9929 - val_loss: 0.0425 - val_accuracy: 0.9878
E | Epoch 5/10
E | 1563/1563 [=====] - 43s 27ms/step - loss: 0.0166 - accuracy: 0.9954 - val_loss: 0.0420 - val_accuracy: 0.9878
E | Epoch 6/10
E | 1563/1563 [=====] - 46s 29ms/step - loss: 0.0113 - accuracy: 0.9970 - val_loss: 0.0484 - val_accuracy: 0.9868
E | Epoch 7/10
E | 1563/1563 [=====] - 42s 27ms/step - loss: 0.0073 - accuracy: 0.9983 - val_loss: 0.0428 - val_accuracy: 0.9878
E | Epoch 8/10
E | 1563/1563 [=====] - 43s 27ms/step - loss: 0.0057 - accuracy: 0.9986 - val_loss: 0.0510 - val_accuracy: 0.9870
E | Epoch 9/10
E | 1563/1563 [=====] - 46s 29ms/step - loss: 0.0041 - accuracy: 0.9992 - val_loss: 0.0458 - val_accuracy: 0.9875
E | Epoch 10/10
E | 1563/1563 [=====] - 41s 26ms/step - loss: 0.0033 - accuracy: 0.9993 - val_loss: 0.0458 - val_accuracy: 0.9889
[0.03649739921092987, 0.9887999892234802]

```

I put my code in GitHub, here is the link:

https://github.com/thanhtd32/advancedpython_part1/blob/main/Exercise37.py