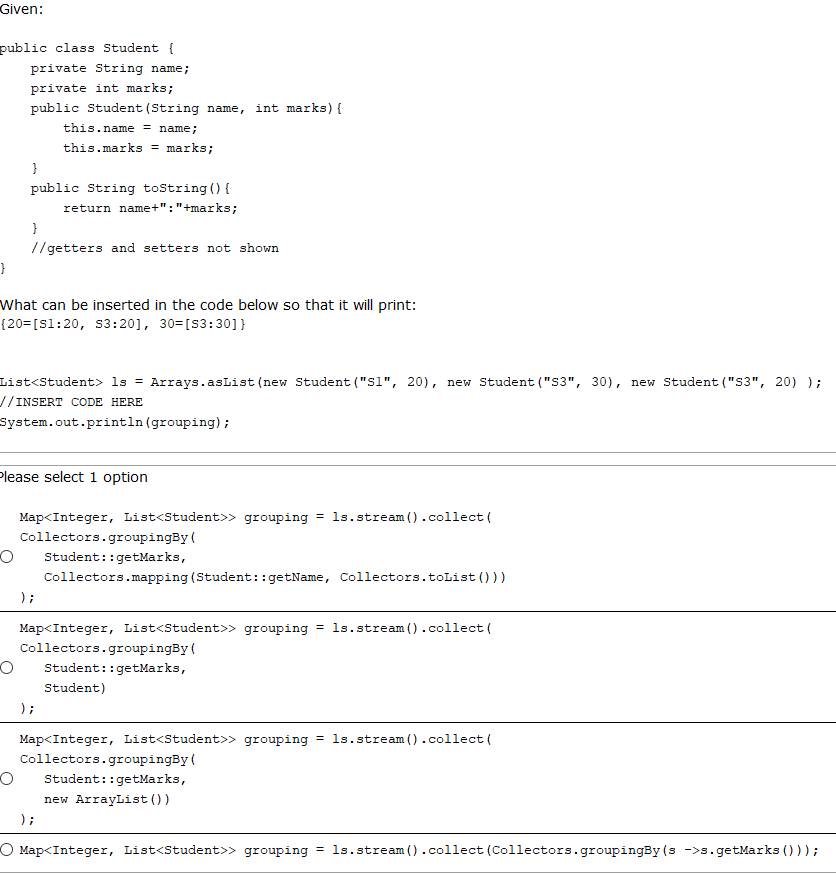
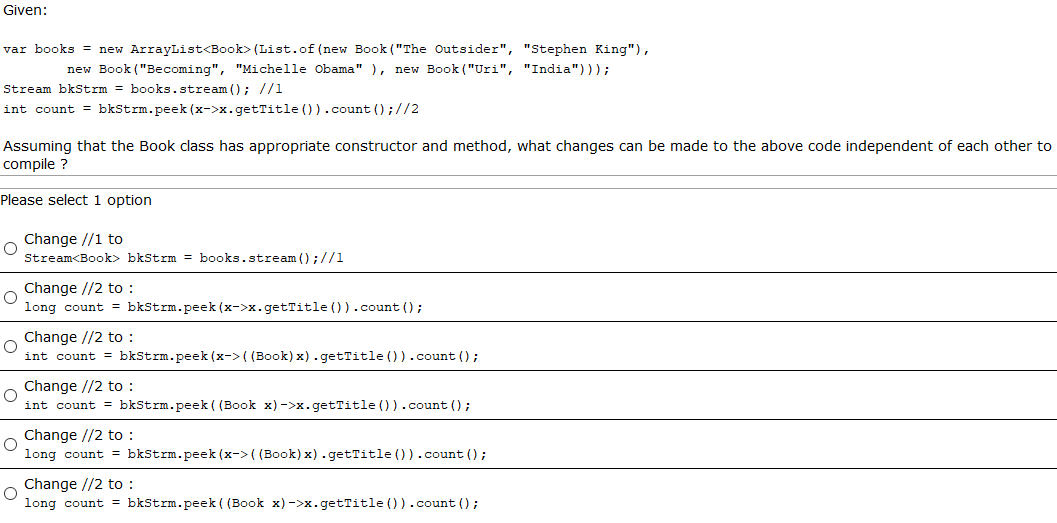
# Preguntas Objetivo 7

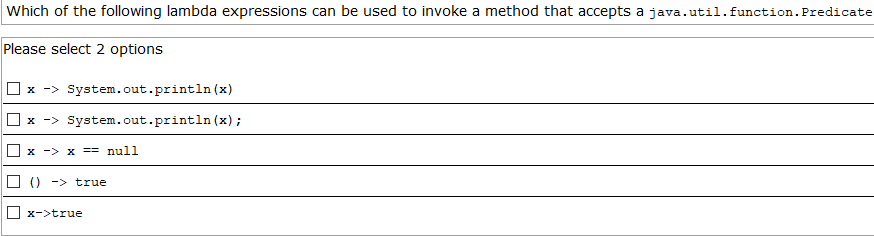
Pregunta 1



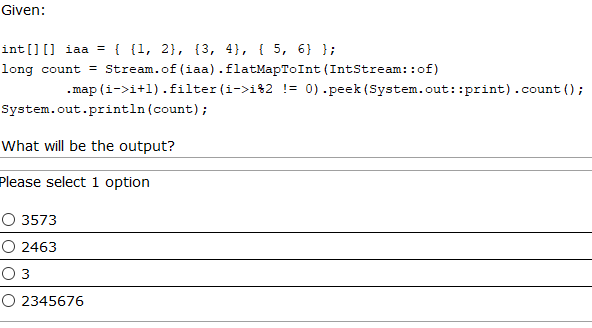
Pregunta 2



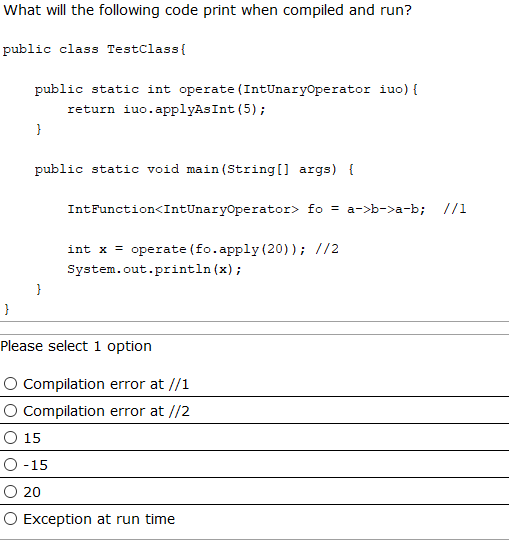
Pregunta 3



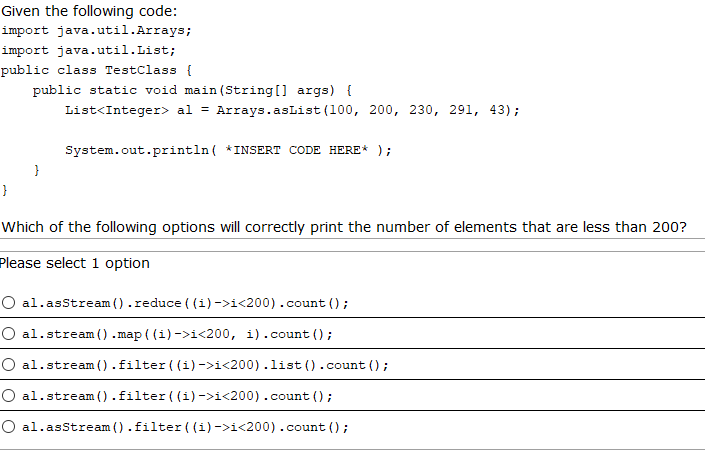
Pregunta 4



Pregunta 5

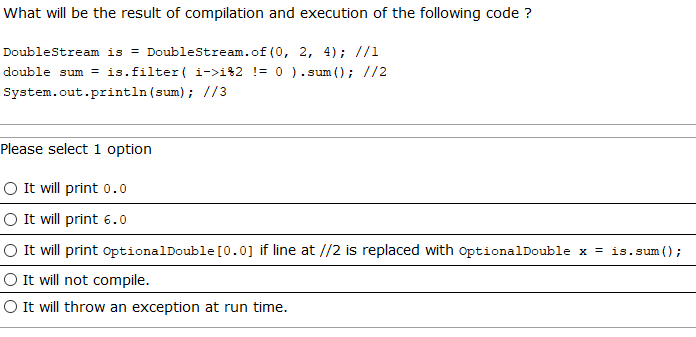


Pregunta 6

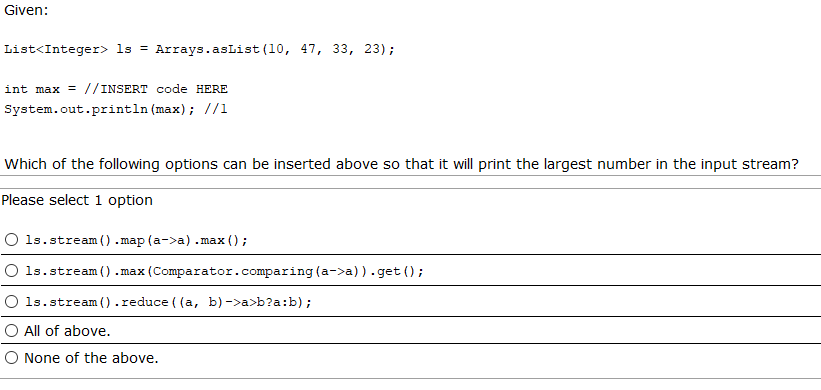


Respuesta:

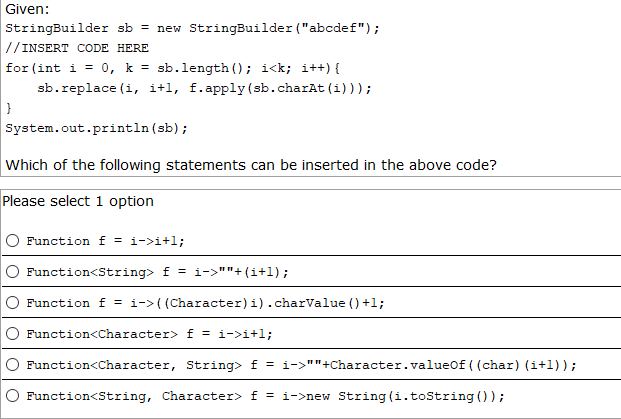
Pregunta 7



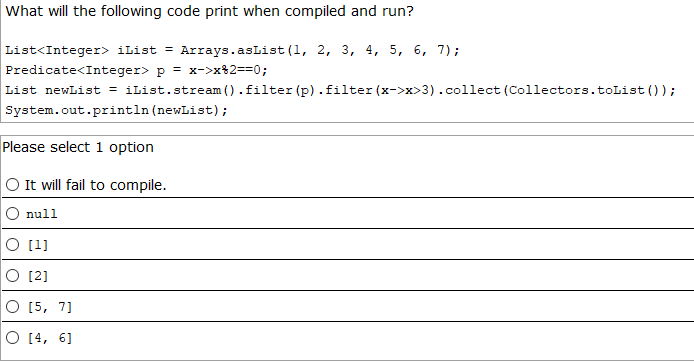
Pregunta 8



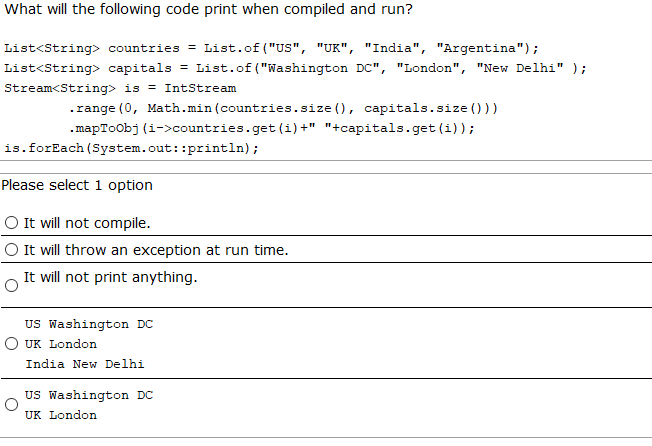
Pregunta 9



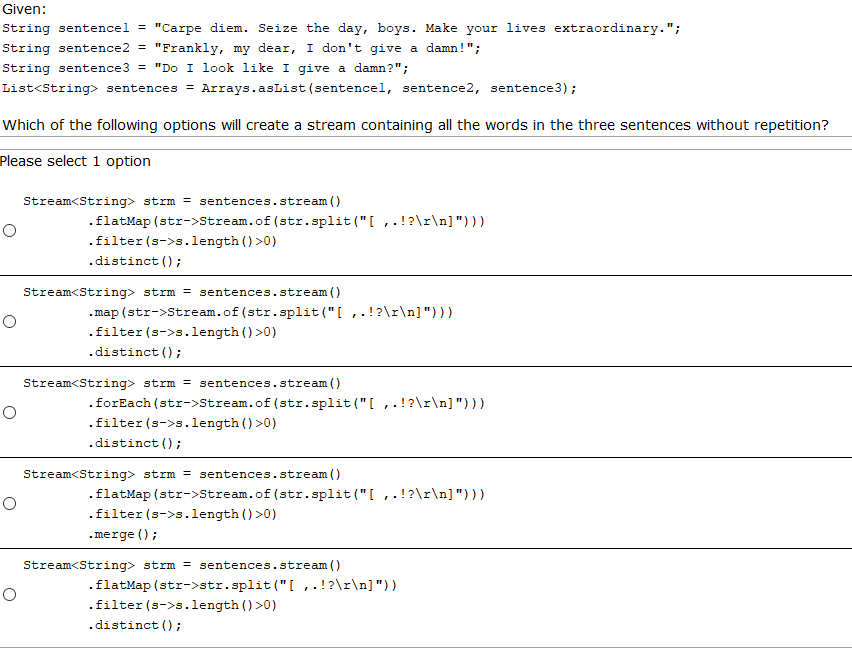
Pregunta 10



Pregunta 11

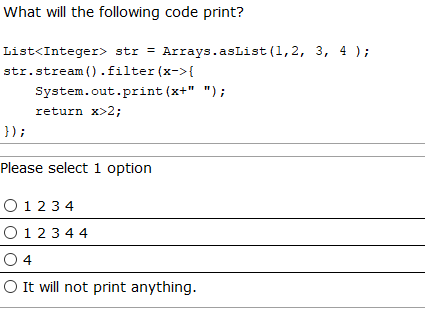


Pregunta 12



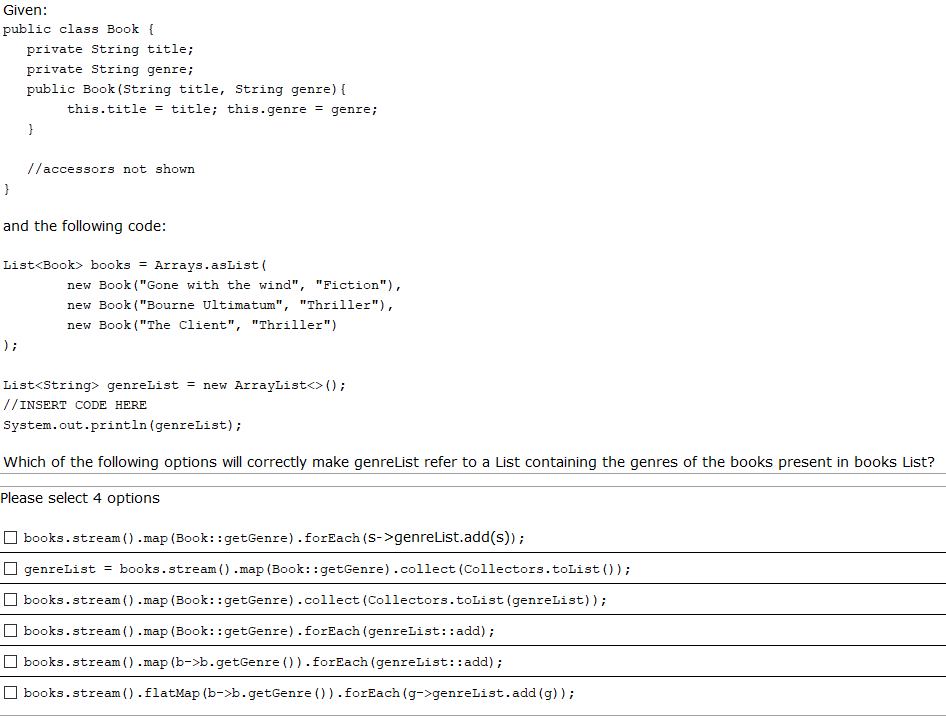
Respuesta:

Pregunta 13



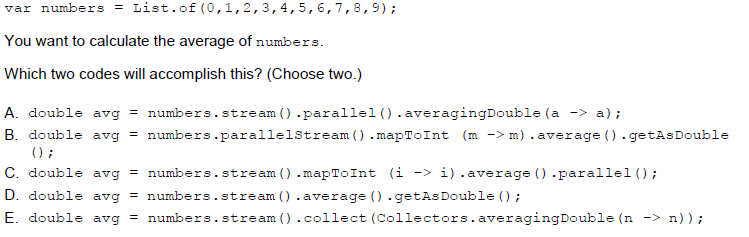
Respuesta:

Pregunta 14



Respuestas:

Pregunta 15



Respuestas:

Pregunta 16



Pregunta 17

What is the result of executing the following code?

Stream<Integer> numbers=Stream.*of*(11,-2,6,4,3,2,6,1,3);

System.***out***.println(Arrays.*binarySearch*(numbers

.parallel()

.distinct()

.sorted()

.toArray(a->**new** Integer[a]),6));

1. 0
2. -6
3. 5
4. -7
5. The result is unpredictable

Pregunta 18

Given:

Integer[] intArray = {7,9,10,5,2,9,7};

List<Integer> list = **new** ArrayList<>(Arrays.*asList*(intArray));

list.parallelStream()

.distinct()

.sequential()

.forEach(e->System.***out***.print(e+" "));

Which two are correct? (Choose two.)

A. The output will be exactly 7 9 10 5 2.

B. The program prints 7 9 10 5 2, but the order is unpredictable.

C. Replacing forEach() with forEachOrdered(), the program prints exactly 2 5 7 9 10.

D. Removing sequential(), the program prints 7 9 10 5 2 9 7, but the order is unpredictable.

E. Replacing forEach() with peek(), the program prints nothing.

Pregunta 19

Which functional interface complete the following code? (choose 2)

/\*complete here\*/ fun = String::new;

1. Consumer<Object>
2. Function<String>
3. UnaryOperator<String>
4. BiPredicate<String,Boolean>
5. Supplier<String>

Pregunta 20

Which of the following lambda expressions are correct? (choose 2)

1. n->return n\*5;
2. ()->10;
3. var c->System.out.println(c);
4. r->int x=r\*10;System.out.println(x);
5. (a,b,c)->(a+b).concat(c);