CSC 590, CSC 690 – Data Analytics
Exam #2, Fall 2024

First/Given Name: _____

Last/Family Name: _____

This exam contains 7 pages (including this cover page) and 6 questions.

- Clearly identify your answer for each problem, and try to organize your work in a reasonably coherent way, in the space provided. If you decided to use the back of a paper, note this clearly so the instructor can find your answer.
- Partial credit will be given for incorrect or incomplete answers that show a partial understanding of the relevant concepts. Irrelevant and meaningless answers will not receive partial credit.
- No electronic devices, including calculators, are allowed.
- Each student is allowed to use only a cheat sheet of size $8.5'' \times 5.75''$, which is equivalent to a half of a standard letter-sized paper. The cheat sheet can be used on both sides. Only handwritten cheat sheets are allowed, and each student is required to write their name on their cheat sheet. The cheat sheet must be submitted along with the exam upon completion.

Question	Points	Score
1	1.50	
2	2.00	
3	1.50	
4	1.50	
5	1.50	
6	1.00	
Total:	9.00	

I acknowledge that it is the responsibility of every student at Missouri State University to adhere to the university's policies on Student Academic Integrity. I confirm that I have neither given nor received any unauthorized assistance during this exam.

Signature: _____

1. (1.50 points) Indicate whether the following statements are true or false.

   (a) Global variables in a driver program are accessible in the worker nodes.
   True. Global variables are sent to the workers together with the code that is meant to be run.

   (b) The following instruction is lazy.

   ```
   rdd.filter(lambda x: x % 2 == 1)
   ```

   True. Spark will know that it has to apply that transformation but without an action, filter() will not be evaluated.

   (c) The following code adds 2 to each element of the RDD.

   ```
   rdd = sc.parallelize([1, 2, 3, 4, 5])
   rdd.mapPartitions(lambda x: x+2)
   ```

   False. This code does not work because mapPartitions() expects a partition (a list), and the function lambda x: x + 2 is adding 2 to an iterator!

   (d) The following code involves wide dependencies.

   ```
   rdd.map(lambda x: x*x)
   ```

   False. It involves narrow dependencies.

   (e) The following code involves narrow dependencies.

   ```
   rdd.reduceByKey(lambda x, y: x+y)
   ```

   False. It involves wide dependencies.

   (f) All the executors of a cluster can change the value of an accumulator variable, but they cannot read the value of the accumulator variable.
   True.

 Points earned for this question: _____

2. We define an RDD as follows:

```
rdd = sc.parallelize([4, 7, 12, 3, 16, 9, 6, 13, 8, 18])
```

(a) (0.75 points) What is the output of the following code?

```
rdd2 = rdd.filter(lambda x: x % 2 == 0)
rdd2.collect()
```

Solution:

```
[4, 12, 16, 6, 8, 18]
```

(b) (0.75 points) What is the output of the following code?

```
rdd3 = rdd2.map(lambda y: (1, 10-y) if y < 10 else (2, y-10))
rdd3.collect()
```

Solution:

```
[(1, 6), (2, 2), (2, 6), (1, 4), (1, 2), (2, 8)]
```

(c) (0.50 points) What is the output of the following code?

```
rdd4 = rdd3.reduceByKey(lambda u, v: u if u >= v else v)
rdd4.collect()
```

Solution:

```
[(2, 8), (1, 6)]
```


Points earned for this question: _____

3. We define an RDD as follows:

```
rdd = sc.parallelize([1, 3, 5, 6])
```

(a) (0.75 points) What is the output of the following code?

```
rdd2 = rdd.flatMap(lambda x: [x, x*2])
rd22.collect()
```

Solution:

```
[1, 2, 3, 6, 5, 10, 6, 12]
```

(b) (0.75 points) What is the output of the following code?

```
rdd2.takeOrdered(3, lambda s: -1 * s)
```

Solution:

```
[12, 10, 6]
```


Points earned for this question: _____

4. We have an RDD, defined as follows, which contains a collection of points:

```
rdd = sc.parallelize([(2, 1), (4, -3), (-2, 3), (1, 5), (-3, -2)]).
```

Also, we have defined a function as follows:

```
def partition_func(s):
    o = []
    for p in s:
        if p[0] > 0:
            o.append((p[1], p[0]))
    return o
```

(a) (0.75 points) What is th output of the following code?

```
rdd2 = rdd.mapPartitions(partition_func)
rdd2.collect()
```

Solution:

```
[(1, 2), (-3, 4), (5, 1)]
```

(b) (0.75 points) What is the output of the following code?

```
rdd2.reduce(lambda u, v: (u[0]+v[0], u[1]+v[1]))
```

Solution:

```
(3, 7)
```


Points earned for this question: _____

5. Consider two RDDs defined as follows:

```
people = sc.parallelize([("Ari", 24), ("Joe", 28), ("Mia", 22)])
incomes = sc.parallelize([("Joe", 40000), ("Mia", 46000), ("Eva", 42000)])
```

(a) (0.75 points) What is the output of the following code?

```
people.join(incomes).collect()
```

Solution:

```
[('Joe', (28, 40000)), ('Mia', (22, 46000))]
```

(b) (0.75 points) What is the output of the following code?

```
people.leftOuterJoin(incomes).collect()
```

Solution:

```
[('Ari', (24, None)), ('Joe', (28, 40000)), ('Mia', (22, 46000))]
```


Points earned for this question: _____

6. (1.00 points) Consider a list that contains a collection of tuples where each tuple indicates two persons that are friend with each other. An example of this list is as follows:

```
friendship = [("Em", "Sid"), ("Sid", "Bet"), ("Em", "Ann"), ("Ann", "Sid")].
```

Not that for a pair of friends a and b, there will be only one tuple in the list, which could be either (a, b) or (b, a). Write a program that prints a person who has the maximum number of friends along with the total number of friends it has. The output for the above input should look like this:

```
("Sid", 3).
```

This is because Sid is friend with three persons.

To make things easier, if more than one person has the maximum number of friends, you need to print only one of them.

Complete the following program.

```
friendship = [("Em", "Sid"), ("Sid", "Bet"), ("Em", "Ann"), ("Ann", "Sid")]
rdd = sc.parallelize(friendship)
```

Solution:

```
result = rdd.flatMap(lambda t: [(t[0], 1), (t[1], 1)]).reduceByKey(lambda x, y: x+y)\
         .takeOrdered(1, lambda u: -u[1])
result
```


Points earned for this question: _____