Pregunta 1

Given:

    var nums = List.of(1, 2, 3, 4, 5, 6, 7).stream();

    Predicate<Integer> p = //a predicate goes here

    Optional<Integer> value = nums.filter(p).reduce((a, b)->a+b);

    value.ifPresent(System.out::println);

Choose 2:

A. setting p to a->a<0; will produce no output.
B. setting p to a->a<0; will generate a NullPointerException.
C. setting p to a->a<0; will generate a NoSuchElementException.
D. setting p to a->a%2==0; will produce 12.
E. setting p to a->a%2==0; will produce 16.

Pregunta 2

Given the Person class with age and name along with getter and setter methods, and this code fragment:

```java
List<Person> persons = new ArrayList(List.of(new Person(44,"Tom"),
                                    new Person(40,"Aman"),
                                    new Person(40,"Peter")));
persons.sort(Comparator.comparing((Person::getAge))
                                .thenComparing(Person::getName)
                                .reversed());
persons.forEach(p1->System.out.print(" "+p1.getName()));
```

What will be the result?

A. Aman Tom Peter

B. Tom Aman Peter

C. Aman Peter Tom

D. Tom Peter Aman

Pregunta 3

Given:

var numbers = List.of(0,1,2,3,4,5,6,7,8,9);

You want to calculate the average of numbers.

Which two codes will accomplish this? (Choose two.)

A. double avg = numbers.stream().parallel().averagingDouble(a –> a);

B. double avg = numbers.parallelStream().mapToInt (m –> m).average().getAsDouble();

C. double avg = numbers.stream().mapToInt (i –> i).average().parallel();

D. double avg = numbers.stream().average().getAsDouble();

E. double avg = numbers.stream().collect(Collectors.averagingDouble(n –> n));

Pregunta 4

Assuming the Widget class has a getPrice method, this code does not compile:

```
List widgets = List.of(new Widget("Basic Widget", 19.55), // line 1
                       new Widget("Enhanced Widget", 35.00),
                       new Widget("Luxury Edition Widget", 55.45));
Stream widgetStream = widgets.stream();                    // line 4
widgetStream.filter(a -> a.getPrice() > 20.00)             // line 5
            .forEach(System.out::println);
```

Which two statements, independently, would allow this code to compile? (Choose two.)

A. Replace line 5 with widgetStream.filter(a –> ((Widget)a).getPrice() > 20.00).

B. Replace line 1 with List<Widget> widgetStream = widgets.stream();.

C. Replace line 5 with widgetStream.filter((Widget a) –> a.getPrice() > 20.00).

D. Replace line 4 with Stream<Widget> widgetStream = widgets.stream();.

Pregunta 5

Given

```java
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
                .flatMapToInt(IntStream::of)
                .map(n -> n + 1)
                .filter(n -> (n % 2 == 0))
                .peek(System.out::print)
                .count();
System.out.println(" " + count);
```

What is the result?

A. 6910 3

B. 10126 3

C. 3

D. 6104 3