You are given the following parallel Map/Reduce “framework” for processing a collection of Strings using Java Streams:

```java
List<String> list = Arrays.asList("Rice", "Owls", "are", "the", "best");
var value =
    list.stream().parallel()
        .filter(____A______)
        .map(____B______)
        .reduce(____C______);
```

Using this framework, solve the following problems by filling in the blanks A, B and C (note that C can be 1, 2 or 3 arguments, depending on which variant of reduce you choose):

1. Find all the strings that contain the letter “s”, convert them all to upper case, and concatenate them
2. Find the total length of all the strings that start with a lowercase letter
Worksheet: Map/Reduce

1. Find all the strings that contain the letter “s”, convert them all to upper case, and concatenate them

List<String> list = Arrays.asList("Rice", "Owls", "are", "the", "best");
var value =
list.stream().parallel()
  .filter(e -> e.contains("s"))
  .map(e -> e.toUpperCase())
  .reduce("", String::concat);

“OWLSBEST"
3. Find the total length of all the strings that start with a lowercase letter

List<String> list = Arrays.asList("Rice", "Owls", "are", "the", "best");

var value =
    list.stream().parallel()
       .filter(e -> e.charAt(0) >= 'a' && e.charAt(0) <= 'z')
       .map(String::length)
       .reduce(0, Integer::sum);

10