



Homework # 10

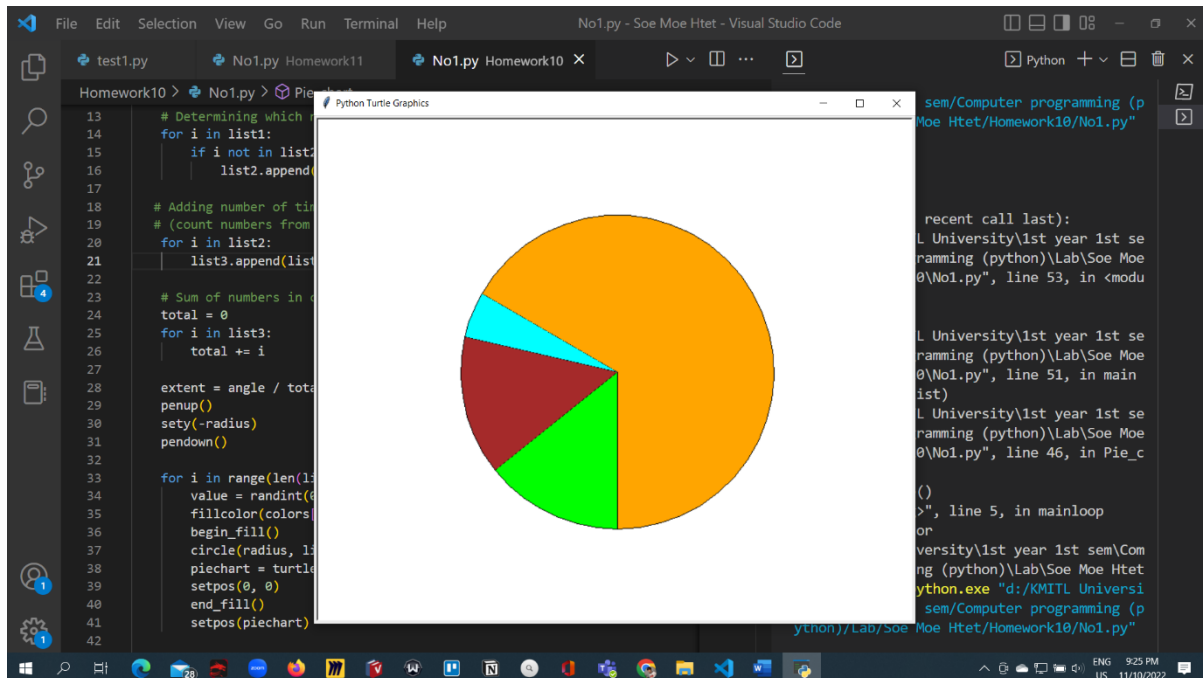
**01286121 Computer Programming
Software Engineering Program,
Department of Computer Engineering,
School of Engineering, KMITL**

By

65011693 Soe Moe Htet
(Nickname – Stephen)

No1

Result:



Code:

```
from random import randint

from turtle import *
import turtle
radius = 200
angle = 360

def Pie_chart(list1):
    list2 = [] # duplicate number list
    list3 = [] # count list
    colors = ["red", "blue", "cyan", "brown", "orange", "green", "purple", "lime"]

    # Determining which numbers are duplicating
    for i in list1:
        if i not in list2:
            list2.append(i)

    # Adding number of times into list3
    # (count numbers from list 2 in list1)
    for i in list2:
        list3.append(list1.count(i))

    # Sum of numbers in count list
    total = 0
    for i in list3:
        total += i

    extent = angle / total
    penup()
    sety(-radius)
```

```
pendown()

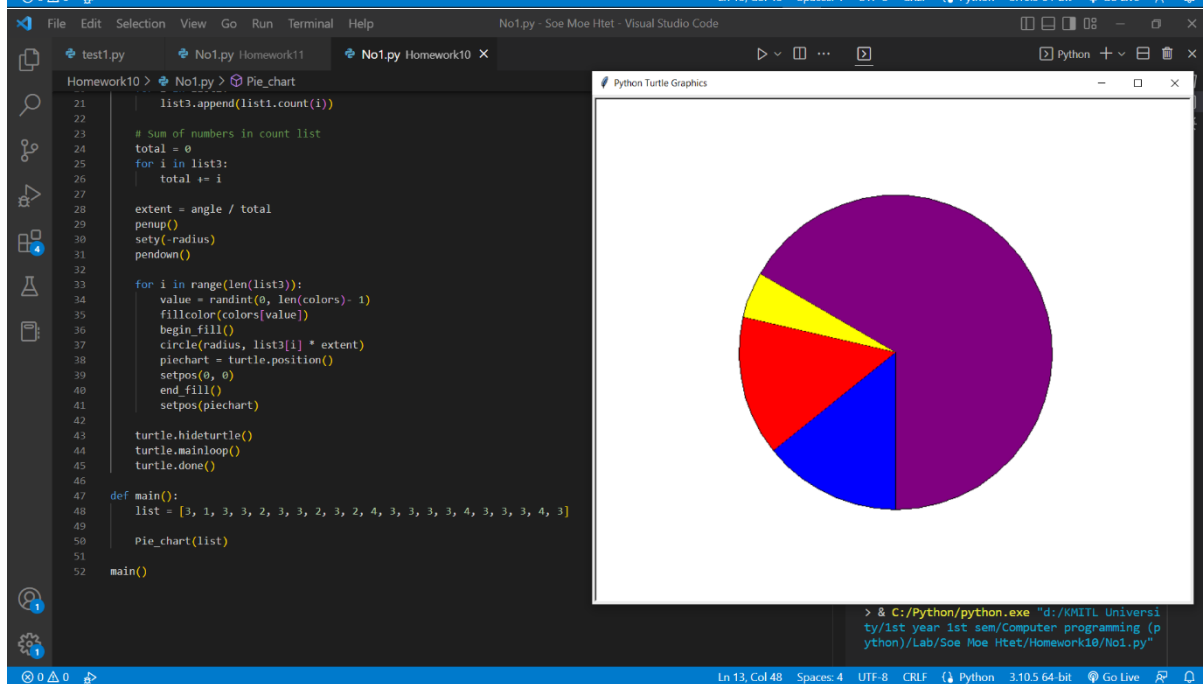
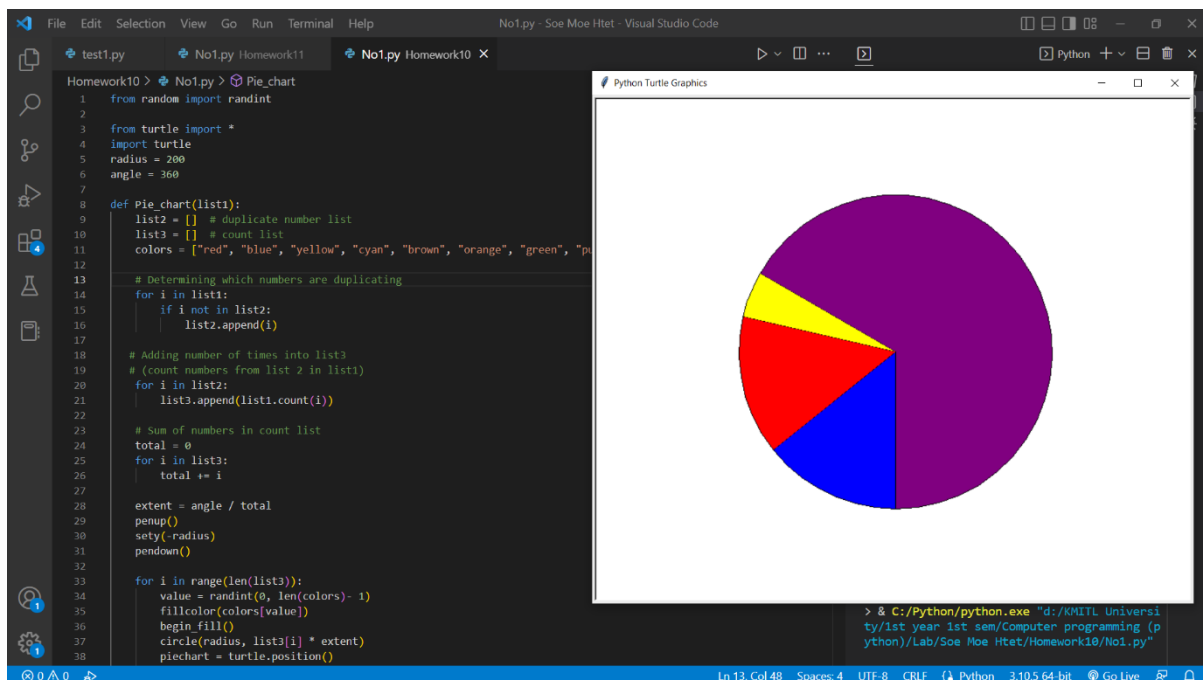
for i in range(len(list3)):
    value = randint(0, len(colors)- 1)
    fillcolor(colors[value])
    begin_fill()
    circle(radius, list3[i] * extent)
    piechart = turtle.position()
    setpos(0, 0)
    end_fill()
    setpos(piechart)

turtle.hideturtle()
turtle.mainloop()
turtle.done()

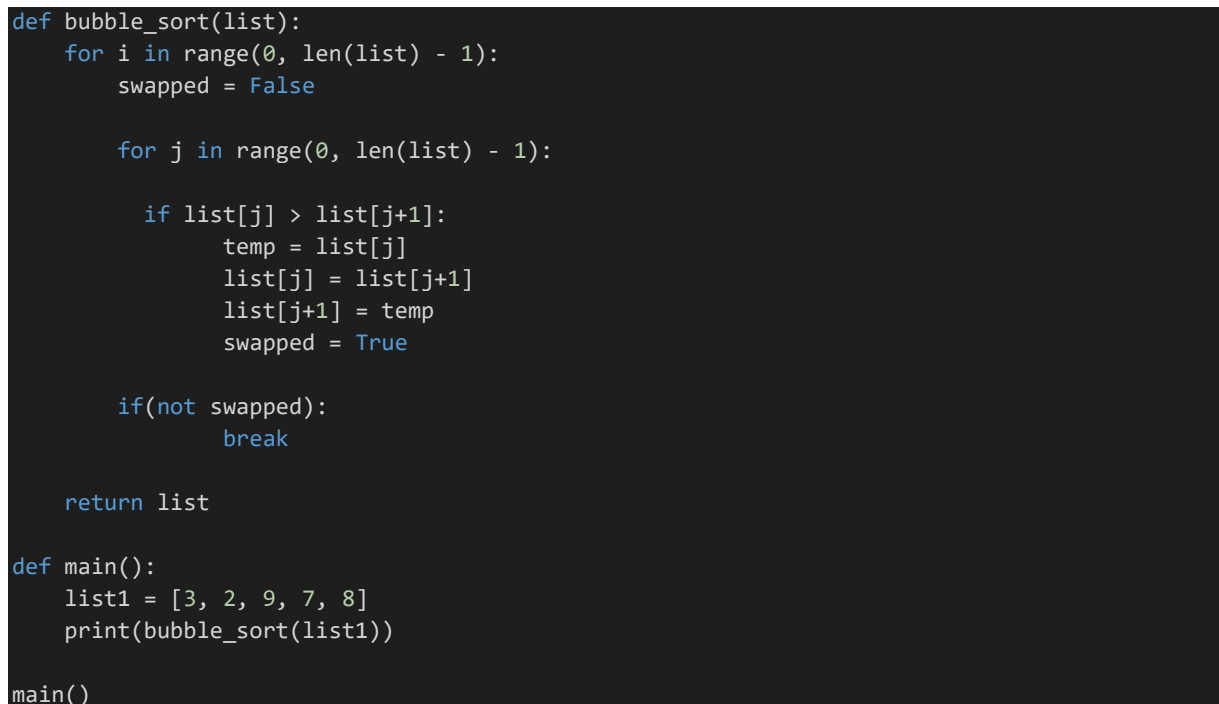
def main():
    list = [3, 1, 3, 3, 2, 3, 3, 2, 3, 2, 4, 3, 3, 3, 3, 4, 3, 3, 3, 4, 3]

    Pie_chart(list)

main()
```

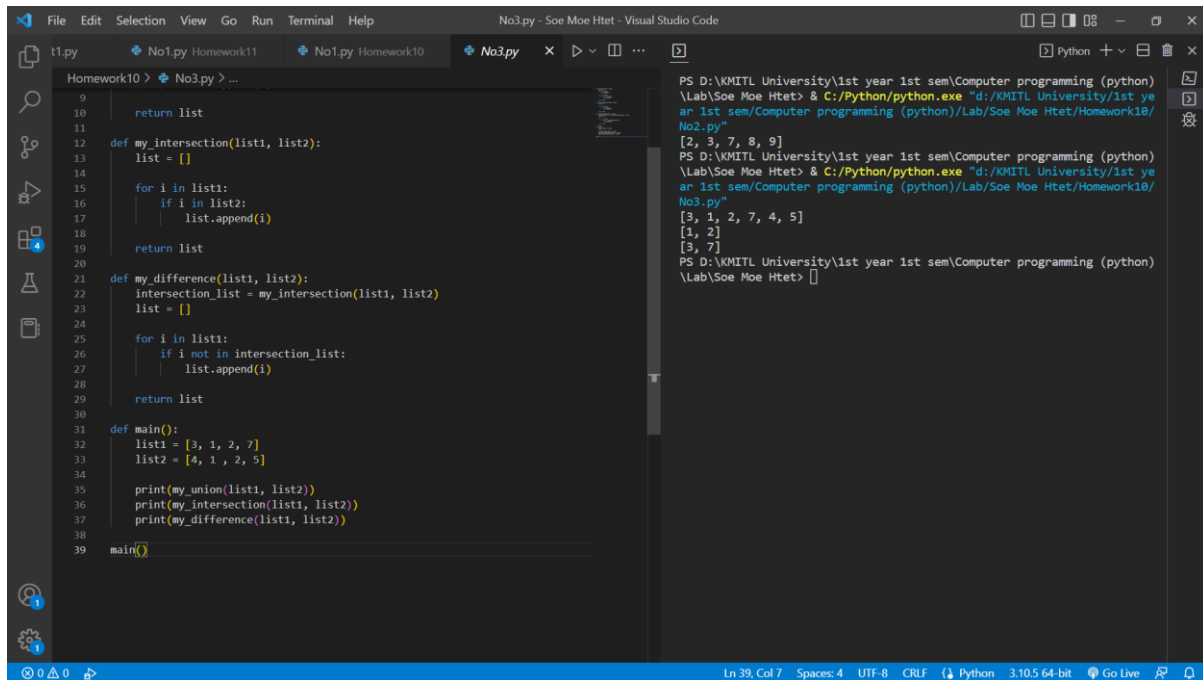


Code:



No.3

Result:



The screenshot shows a Visual Studio Code window with a file named 'No3.py' open. The code defines three functions: `my_union`, `my_intersection`, and `my_difference`, along with a `main` function. The `main` function initializes two lists, `list1 = [3, 1, 2, 7]` and `list2 = [4, 1, 2, 5]`, and prints the results of `my_union(list1, list2)`, `my_intersection(list1, list2)`, and `my_difference(list1, list2)`. The terminal on the right shows the execution of the script, displaying the output: `[2, 3, 7, 8, 9]`, `[1, 2]`, and `[3, 7]`.

```
def my_union(list1, list2):
    list = []
    for i in list1:
        list.append(i)

    for i in list2:
        if i not in list:
            list.append(i)

    return list

def my_intersection(list1, list2):
    list = []

    for i in list1:
        if i in list2:
            list.append(i)

    return list

def my_difference(list1, list2):
    intersection_list = my_intersection(list1, list2)
    list = []

    for i in list1:
        if i not in intersection_list:
            list.append(i)

    return list

def main():
    list1 = [3, 1, 2, 7]
    list2 = [4, 1, 2, 5]

    print(my_union(list1, list2))
    print(my_intersection(list1, list2))
    print(my_difference(list1, list2))

main()
```

```
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No2.py"
[2, 3, 7, 8, 9]
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No3.py"
[3, 1, 2, 7, 4, 5]
[1, 2]
[3, 7]
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet>
```

Code:

```
def my_union(list1, list2):
    list = []
    for i in list1:
        list.append(i)

    for i in list2:
        if i not in list:
            list.append(i)

    return list

def my_intersection(list1, list2):
    list = []

    for i in list1:
        if i in list2:
            list.append(i)

    return list

def my_difference(list1, list2):
    intersection_list = my_intersection(list1, list2)
    list = []

    for i in list1:
        if i not in intersection_list:
            list.append(i)

    return list
```

```
def main():
    list1 = [3, 1, 2, 7]
    list2 = [4, 1, 2, 5]

    print(my_union(list1, list2))
    print(my_intersection(list1, list2))
    print(my_difference(list1, list2))

main()
```

No.4

Result

The screenshot shows a Visual Studio Code window with a file named 'No4.py'. The code defines a 'print_table' function and a 'main' function. The 'main' function creates two lists, 'list1' and 'list2', and calls 'print_table' on both. The 'print_table' function prints the header and body of a table. The terminal output shows the execution of the script, displaying the union, intersection, and difference of the two lists, followed by a table with 3 columns: ID, Name, and Surname. The table contains 3 rows of data.

```
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No2.py"
[2, 3, 7, 8, 9]
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No3.py"
[3, 1, 2, 7, 4, 5]
[3, 2]
[3, 7]
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\> & C:/Python/python.exe "d:/KMITL University/1st year 1st sem/Comp
uter programming (python)/Lab/Soe Moe Htet/Homework10/No4.py"
X      Y
0      0
10     10
200    200

ID      Name      Surname
001     Guido    van Rossum
002     Donald    Knuth
003     Gordon    Moore

PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet>
```

Code:

```
def print_table(list):
    Header = []
    Body = []

    Header = list[0]
    Body = list[1:]

    for i in range(len(Header)):
        print(f"{Header[i] :<10}", end = "")
    print()

    i = 0
    while(i != len(Body)):
        j = 0
        while(j != len(Body[i])):
            print(f"{Body[i][j] :<10}", end = "")
            j += 1
```

```

        print()
        i += 1
    print()

def main():
    list1 = [
        ["X", "Y"],
        [0, 0],
        [10, 10],
        [200, 200]
    ]

    list2 = [
        ["ID", "Name", "Surname"],
        ["001", "Guido", "van Rossum"],
        ["002", "Donald", "Knuth"],
        ["003", "Gordon", "Moore"]
    ]
    print_table(list1)
    print_table(list2)
main()

```

No.5

Result:

The screenshot shows the Visual Studio Code interface with a file named 'No5.py' open. The code in the editor defines a function 'isAnagram' and a 'main' function. The 'isAnagram' function takes two strings and returns a boolean. The 'main' function calls 'isAnagram' with 'listen' and 'silent', and 'hi' and 'hello'. The output window on the right shows the execution results, including the command prompt path, the command executed, and the output of the program.

```

No5.py
def isAnagram(String1, String2):
    list1, list2 = [], []

    for i in String1:
        list1.append(i)

    for i in String2:
        list2.append(i)

    isAnagram = True
    for i in list1:
        if i in list2:
            continue
        else:
            isAnagram = False

    return isAnagram

def main():
    print(isAnagram("listen", "silent"))
    print(isAnagram("hi", "hello"))

main()

```

```

PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No5.py"
True
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet> & C:/Python/python.exe "d:/KMITL University/1st ye
ar 1st sem/Computer programming (python)/Lab/Soe Moe Htet/Homework10/
No5.py"
True
False
PS D:\KMITL University\1st year 1st sem\Computer programming (python)
\Lab\Soe Moe Htet>

```

Code:

```

def isAnagram(String1, String2):
    list1, list2 = [], []

    for i in String1:
        list1.append(i)

    for i in String2:

```



```
        list2.append(i)

    isAnagram = True
    for i in list1:
        if i in list2:
            continue
        else:
            isAnagram = False

    return isAnagram

def main():
    print(isAnagram("listen", "silent"))
    print(isAnagram("hi", "hello"))

main()
```