

CS 210: Data Management for Data Science

Midterm 1

Spring 2023

Name: Khazal Sevd NetID: KS881

This is a closed book, closed notes exam.

No electronic devices are permitted.

Name: _____

1. (10 points) Suppose we have the following python snippet:

```
x = [2, 1, 4, 5, 7]
```

```
m = 3
```

```
i = 0
```

```
y = 0
```

```
while y < m and i < len(x):
```

```
    i += 1
```

```
    y += x[i]
```

y m F
while $0 < 3$ and $0 < 5$
1
2 0 + 1

After running this code, what would the values of i and y be?

i: 2 y: 5

1 1, 1

2. 2, 4 +)

2.5 2.5

4368
4338
 $A[2] = 2 * A[1]$

42, 38 I = 1
J = 2

4, 2, 38
4338

2. (10 points) Suppose we have the following python snippet:

```
a = [1, 2, 3, 4]
```

```
i = 0
```

```
j = len(a) - 1
```

```
while i < j:
```

```
    a[i], a[j] = a[j], 2 * a[i]
```

```
    i += 1
```

```
    j -= 1
```

$A[0] = A[2]$
44 238

$0 < 3$

$A[0] = A[1]$

$A[0] = 4$

4 $A[3] = 2 * 4$

After running this code, what would the value of a be?

a: 4368

2 < 5

1 < 2

4 2, 38

(3, 2, 4, 0)

1, 3

$A[1], A[2] = A[2], 2 * a[1]$

43 49

Name: Khizel Saad

3. (20 points) Write a list comprehension for each of the following. You should not write any code outside of the [] list enclosures.

(a) Create a list of all multiples of 5 between 0 and 200, inclusive (i.e., [0, 5, ..., 195, 200]).

[num%5==0 for i in range(205)]

(b) Given a list named students containing (name, gpa) tuples, create a list of students that have a gpa greater than 3.0.

[student[1] > 3.0 for student in students]

(c) The columns of an 8×8 chess board are labelled with the letters A through H. The rows of the board are labelled with numbers 1 through 8. Then each cell can be labelled by a column followed by a row: D4, F6, A3, etc. Create a list of all possible cell labels.

letters [A, B, C, ..., H]
num [1-8]
[f"{x}{y}" for x in letters, y in num]

(d) Create a list of all pairs of integers between 1 and 10 (inclusive) whose sum is a multiple of 3. The integers can appear in any order within a pair, but no pair should be repeated. E.g., only one of (1, 5) and (5, 1) should appear in the resulting list.

[(x, y) for i in range(10) if (x+y)%3==0]

Name: _____

4. (20 points) Suppose we gather data on how many Rutgers busses break down each day and put this data in a file with the following form:

Mar 15: 13
Jan 2 :27
Sep 09 : 6
Aug 16: 192
Apr 5 : 44

Each line of the file lists the date of the reading and the number of broken-down busses that day. The dates are not in any particular order.

The date is listed with the standard 3-letter month name abbreviation with the first letter capitalized, followed by a single space, then date in 1 or 2 digits (with a possible leading zero if date is less than 10).

The date field is separated from the bus count with a colon, with any number of optional whitespaces before and after the colon.

Write a function named `filterDays` that takes as parameters an input file name, an integer lower bound, and an integer upper bound. It should return a list of all days for which the broken bus count was between the lower and upper bounds (both inclusive). The items of the list should be tuples of the form ('mm/dd', count), where mm and dd are 2-digit numeric values.

For instance, with the sample input file above, a lower bound of 10, and an upper bound of 50, the resulting list should look like this:

[('03/15', 13), ('01/02', 27), ('04/05', 44)]

filterDays(file, upper, lower)
with `open(file) as f`
`Date, occurrence = f.split(':')`
`if Date == "Mar"`
`month = 3`
`if date =`

Name: Khizel Saad

5. (20 points) Suppose we've parsed some input text to count the number of occurrences of each word and created a list of (word, count) tuples such as the following:

```
[('this', 2), ('absolutely', 1), ('amazing', 1), ('crazy', 1), ('movie', 2),  
 ('what', 1), ('funniest', 1), ('year', 1)]
```

Write a function named classify that takes this list as a parameter and returns a list of tuples grouped on word length, e.g.:

```
[(4, [('this', 2), ('what', 1), ('year', 1)]),  
 (5, [('crazy', 1), ('movie', 2)]),  
 (7, [('amazing', 1)]),  
 (8, [('funniest', 1)]),  
 (10, [('absolutely', 1)])]
```

Each tuple in the resulting list has a word length and a list of all words of that length, with their frequency counts. The list is sorted in ascending order of the word lengths, so the list of words of length 4 appears before the list of words of length 5, etc. Within each list of words of the same length, the words should appear in alphabetical order.

def classify(list):
 dict = {}
 for x in list:
 if x not in dict:
 dict[len(x)] = [x]

dict = {}

def classify(list):

for x in list:

if x not in dict:
 dict[len(x)] = [x]

=

dict

Name: _____

6. (20 points) Suppose we're given numpy arrays of names and scores, where each row in the scores array lists the midterm and final scores for that respective student. For example, in the following arrays, Carol has a 75 on the midterm and 99 on the final.

```
names = np.array([ "Alice", "Bob", "Carol", "Derek", "Erin" ])
scores = np.array([ [ 95, 98 ],
                    [ 82, 88 ],
                    [ 75, 99 ],
                    [ 80, 90 ],
                    [ 85, 82 ] ])
```

Write a function `avgMidtermNoC` that takes as parameters names and scores arrays, and returns the average score on the midterm, excluding anyone who's name starts with 'C'.

Dict = {}
list

`avgmg(name, scores)`

key = name
values = score

for x in range(len(scores):
 Dict[x] = update(y)

for key, value in Dict:
 if key[0] == 'C':
 continue

else
 List.append(Dict[key])

for x, y in List:
 x += y

avg = x / len(names) - 1

return avg