

Python Class Exercise Set 4

Exercise

Type in below as-is and see what gets printed

```
num = 10

print ("before if statement")

if (num==10):
    print ("inside if statement")
    print (num)

print ("after if statement")
```

Exercise

Type in below as-is and see what gets printed

```
num = 20

print ("before if statement")

if (num==10):
    print ("inside if statement")
    print (num)

print ("after if statement")
```

Exercise

Type in below as-is and see what gets printed

```
num1 = 10
num2 = 20

print ("before if statement")

if (num1==10 and num2==10):
    print ("inside if statement")
    print (num1, num2)

print ("after if statement")
```

Exercise

Type in below as-is and see what gets printed

```
num1 = 10
num2 = 20

print ("before if statement")

if (num1==10 or num2==10):
    print ("inside if statement")
    print (num1, num2)

print ("after if statement")
```

Exercise

Type in below as-is and see what gets printed

```
UserXRatingsD = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5}

print ("before if statement")

if ('Z' in UserXRatingsD.keys()):
    print("key found")

print ("after if statement")
```

Exercise

Type in below as-is and see what gets printed

```
for i in range(10):  
    if (i%2==0):  
        print(i)
```

Exercise

- Let's say Users X and Y have rated 5 items (A, B, C, D, E):
 - UserXRatings^D = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5}
 - UserYRatings^D = {'A':10, 'B':20, ~~'C':30~~, 'D':40, 'E':50}
- Use a single for loop to iterate through the ratings dictionary and print the items and ratings, with keys in sorted order – but print the item and ratings only if the item was rated by both users.

Answer:

A	1	10
B	2	20
C	3	30
D	4	40
E	5	50

Answer

```
UserXRatingsD = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5}
UserYRatingsD = {'A':10, 'B':20, 'D':40, 'E':50}

for k in sorted(UserXRatingsD.keys()):
    if (k in UserYRatingsD.keys()):
        print(k, UserXRatingsD[k], UserYRatingsD[k])
```

Exercise

- Let's say we have two lists:
 - UserXRatings = [1, 2, 3, 4, 5]
 - UserYRatings = [10, 20, 30, 40, 50]
- Use a for loop and zip function to iterate through the two lists and print corresponding elements:

Answer:

1	10
2	20
3	30
4	40
5	50

Answer

```
UserXRatings = [1, 2, 3, 4, 5]
UserYRatings = [10, 20, 30, 40, 50]

for x, y in zip(UserXRatings, UserYRatings):
    print(x,y)
```

Exercise

- Let's say we have two lists:
 - UserXRatings = [1, 2, 3, 4, 5]
 - UserYRatings = [10, 20, 30, 40, 50]
- Write a function to calculate and return the Minkowski Distance:
 - `minkowskiL (ratings1, ratings2, r)`
 - check for common errors/boundary conditions
- Call the function as follows, and print the returned values:
 - `minkowskiL (UserXRatings, UserYRatings, 1)` *Answer: 135.0*
 - `minkowskiL (UserXRatings, UserYRatings, 2)` *Answer: 66.75*
 - `minkowskiL (UserXRatings, UserYRatings, 3)` *Answer: 54.74*

Answer

```
# Minkowski Distance between two vectors
def minkowskiL(ratings1, ratings2, r):

    # error check
    if (r <= 0):
        print(">>> minkowski debug: r<=0; returning -2 distance!")
        return -2

    # calculate minkowski distance
    distance = 0
    for x, y in zip(ratings1, ratings2):
        distance += pow(abs(x - y), r)

    # return value of minkowski distance
    return pow(distance, 1/r)

UserXRatings = [1, 2, 3, 4, 5]
UserYRatings = [10, 20, 30, 40, 50]

md = minkowskiL(UserXRatings, UserYRatings, 1)
print("Manhattan Distance: ", round(md, 2))

md = minkowskiL(UserXRatings, UserYRatings, 2)
print("Euclidean Distance: ", round(md, 2))

md = minkowskiL(UserXRatings, UserYRatings, 3)
print("Minkowski Distance (r=3): ", round(md, 2))
```

Exercise

- Let's say we have two dictionaries:
 - UserXRatings \mathbf{D} = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5}
 - UserYRatings \mathbf{D} = {'A':10, 'B':20, ~~'C':30,~~ 'D':40, 'E':50}
- Update the Minkowski Distance function to process Dictionaries instead of Lists (consider an item rating only if it was rated by both users):
 - minkowski \mathbf{D} (ratings1 \mathbf{D} , ratings2 \mathbf{D} , r)
- Call the function as follows, and print the returned values:
 - minkowski \mathbf{D} (UserXRatings \mathbf{D} , UserYRatings \mathbf{D} , 1) *Answer: 108.0*
 - minkowski \mathbf{D} (UserXRatings \mathbf{D} , UserYRatings \mathbf{D} , 2) *Answer: 61.04*
 - minkowski \mathbf{D} (UserXRatings \mathbf{D} , UserYRatings \mathbf{D} , 3) *Answer: 52.46*

Answer

```
# Minkowski Distance between two vectors
def minkowskiD(ratings1, ratings2, r):

    # calculate minkowski distance
    distance = 0
    for item in ratings1.keys():
        # consider item rating only if both users have rated item
        if item in ratings2.keys():
            x = ratings1[item]
            y = ratings2[item]
            distance += pow(abs(x - y), r)

    # return value of minkowski distance
    return pow(distance, 1/r)
```

```
UserXRatings = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5}
UserYRatings = {'A':10, 'B':20, 'D':40, 'E':50}
```

```
md = minkowskiD(UserXRatings, UserYRatings, 1)
print ("Manhattan Distance: ", round(md,2))
```

```
md = minkowskiD(UserXRatings, UserYRatings, 2)
print ("Euclidean Distance: ", round(md,2))
```

```
md = minkowskiD(UserXRatings, UserYRatings, 3)
print ("Minkowski Distance (r=3): ", round(md,2))
```