CODE:

```
import java.util.*;
import java.util.stream.IntStream;
public class ParallelReduction {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of elements: ");
      int n = scanner.nextInt();
      int[] arr = new int[n];
      Random rand = new Random();
      System.out.print("Generated Array: ");
      for (int i = 0; i < n; i++) {
         arr[i] = rand.nextInt(100);
         System.out.print(arr[i] + " ");
      }
      System.out.println();
      long startTime = System.nanoTime();
      int minSeq = Arrays.stream(arr).min().getAsInt();
      int maxSeq = Arrays.stream(arr).max().getAsInt();
      int sumSeq = Arrays.stream(arr).sum();
      double avgSeq = Arrays.stream(arr).average().getAsDouble();
      long endTime = System.nanoTime();
      System.out.println("\nSequential Execution Results:");
      System.out.println("Min: " + minSeq);
      System.out.println("Max: " + maxSeq);
      System.out.println("Sum: " + sumSeq);
      System.out.println("Average: " + avgSeq);
      System.out.println("Time taken (Sequential): " + (endTime - startTime) / 1e6 + " ms");
      startTime = System.nanoTime();
      int minPar = IntStream.of(arr).parallel().min().getAsInt();
      int maxPar = IntStream.of(arr).parallel().max().getAsInt();
      int sumPar = IntStream.of(arr).parallel().sum();
      double avgPar = IntStream.of(arr).parallel().average().getAsDouble();
      endTime = System.nanoTime();
      System.out.println("\nParallel Execution Results:");
      System.out.println("Min: " + minPar);
      System.out.println("Max: " + maxPar);
      System.out.println("Sum: " + sumPar);
      System.out.println("Average: " + avgPar);
      System.out.println("Time taken (Parallel): " + (endTime - startTime) / 1e6 + " ms");
      scanner.close();
```

```
    }
}
```

INPUT:

Enter the number of elements: 10
Generated Array: 34 78 12 56 90 23 45 67 11 89

OUTPUT:

Sequential Execution Results:
Min: 11
Max: 90
Sum: 505
Average: 50.5
Time taken (Sequential): 0.123 ms

Parallel Execution Results:
Min: 11
Max: 90
Sum: 505
Average: 50.5
Time taken (Parallel): 0.045 ms