Homework #10

01286121 Computer Programming

Software Engineering Program,

Department of Computer Engineering,

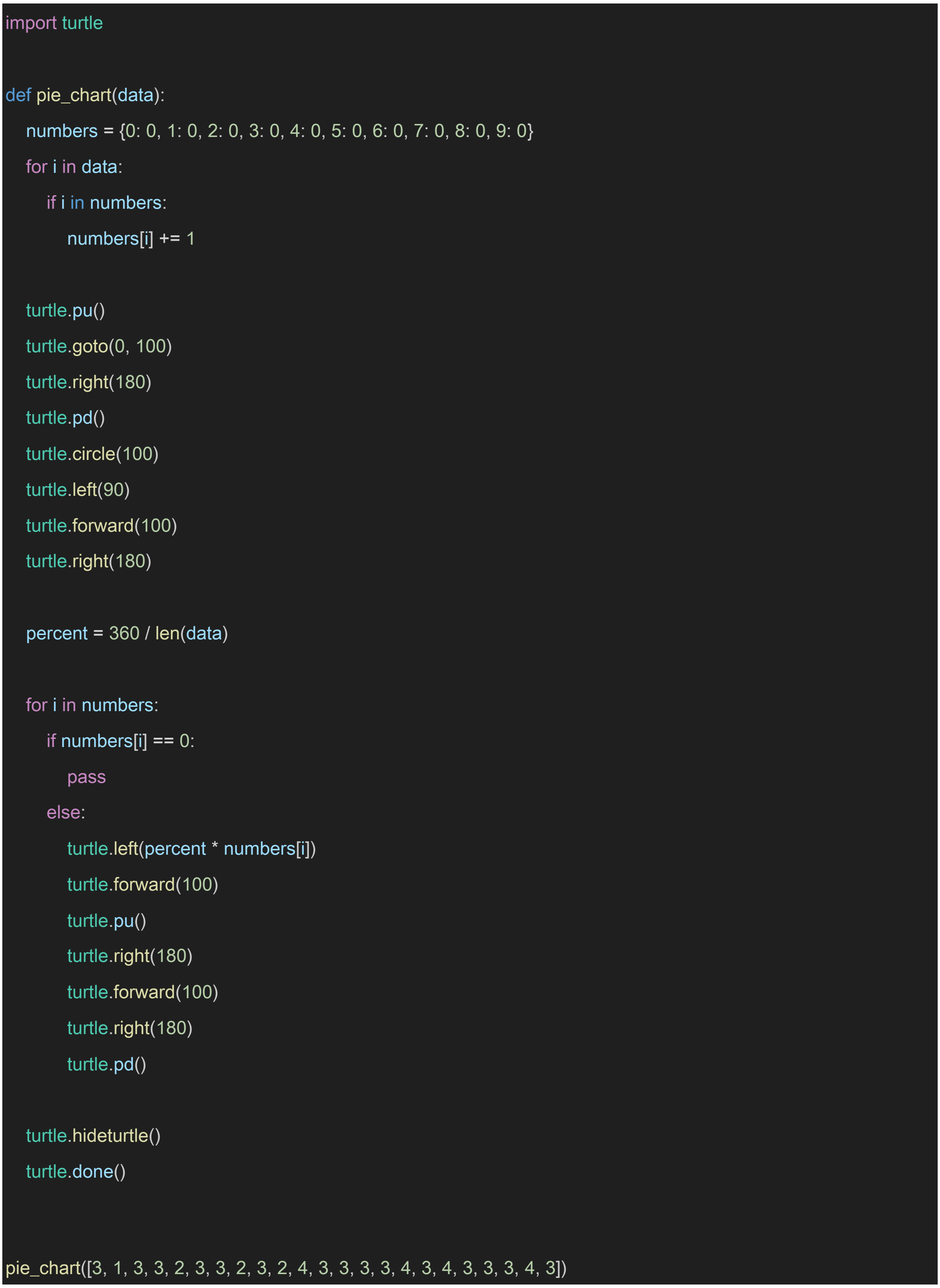
School of Engineering, KMITL

By

68011278 Ananda Stallard

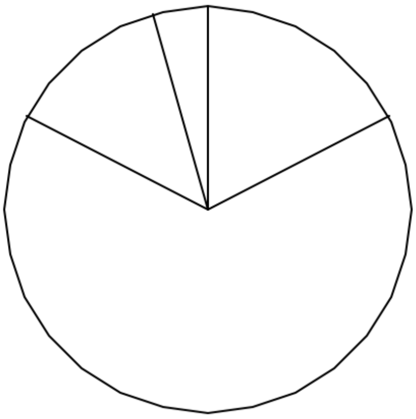
1.

Code:



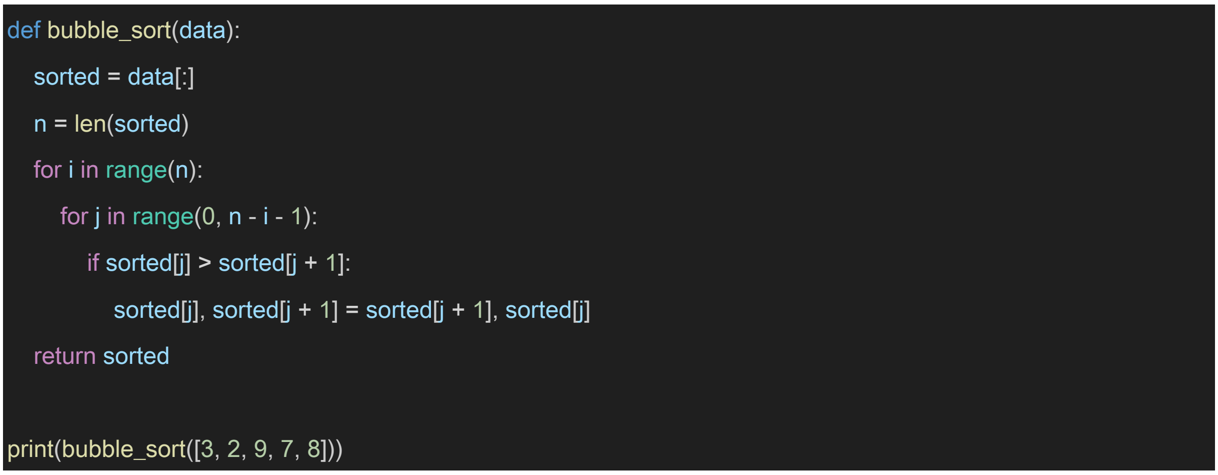
```
import turtle

def pie_chart(data):
    numbers = {0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0}
    for i in data:
        if i in numbers:
            numbers[i] += 1

    turtle.pu()
    turtle.goto(0, 100)
    turtle.right(180)
    turtle.pd()
    turtle.circle(100)
    turtle.left(90)
    turtle.forward(100)
    turtle.right(180)

    percent = 360 / len(data)

    for i in numbers:
        if numbers[i] == 0:
            pass
        else:
            turtle.left(percent * numbers[i])
            turtle.forward(100)
            turtle.pu()
            turtle.right(180)
            turtle.forward(100)
            turtle.right(180)
            turtle.pd()

    turtle.hideturtle()
    turtle.done()


pie_chart([3, 1, 3, 3, 2, 3, 3, 2, 3, 2, 4, 3, 3, 3, 3, 4, 3, 4, 3, 3, 3, 4, 3])
```

Result:



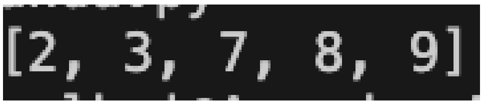
2.

Code:



```
def bubble_sort(data):
    sorted = data[:]
    n = len(sorted)
    for i in range(n):
        for j in range(0, n - i - 1):
            if sorted[j] > sorted[j + 1]:
                sorted[j], sorted[j + 1] = sorted[j + 1], sorted[j]
    return sorted


print(bubble_sort([3, 2, 9, 7, 8]))
```

Result:



```
[2, 3, 7, 8, 9]
```

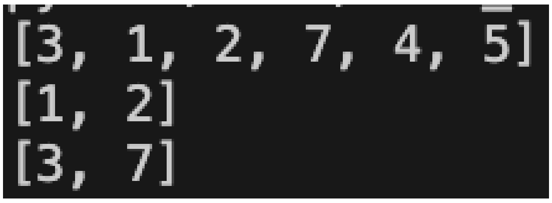
3.

Code:

```
def my_union(list1, list2):
    new_list = []
    for i in list1 + list2:
        if i not in new_list:
            new_list.append(i)

    return new_list

def my_intersection(list1, list2):
    new_list = []
    for i in list1:
        if i in list2 and i not in new_list:
            new_list.append(i)
    return new_list

def my_difference(list1, list2):
    new_list = []
    for i in list1:
        if i not in list2:
            new_list.append(i)

    return new_list
```

```
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))
```

Result:



```
[3, 1, 2, 7, 4, 5]
[1, 2]
[3, 7]
```

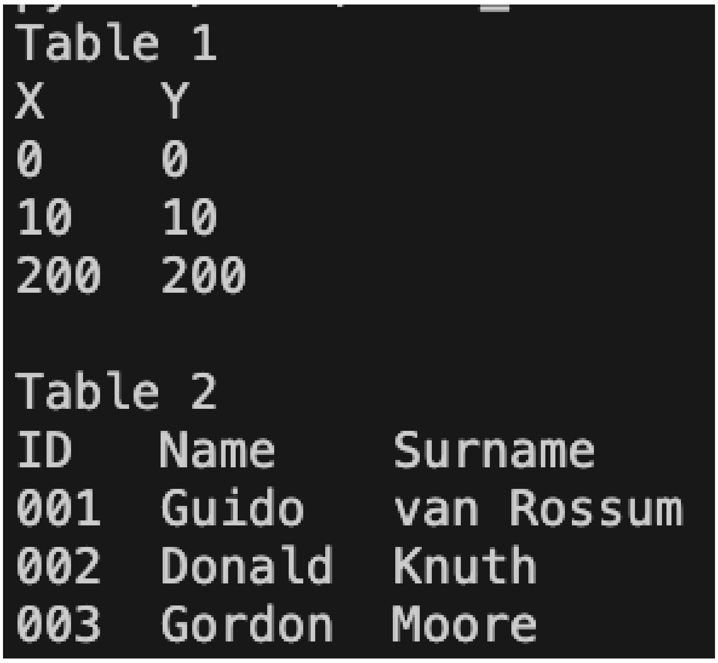
4.

Code:

```
def print_table(table):
    #Checks the longest of each column and returns as a list
    col_widths = [max(len(str(row[i])) for row in table) for i in range(len(table[0]))]

    for row in table:
        line = ""
        for i, cell in enumerate(row):
            line += str(cell).ljust(col_widths[i] + 2)
        print(line)

print("Table 1")

print_table([["X", "Y"], [0, 0], [10, 10], [200, 200]])
```

```
print("\nTable 2")

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

Result:



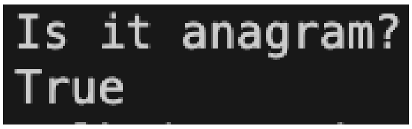
```
Table 1
X    Y
0    0
10   10
200  200

Table 2
ID   Name    Surname
001  Guido   van Rossum
002  Donald  Knuth
003  Gordon  Moore
```

5.

Code:

```
def isAnagram(String1, String2):
    char_count = {'a': 0, 'b': 0, 'c': 0, 'd': 0, 'e': 0, 'f': 0, 'g': 0, 'h': 0, 'i': 0, 'j': 0, 'k': 0, 'l': 0,
                  'm': 0, 'n': 0, 'o': 0, 'p': 0, 'q': 0, 'r': 0, 's': 0, 't': 0, 'u': 0, 'v': 0, 'w': 0, 'x': 0, 'y': 0, 'z': 0}

    count_1 = char_count.copy()
    count_2 = char_count.copy()

    for i in String1.lower():
        count_1[i] += 1
    for i in String2.lower():
        count_2[i] += 1

    if count_1 == count_2:
        return True
    else:
        return False

print(f"Is it anagram?\n{isAnagram("silent", "listen")}")
```

Result:



```
Is it anagram?
True
```