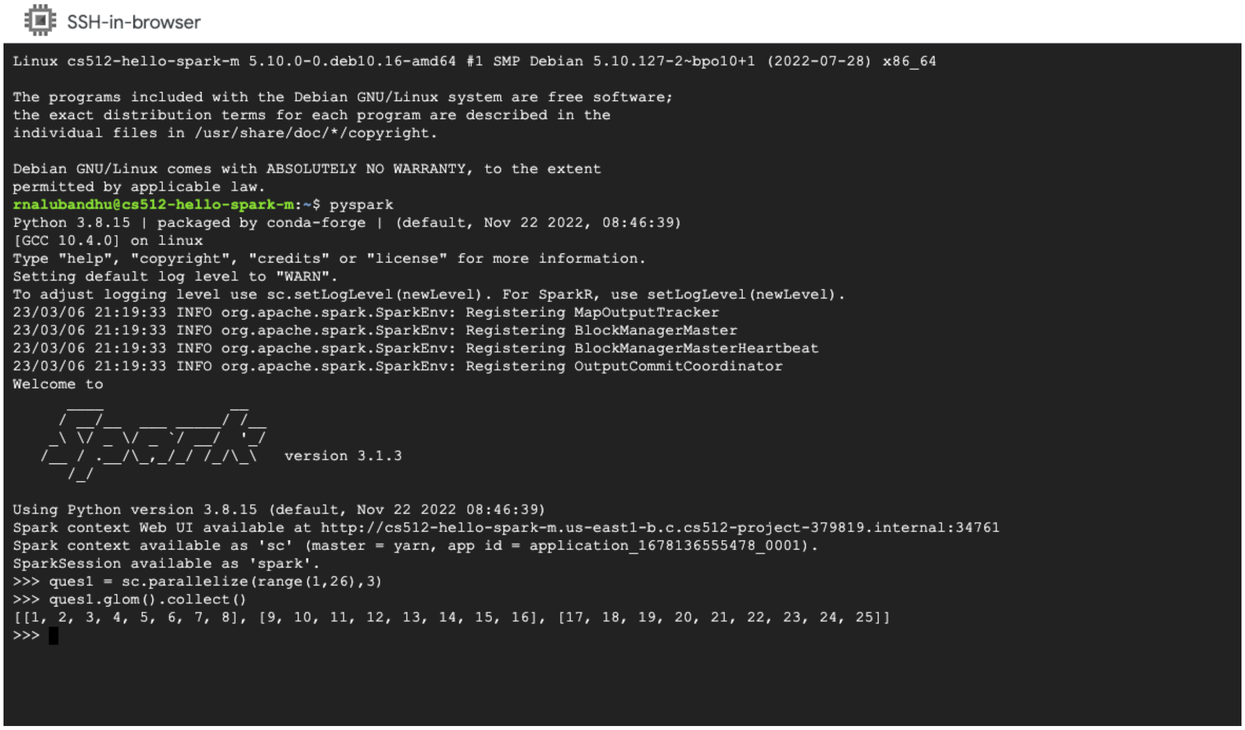
HW: Spark Activities

1) Make a list of 25 integers across 3 partitions.



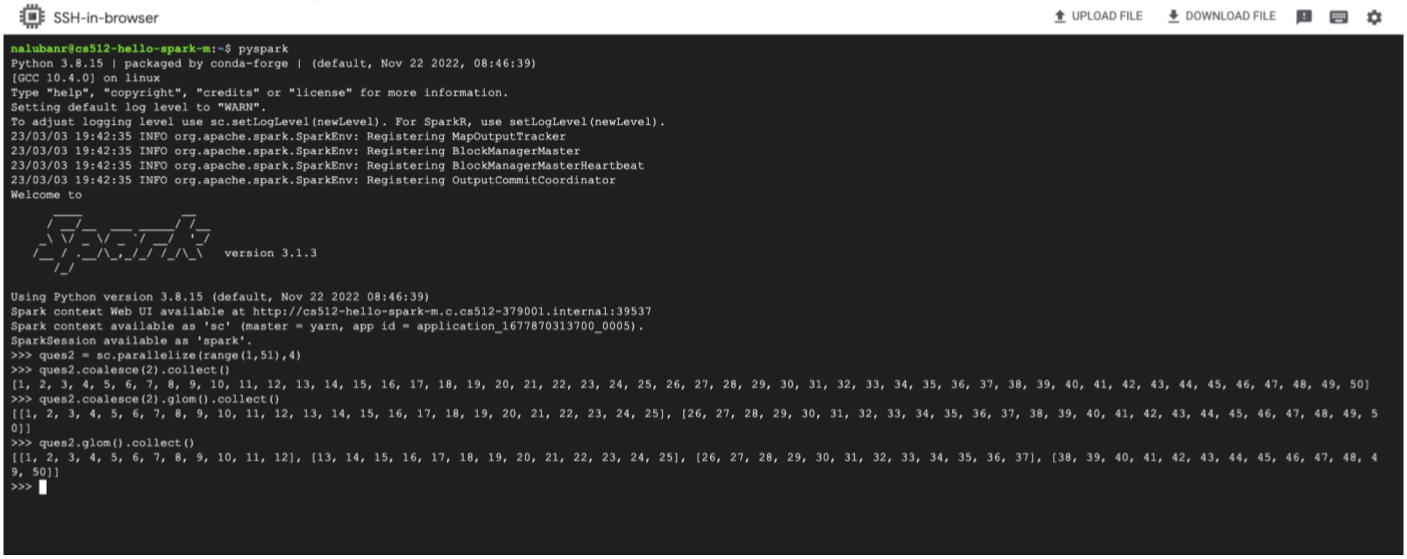
```
Linux cs512-hello-spark-m 5.10.0-0.deb10.16-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
rnalubandhu@cs512-hello-spark-m:~$ pyspark
Python 3.8.15 | packaged by conda-forge | (default, Nov 22 2022, 08:46:39)
[GCC 10.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/06 21:19:33 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/06 21:19:33 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/06 21:19:33 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/06 21:19:33 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.15 (default, Nov 22 2022 08:46:39)
Spark context Web UI available at http://cs512-hello-spark-m.us-east1-b.c.cs512-project-379819.internal:34761
Spark context available as 'sc' (master = yarn, app id = application_1678136555478_0001).
SparkSession available as 'spark'.
>>> ques1 = sc.parallelize(range(1,26),3)
>>> ques1.glom().collect()
[[1, 2, 3, 4, 5, 6, 7, 8], [9, 10, 11, 12, 13, 14, 15, 16], [17, 18, 19, 20, 21, 22, 23, 24, 25]]
>>> 
```

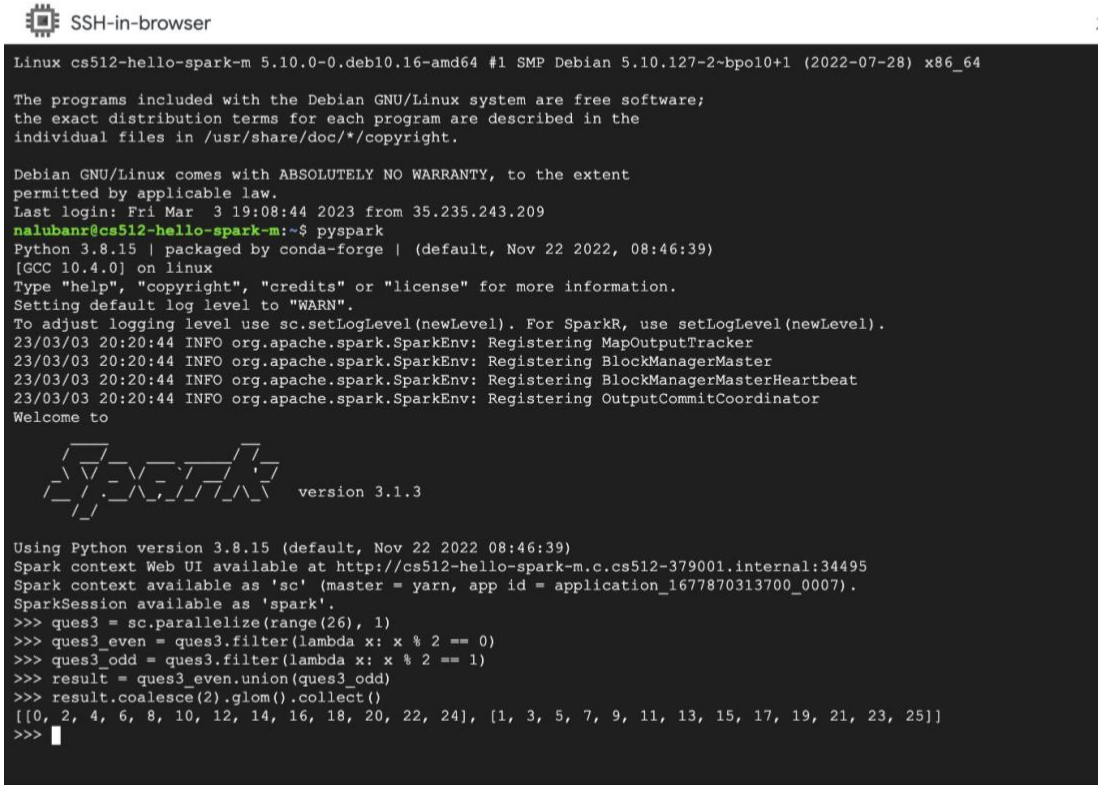
2) Make a list of 50 integers across 4 partitions, efficiently convert it to 2 partitions.



```
nalubanr@cs512-hello-spark-m:~$ pyspark
Python 3.8.15 | packaged by conda-forge | (default, Nov 22 2022, 08:46:39)
[GCC 10.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/03 19:42:35 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/03 19:42:35 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/03 19:42:35 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/03 19:42:35 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.15 (default, Nov 22 2022 08:46:39)
Spark context Web UI available at http://cs512-hello-spark-m.c.cs512-379001.internal:39537
Spark context available as 'sc' (master = yarn, app id = application_1677870313700_0005).
SparkSession available as 'spark'.
>>> ques2 = sc.parallelize(range(1,51),4)
>>> ques2.coalesce(2).collect()
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]
>>> ques2.coalesce(2).glom().collect()
[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25], [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5
0]]
>>> ques2.glom().collect()
[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25], [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37], [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 4
9, 50]]
>>> 
```

3) Starting with a list of 26 integers 0 through 25 on 1 partition, end with a list of 26 integers split among two partitions, even numbers on one and odd on the other.



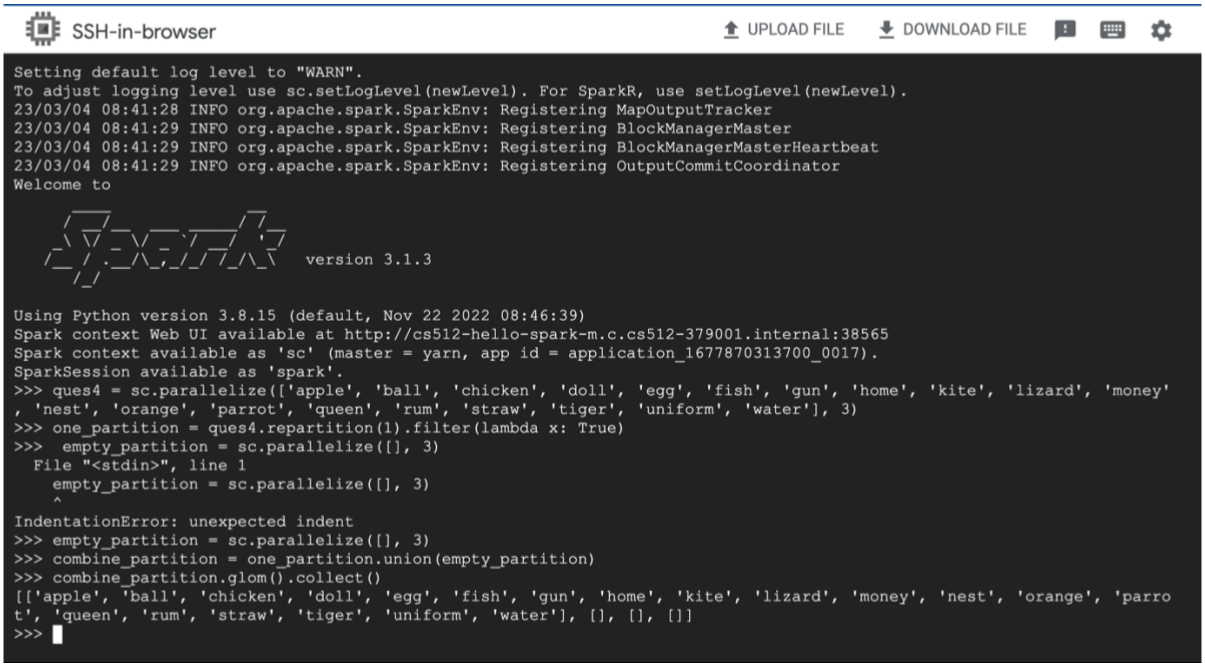
```
Linux cs512-hello-spark-m 5.10.0-0.deb10.16-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar  3 19:08:44 2023 from 35.235.243.209
nalubanr@cs512-hello-spark-m:~$ pyspark
Python 3.8.15 | packaged by conda-forge | (default, Nov 22 2022, 08:46:39)
[GCC 10.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/03 20:20:44 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/03 20:20:44 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/03 20:20:44 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/03 20:20:44 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.15 (default, Nov 22 2022 08:46:39)
Spark context Web UI available at http://cs512-hello-spark-m.c.cs512-379001.internal:34495
Spark context available as 'sc' (master = yarn, app id = application_1677870313700_0007).
SparkSession available as 'spark'.
>>> ques3 = sc.parallelize(range(26), 1)
>>> ques3_even = ques3.filter(lambda x: x % 2 == 0)
>>> ques3_odd = ques3.filter(lambda x: x % 2 == 1)
>>> result = ques3_even.union(ques3_odd)
>>> result.coalesce(2).glom().collect()
[[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24], [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25]]
>>> 
```

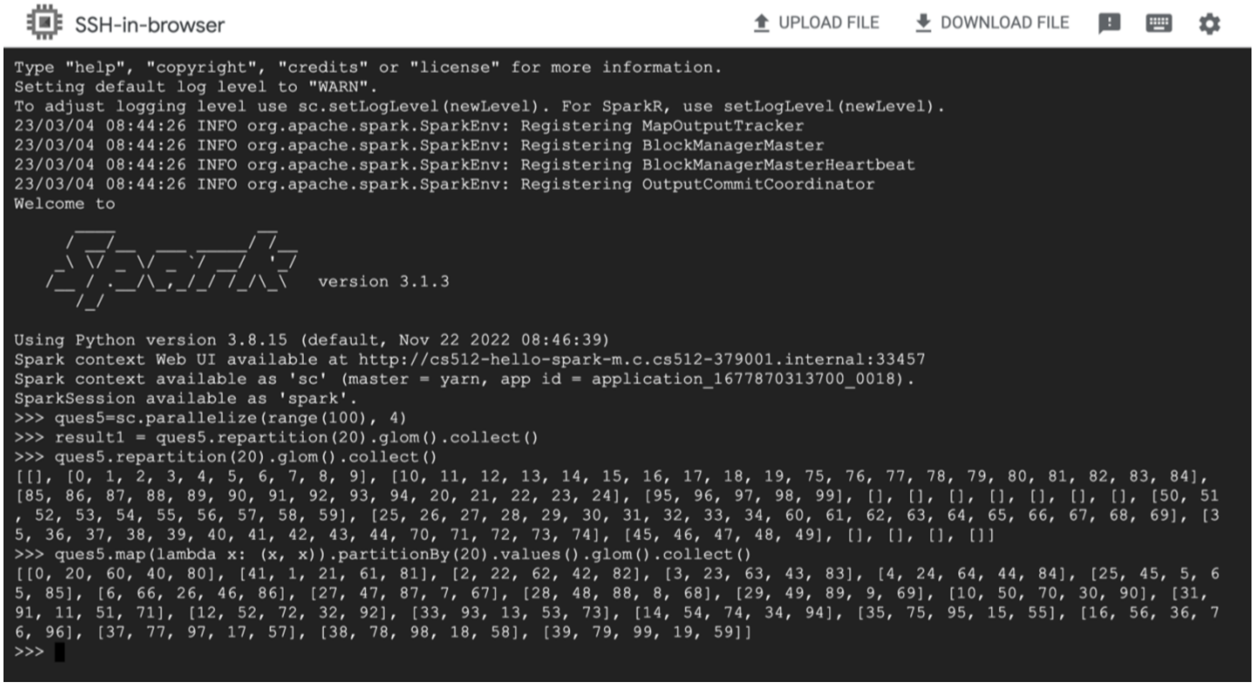
4) Starting with 20 strings split somewhat evenly across 3 partitions, end with 4 partitions will ALL the strings stored in one with the other 3 empty.



```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/04 08:41:28 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/04 08:41:29 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/04 08:41:29 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/04 08:41:29 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.15 (default, Nov 22 2022 08:46:39)
Spark context Web UI available at http://cs512-hello-spark-m.c.cs512-379001.internal:38565
Spark context available as 'sc' (master = yarn, app id = application_1677870313700_0017).
SparkSession available as 'spark'.
>>> ques4 = sc.parallelize(['apple', 'ball', 'chicken', 'doll', 'egg', 'fish', 'gun', 'home', 'kite', 'lizard', 'money'
, 'nest', 'orange', 'parrot', 'queen', 'rum', 'straw', 'tiger', 'uniform', 'water'], 3)
>>> one_partition = ques4.repartition(1).filter(lambda x: True)
>>>  empty_partition = sc.parallelize([], 3)
  File "<stdin>", line 1
    empty_partition = sc.parallelize([], 3)
    ^
IndentationError: unexpected indent
>>> empty_partition = sc.parallelize([], 3)
>>> combine_partition = one_partition.union(empty_partition)
>>> combine_partition.glom().collect()
[['apple', 'ball', 'chicken', 'doll', 'egg', 'fish', 'gun', 'home', 'kite', 'lizard', 'money', 'nest', 'orange', 'parro
t', 'queen', 'rum', 'straw', 'tiger', 'uniform', 'water'], [], [], []]
>>> 
```

5) Compare the results of using repartition(20) directly on an RDD containing the values 0 through 99 with the results of first making a key value pair using the value as the key, then using partition By(20)



```
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/04 08:44:26 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/04 08:44:26 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/04 08:44:26 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/04 08:44:26 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.15 (default, Nov 22 2022 08:46:39)
Spark context Web UI available at http://cs512-hello-spark-m.c.cs512-379001.internal:33457
Spark context available as 'sc' (master = yarn, app id = application_1677870313700_0018).
SparkSession available as 'spark'.
>>> ques5=sc.parallelize(range(100), 4)
>>> result1 = ques5.repartition(20).glom().collect()
>>> ques5.repartition(20).glom().collect()
[[], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84],
[85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 20, 21, 22, 23, 24], [95, 96, 97, 98, 99], [], [], [], [], [], [], [], [50, 51
, 52, 53, 54, 55, 56, 57, 58, 59], [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69], [3
5, 36, 37, 38, 39, 40, 41, 42, 43, 44, 70, 71, 72, 73, 74], [45, 46, 47, 48, 49], [], [], [], []]
>>> ques5.map(lambda x: (x, x)).partitionBy(20).values().glom().collect()
[[0, 20, 60, 40, 80], [41, 1, 21, 61, 81], [2, 22, 62, 42, 82], [3, 23, 63, 43, 83], [4, 24, 64, 44, 84], [25, 45, 5, 6
5, 85], [6, 66, 26, 46, 86], [27, 47, 87, 7, 67], [28, 48, 88, 8, 68], [29, 49, 89, 9, 69], [10, 50, 70, 30, 90], [31,
91, 11, 51, 71], [12, 52, 72, 32, 92], [33, 93, 13, 53, 73], [14, 54, 74, 34, 94], [35, 75, 95, 15, 55], [16, 56, 36, 7
6, 96], [37, 77, 97, 17, 57], [38, 78, 98, 18, 58], [39, 79, 99, 19, 59]]
>>> 
```

Using repartition(20) directly on an RDD containing the values 0 through 99 it will result in creating 20 partitions with varying numbers of elements in each partition, depending on the distribution of the data whereas creating a key-value pair using the value as the key, and then using partitionBy(20) will create 20 partitions where each partition will contain a contiguous range of keys.