Data series monitor

The goal of this assignment is to 1) write a program that tracks a continuous stream of input values and computes some properties over them and 2) analyze/describe general performance properties over that program. Any programming language is accepted for which public documentation is available, with preference for C, C++, Java, Python, Haskell, ML, Go or Rust.

This assignment aims to exercise your ability to:

- write a program working with asynchronous input and output;
- perform simple computations on finite time series;
- reason about basic run-time behavior of your own code.

Questions are free: the event of you asking a question and which question you ask will not influence how we evaluate your work (neither positively or negatively). Feel free to ask anything.

Specification of the program

The program must read on a continuous input stream and produce results on two continous output streams. We will call these streams 0, 1 and 2.

Data will be delivered by an external system to stream 0 as input to the program. The input format is textual: integer numbers in base 10, each followed by a newline character.

The numbers arrive on stream 0 *irregularly over time* with unpredictable and unbounded delays.

The program must output a copy of each input from stream 0 immediately on stream 1 and, every 1 second, independently of the rate of arrival of numbers on stream 0, output the following values separated by spaces and terminated with a newline character on stream 2:

- the sliding average of the inputs received in the preceeding 10 seconds,
- the quantized time integral of the inputs received in the preceeding 10 seconds,
- the age of the oldest input value in the last 10 seconds,
- the age of the youngest input input value in the last 10 seconds.

The 4 values must be computed approximately in floating point. The first two should use at least one digit of decimal precision, the last two up to 6 (microsecond precision).

Stream 1 or 2 may occasionally block. If this happens, the output due to be delivered while a stream is blocked must be abandoned. In particular the first full line of values emitted after an output stream becomes unblocked must not be outdated.

Analysis of the program

You should provide a short explanation to estimate:

- memory usage of your program in bytes as a function of the number of input values received in a 10 seconds interval;
- the stream-to-stream latency in seconds between streams 0 and 1 (delay between the arrival of input on stream 0 and the emission of its copy on stream 1);
- the time jitter in seconds on stream 2 assuming stream 2 is not blocked.

(Your explanation can either be *analytical*, or you can propose a *methodology* about how to estimate this experimentally. When choosing the latter, you do not need to run the experiment yourself.)

Tips

The quantized time integral of a time series:

```
t1  v1
t2  v2
t3  v3
t4  v4
...
```

is defined either as:

```
(t2 - t1) * v1 + (t3 - t2) * v2 + (t4 - t3) * v3 ...
```

or as:

```
(t2 - t1) * v2 + (t3 - t2) * v3 + (t4 - t3) * v4 ...
```

We suggest you use the 2nd form. (But we accept both)

Example input

You can use the following program (Python) to generate a test input stream:

```python
from random import random
from time import sleep
while True:
   print(int(random()*(10**(1+random()*4))))
   sleep(random()*(2**(-2+random()*3)))
```