Lab 9b: Parallel Encode

- Deadline: 15 November, 2022, Tuesday, 23:59, SST
- Mark: 2%

Files

You are given the following files:

- `Lab9b.java` : The main function.
- `Streaming.java` : The skeleton file.
- `Pair.java` : The usual pair class.

There are test cases inside `input` and `output` folder. You are not allowed to change `Lab9b.java` and `Pair.java` . Any modification will nullify your mark immediately.

Problem Description

Consider a sequence of element (*can be of any type*), for instance:

```
1   A A A B B A A A A C A A A A
```

We can encode this sequence into something more compact by recording the number of consecutive elements together with the element. We say that two elements are equal if the `equals` method returns `true` .

The above sequence can first be split into several consecutive segments:

Consecutive Segment	Encoded Segment
A A A	(3, A)
B B	(2, B)
A A A A	(4, A)

Consecutive Segment	Encoded Segment
C	(1, C)
A A A A	(4, A)

Each element is now a pair. So the run-length encoding is:

```
1   (3, A), (2, B), (4, A), (1, C), (4, A)
```

Write a static method using only streams:

- `public static List<Pair<Integer, T>> decode(Stream<T> stream)` : returns a list that is the result of encoding the given stream

Submission

Submit only the following files:

- `Lab9b.java`
- `Streaming.java`
- `Pair.java`