# BigData: Assignment-1

## Problem Statement

Strike Rate is the average runs a batsman scores in 100 balls. Given the input, find the final strike rate of each batsman.

## Mapper

Input : Array of JSON Objects

Example :

```
[
    { "name": "xyz", "runs": 100, "balls": 100 },
    { "name": "xyz", "runs": 10, "balls": 10 },
    { "name": "abc", "runs": 30, "balls": 10 },
    { "name": "abc", "runs": 20, "balls": 5 },
    { "name": "abc", "runs": 10, "balls": 42 }
]
```

Output :

- Output must be the name,local_strike_rate
- local_strike_rate refers to the strike rate of the particular match

**Strike Rate formula** = (runs/balls) * 100 [rounded upto 3 decimal places].

## Reducer

Input : Same format as Mapper output

Output : Independent JSON Objects

- output of the reducer has the following keys : name of batsman, and average strike rate accross all matches

**average strike rate = sum of all local strike rates / total matches** [rounded upto 3 decimal places]

Example :

```
{ "name": "xyz", "strike_rate": 100 }
{ "name": "abc", "strike_rate": 241.27 }
```

## Sample Input

1. sample_data.json
2. expected_output_sample_data.txt for the above input

```
{"name": "Deepti", "strike_rate": 61.551}
{"name": "Harmanpreet", "strike_rate": 87.124}
{"name": "Ishan", "strike_rate": 85.077}
{"name": "Jemimah", "strike_rate": 77.407}
{"name": "Renuka", "strike_rate": 74.35}
{"name": "Rohit", "strike_rate": 71.464}
{"name": "Shubman", "strike_rate": 66.041}
{"name": "Smriti", "strike_rate": 57.807}
{"name": "VVS Laxman", "strike_rate": 64.078}
{"name": "Virat", "strike_rate": 89.928}
```

## Test Dataset

Test your code with the following dataset once it passes the sample input dataset

1. **Input**: large_data.json
2. **Output**: expected_output_large_data.txt

The link for Drive having the above files

https://drive.google.com/drive/folders/10N7VRIOhwi4O70D8PHmnzRRhC1zkIw8s?usp=share_link

# Instructions

1. Write a python mapper
    - Name : `mapper.py`
    - Read the specification for input and output as mentioned above
    - Only packages that can be imported are : json and sys
2. Write a python reducer to perform the aggregation
    - Name : `reducer.py`
    - Read the specification for input and output as mentioned above
    - Only packages that can be imported are : sys
3. Test it out with the sample dataset given and check the expected output
4. Adhere to the submission guidelines

# Testing instructions

## Local testing

```
cat <path_to_dataset>.json | ./mapper.py | sort -k 1,1
| ./reducer.py
```

## Hadoop testing

Put the input file in hdfs first

- Make a dir in HDFS
    - `hdfs dfs -mkdir /example`
- Put the input file in HDFS
    - `hdfs dfs -put <path to input file in local system> /example`
- Now your input file is in /example/<input_file_name>
- This HDFS path to input file is the input path in the below command.

```
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-
streaming-3.3.6.jar \
-mapper "$PWD/mapper.py" \
-reducer "$PWD/reducer.py" \
-input  <path_to_input_in_hdfs> \
-output <path_to_output_folder_in_hdfs>
```

Reference Links

https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/

https://www.geeksforgeeks.org/hadoop-streaming-using-python-word-count-problem/

**Last Date for Submission:09-02-2024**

**Upload should should contain 3 Files**

**1) Code PDF**

**Prepare PDF with the mapper code , reducer code, commands to execute.(keep separate sub headings)**

**2) Mapper_output File**

**3) Reducer_output File**