Task5 - Lazy Evaluation in Spark

```
from pyspark.sql import SparkSession
import math
```

```
spark = SparkSession.builder.appName("LazyEvaluation").getOrCreate()
sc = spark.sparkContext
```

```
# create an rdd containing 100 numbers with 4 partitions
data = [i for i in range(100)]
data_rdd = sc.parallelize(data, 4)
print(data_rdd)
print("No. of partitions: {}".format(data_rdd.getNumPartitions()))
```

```
ParallelCollectionRDD[0] at readRDDFromFile at PythonRDD.scala:274
No. of partitions: 4
```

```
# Transformation: multiplying each number by 2
transform_rdd = data_rdd.map(lambda x: x*2)
print(transform_rdd)
print("Lazy Evaluation Execution plan:\n {}".format(transform_rdd.toDebugString()))
```

```
PythonRDD[1] at RDD at PythonRDD.scala:53
Lazy Evaluation Execution plan:
 b'(4) PythonRDD[1] at RDD at PythonRDD.scala:53 []\n |  ParallelCollectionRDD[0] at readRDDF
romFile at PythonRDD.scala:274 []'
```

```
# Add another transformation and check execution plan
new_transform_rdd = transform_rdd.map(lambda x: x-2)
print(new_transform_rdd)
print("Updated Lazy Evaluation Execution plan:\n {}".format(new_transform_rdd.toDebugString()
```

```
PythonRDD[2] at RDD at PythonRDD.scala:53
Updated Lazy Evaluation Execution plan:
 b'(4) PythonRDD[2] at RDD at PythonRDD.scala:53 []\n |  ParallelCollectionRDD[0] at readRDDF
romFile at PythonRDD.scala:274 []'
```

```
# Perform Action, all transformations gets executed now
rdd_result = new_transform_rdd.collect()
print(rdd_result)
```

```
[-2, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 4
6, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92
, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130,
132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168
, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196]
```

```
sc.stop()
```

One of the ways by which spark saves computation and increases speed is through Lazy Evaluation. Spark maintains a DAG of transformations/operations to perform and keeps on simplifying the DAG as new transformations are added. The transformations are only executed once an action is triggered.

In the program, we can clearly observe that no transformations are executed until we trigger `collect()` action on RDD. Until this step, Spark keeps on adding and simplifying transformations in the form of DAG, which we can see using `toDebugString()` command. As soon as we trigger `collect()` action, all required transformations are executed and we obtain the result in form a list.