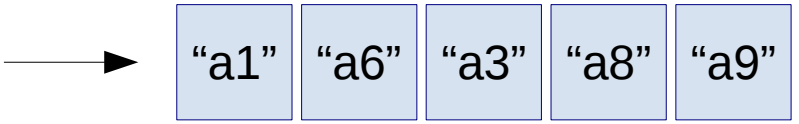
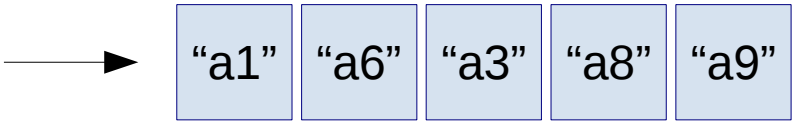
Streams revisited

```
public static void pipe7Test() {
    String sa[] = {"a1", "a6", "a3", "a8", "a9"};

    int summe = Arrays.stream(sa)
                .map(s -> s.substring(1))
                .map(s -> Integer.parseInt(s))
                .filter(i -> i > 6)
                .reduce(0, Integer::sum);

    System.out.println("Summe: " + summe);
}
```

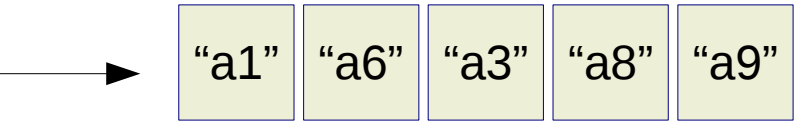
String sa[] = {"a1", "a6", "a3", "a8", "a9"};



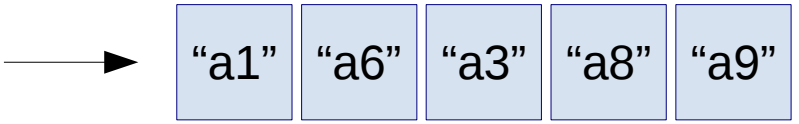
String sa[] = {"a1", "a6", "a3", "a8", "a9"};



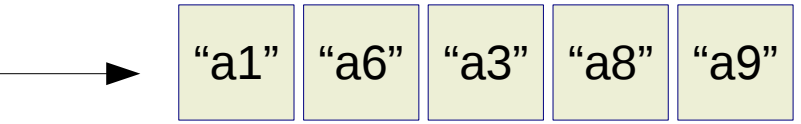
Arrays.stream(sa)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

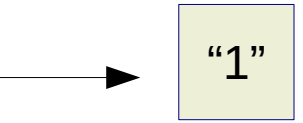


Arrays.stream(sa)

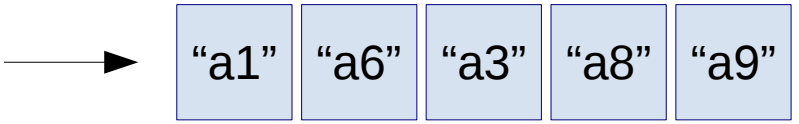


map(s -> s.substring(1))

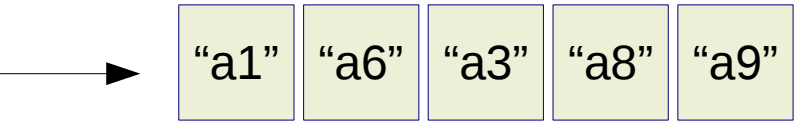
“a1”->”a1”.substring(1)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

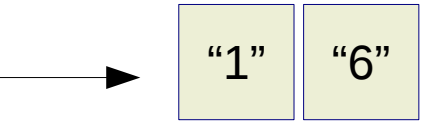


Arrays.stream(sa)

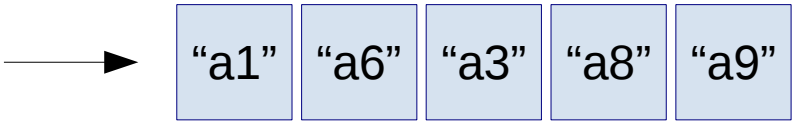


map(s -> s.substring(1))

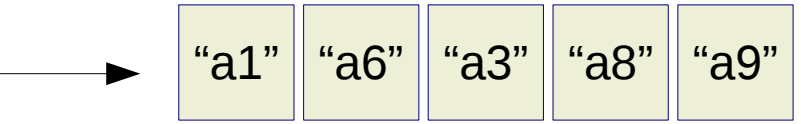
“a6”->”a6”.substring(1)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

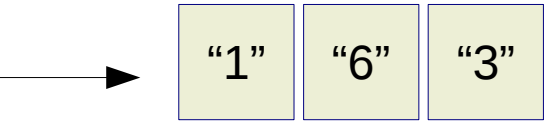


Arrays.stream(sa)

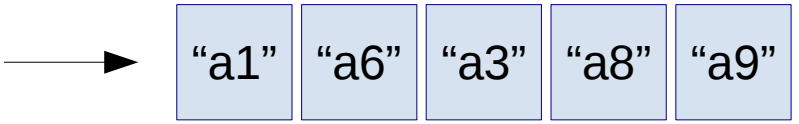


map(s -> s.substring(1))

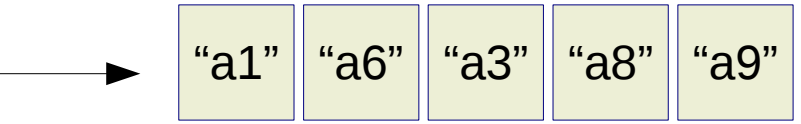
“a3”->”a3”.substring(1)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

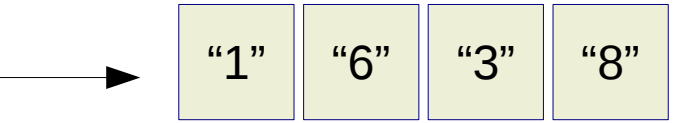


Arrays.stream(sa)

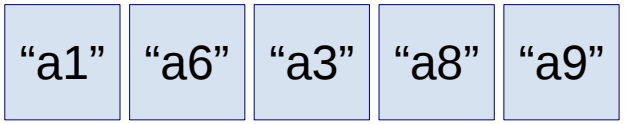


map(s -> s.substring(1))

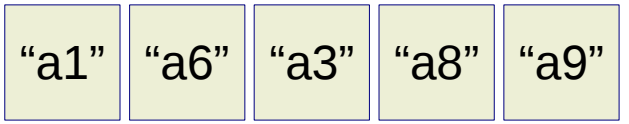
“a8”->”a8”.substring(1)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

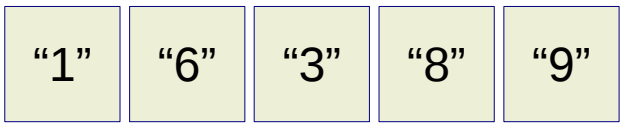


Arrays.stream(sa)



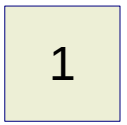
map(s -> s.substring(1))

“a9”->”a9”.substring(1)

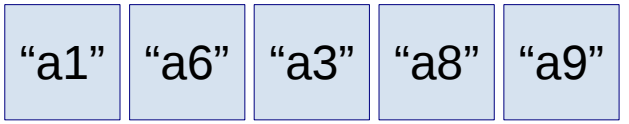


map(s -> Integer.parseInt(s))

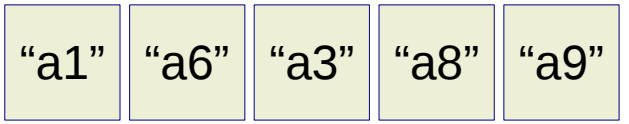
“1”->Integer.parseInt(“1”)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

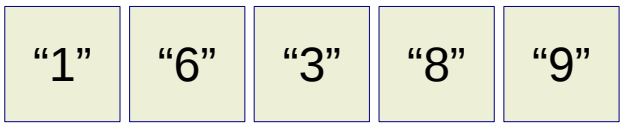


Arrays.stream(sa)



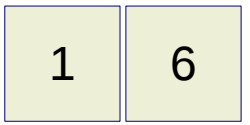
map(s -> s.substring(1))

“a9”->”a9”.substring(1)

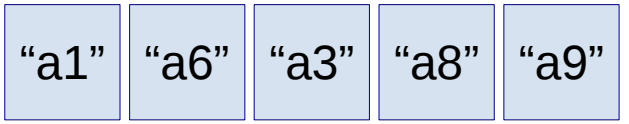


map(s -> Integer.parseInt(s))

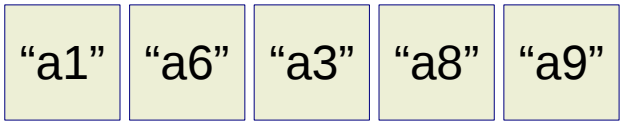
“6”->Integer.parseInt(“6”)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

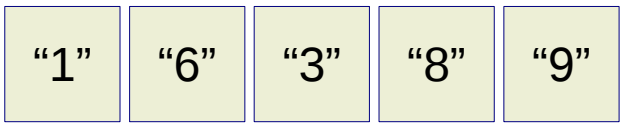


Arrays.stream(sa)



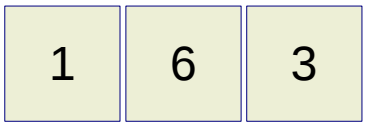
map(s -> s.substring(1))

“a9”->”a9”.substring(1)

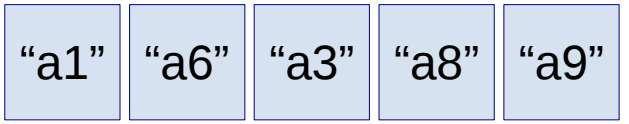


map(s -> Integer.parseInt(s))

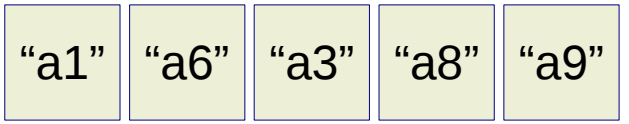
“3”->Integer.parseInt(“3”)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

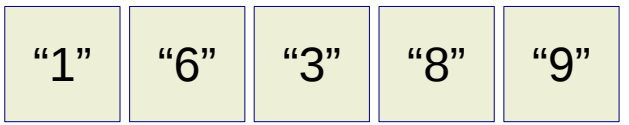


Arrays.stream(sa)



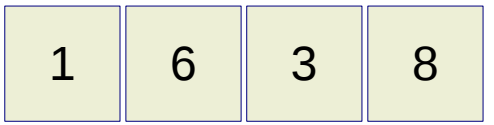
map(s -> s.substring(1))

“a9”->”a9”.substring(1)

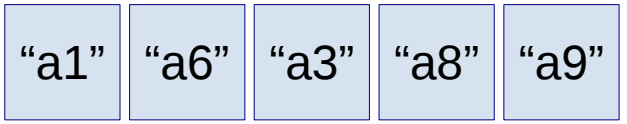


map(s -> Integer.parseInt(s))

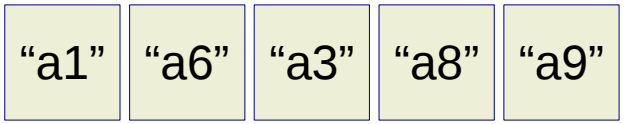
“8”->Integer.parseInt(“8”)



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

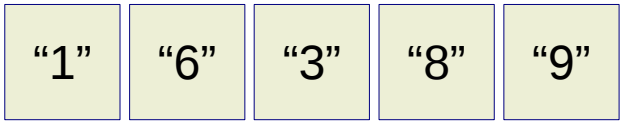


Arrays.stream(sa)



map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

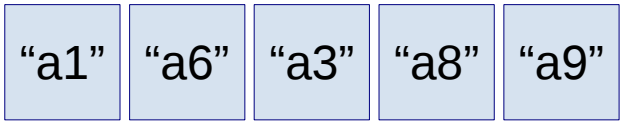
“9”->Integer.parseInt(“9”)

filter(i -> i > 6)

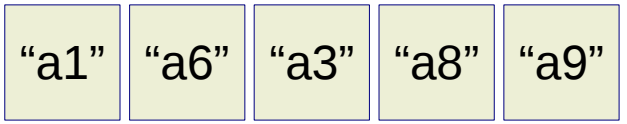
1->1>6

Results false. Nothing is added to result set

String sa[] = {"a1", "a6", "a3", "a8", "a9"};

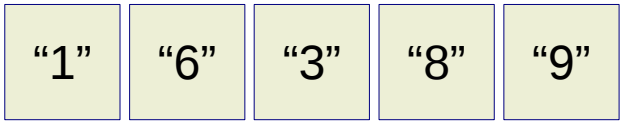


Arrays.stream(sa)



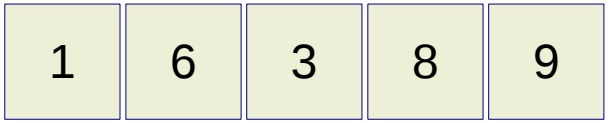
map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

“9”->Integer.parseInt(“9”)

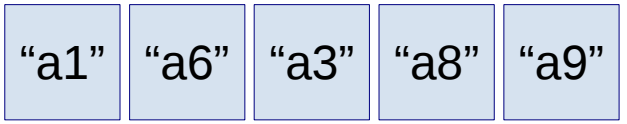


filter(i -> i > 6)

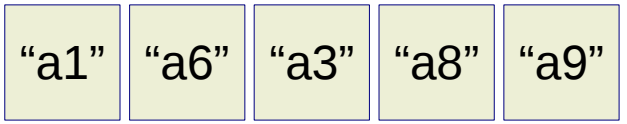
6->6>6

Results false. Nothing is added to result set

String sa[] = {"a1", "a6", "a3", "a8", "a9"};

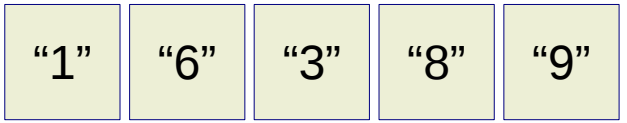


Arrays.stream(sa)



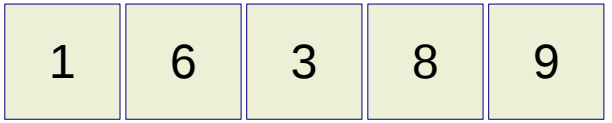
map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

“9”->Integer.parseInt(“9”)

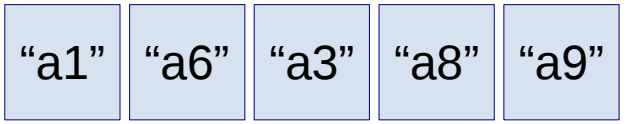


filter(i -> i > 6)

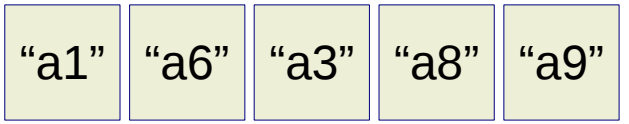
3->3>6

Results false. Nothing is added to result set

String sa[] = {"a1", "a6", "a3", "a8", "a9"};

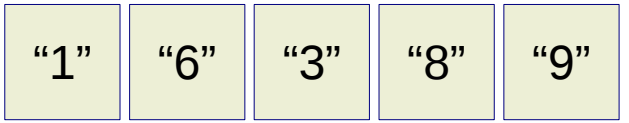


Arrays.stream(sa)



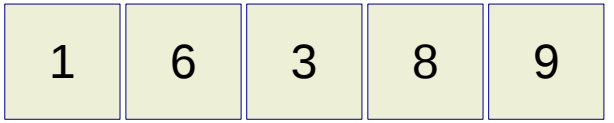
map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

“9”->Integer.parseInt(“9”)

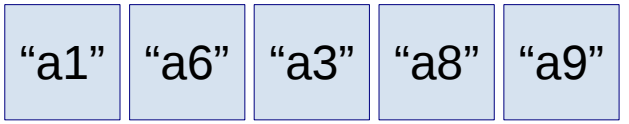


filter(i -> i > 6)

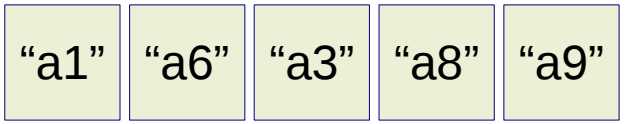
8->8>6

Results true. 8 is added to result set

String sa[] = {"a1", "a6", "a3", "a8", "a9"};

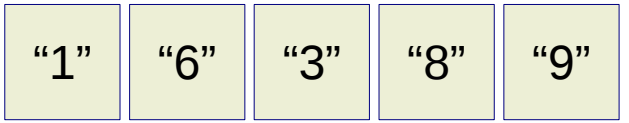


Arrays.stream(sa)



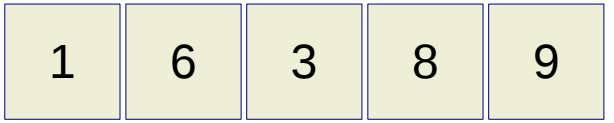
map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

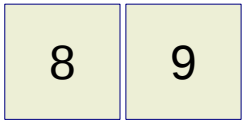
“9”->Integer.parseInt(“9”)



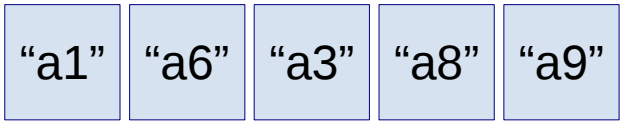
filter(i -> i > 6)

9->9>6

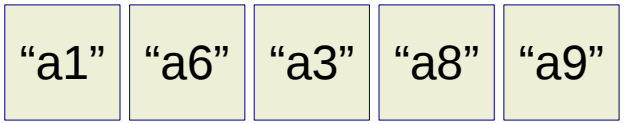
Results true. 9 is added to result set



String sa[] = {"a1", "a6", "a3", "a8", "a9"};

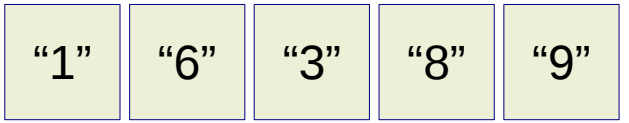


Arrays.stream(sa)



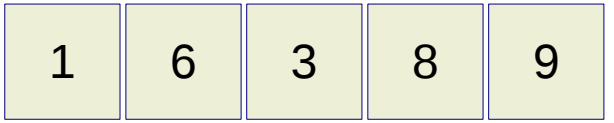
map(s -> s.substring(1))

“a9”->”a9”.substring(1)



map(s -> Integer.parseInt(s))

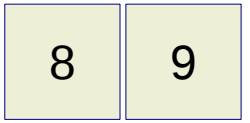
“9”->Integer.parseInt(“9”)



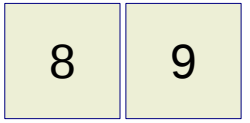
filter(i -> i > 6)

9->9>6

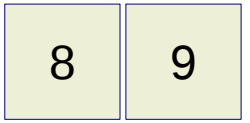
Results true. 9 is added to result set



reduce(0, Integer::sum);



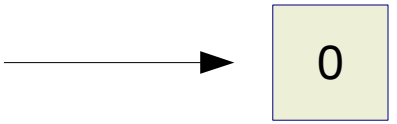
reduce(0, Integer::sum);



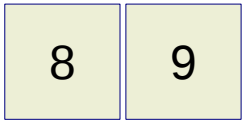
`Integer::sum` is a `BiFunction`

```
public static int sum(int a,int b)
```

Initialize result with 0

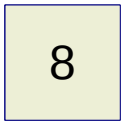


reduce(0, Integer::sum);

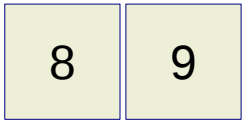


`Integer::sum` is a `BiFunction`

```
public static int sum(int a,int b)
```

`Integer.sum(0,8)` → 

reduce(0, Integer::sum);

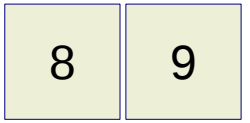


`Integer::sum` is a `BiFunction`

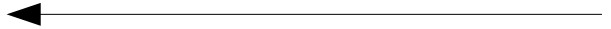
```
public static int sum(int a,int b)
```

`Integer.sum(8,9)` → 17

reduce(0, Integer::sum);



`Integer::sum` is a `BiFunction`

```
public static int sum(int a,int b)
```

`int summe`