Lab2

September 10, 2024

1 Lab 2

Labs in general are for you to solve short programming challenges in class. In contrast, homework assignments will involve more challenging and lengthy problems.

Feel free to ask the TAs for help if there is anything you do not understand. The TAs will go through suggested solutions in the last 15 minutes of the lab - typically by solving them in a live demo. Your midterm exams will be like this, so it is highly beneficial for you to attend these labs.

- You are not to use any 3rd party library such as numpy for this lab
- Do this lab without using the web to search for solutions

Please remember to submit the lab on Gradescope within 2 hours after the lab.

1.0.1 A few points to review

1. Collections (list, tuple, set, dictionary)
2. Control flows
   - if statement
   - for and while loop
3. functions

1. (1 point). Write a function `find_min_max(x)` that takes a list of numbers and returns the smallest and largest numbers in a new list.

Print the result given the input list `[0.3, 5, 3, 4, 2, 3, 1]`

[ ]:
```python
def find_min_max(x: list[int]) -> int:
    min_x: int = x[0]
    max_x: int = x[0]
    for n in x:
        if n > max_x:
            max_x = n
        elif n < min_x:
            min_x = n
    return (min_x, max_x)
```

[ ]:
```python
x = [0.3, 5, 3, 4, 2, 3, 1]
print(find_min_max(x))
```

```
(0.3, 5)
```

2. (1 points). Write a program to print all the numbers between 1000 and 2000 which are divisible by 7 but are not a multiple of 5.

[ ]:
```
def cond(x: int) -> bool:
    if x % 7 == 0 and x % 5 != 0:
        return True
    else:
        return False

for x in range(1000, 2001):
    if cond(x):
        print(x)
```

```
1001
1008
1022
1029
1036
1043
1057
1064
1071
1078
1092
1099
1106
1113
1127
1134
1141
1148
1162
1169
1176
1183
1197
1204
1211
1218
1232
1239
1246
1253
1267
1274
1281
1288
1302
```

1309
1316
1323
1337
1344
1351
1358
1372
1379
1386
1393
1407
1414
1421
1428
1442
1449
1456
1463
1477
1484
1491
1498
1512
1519
1526
1533
1547
1554
1561
1568
1582
1589
1596
1603
1617
1624
1631
1638
1652
1659
1666
1673
1687
1694
1701
1708
1722

1729
1736
1743
1757
1764
1771
1778
1792
1799
1806
1813
1827
1834
1841
1848
1862
1869
1876
1883
1897
1904
1911
1918
1932
1939
1946
1953
1967
1974
1981
1988

3. (2 point). Write a function `factorial(x)` that takes a number and prints the factorial of the input number. Print the result given the input number equal to 10.

[ ]:
```
def factorial(x: int) -> int:
    for n in range (1, x):
        x *= n
    print(x)
```

[ ]:
```
factorial(10)
```

3628800

4 (2 point). Given a list of tuple `x=[(3.0,6.9), (4.5, 6.8), (3.4, 5.7), (5.8, 0.1), (4.9, 3.3)]`, use `lambda` to sort a the list of tuples based on the second item in each tuple.

[ ]:
```
x=[(3.0,6.9), (4.5, 6.8), (3.4, 5.7), (5.8, 0.1), (4.9, 3.3)]
```

[ ]:
```
sorted(x, key = lambda y: y[1])
```

[ ]:
```
[(5.8, 0.1), (4.9, 3.3), (3.4, 5.7), (4.5, 6.8), (3.0, 6.9)]
```

5. (2 point). Given a list `x=[1,2,3,4,5]`, use `map` and `lambda` to create a new list with the square of each item.

[ ]:
```
x = [1,2,3,4,5]
y = lambda a: a**2
y_of_x: list = list((map(y, x)))
```

[ ]:
```
y_of_x
```

[ ]:
```
[1, 4, 9, 16, 25]
```

6. (2 point). Write a function `gcd(a, b)` that returns the greatest common divisor of two numbers.

[ ]:
```
def gcd(a: int, b: int) -> int:
    gcd: int = 1
    for n in range(1, a + 1):
        if a % n == 0 and b % n == 0:
            gcd = n
    return gcd
```

[ ]:
```
gcd(19, 20)
```

[ ]:
```
1
```

[ ]:
```

```

[ ]:
```

```