Welcome to exercise two of “Apache Spark for Scalable Machine Learning on BigData”. In this exercise you’ll apply the basics of functional and parallel programming.

Again, please use the following two links for your reference:https://spark.apache.org/docs/latest/api/python/pyspark.html#pyspark.RDDhttps://spark.apache.org/docs/latest/rdd-programming-guide.html

Let’s actually create a python function which decides whether a value is greater than 50 (True) or not (False).

```
In [ ]: def gt50(i):
            if i > 50:
                return True
            else:
                return False
```

```
In [ ]: print(gt50(4))
        print(gt50(51))
```

Let’s simplify this function

```
In [ ]: def gt50(i):
            return i > 50
```

```
In [ ]: print(gt50(4))
        print(gt50(51))
```

Now let’s use the lambda notation to define the function.

```
In [ ]: gt50 = lambda i: i > 50
```

```
In [ ]: print(gt50(4))
        print(gt50(51))
```

```
In [ ]: #let's shuffle our list to make it a bit more interesting
        from random import shuffle
        l = list(range(100))
        shuffle(l)
        rdd = sc.parallelize(l)
```

Let’s filter values from our list which are equals or less than 50 by applying our “gt50” function to the list using the “filter” function. Note that by calling the “collect” function, all elements are returned to the Apache Spark Driver. This is not a good idea for BigData, please use “.sample(10,0.1).collect()” or “take(n)” instead.

```
In [ ]: rdd.filter(gt50).collect()
```

We can also use the lambda function directly.

```
In [ ]: rdd.filter(lambda i: i > 50).collect()
```

Let’s consider the same list of integers. Now we want to compute the sum for elements in that list which are greater than 50 but less than 75. Please implement the missing parts.

```
In [ ]: rdd.filter(lambda x: $$).filter(lambda x: $$).$$()
```

You should see "1500" as answer. Now we want to know the sum of all elements. Please again, have a look at the API documentation and complete the code below in order to get the sum.