Question **1**

Not yet answered

Marked out of 1.00

In java, array elements are stored in ________ memory locations.

○ a. Sequential &amp; Random
● b. Sequential
○ c. Random
○ d. Binary search

Clear my choice

Question **2**

Not yet answered

Marked out of 1.00

Array data access using _____.

○ a. Variable
○ b. Operator
● c. index
○ d. Pointer

Clear my choice

Question 3

Not yet answered

Marked out of 1.00

Which of the following is an incorrect array declaration?

- ☐ a. int arr[] = new int[].
- ☑ b. int arr[] = int [5] new
- ☐ c. int [] arr = new int[5].
- ☐ d. int arr[] = new int[5].

Question 4

Not yet answered

Marked out of 1.00

Which of the following is an outdated class but still in use?

- ☑ a. Vector
- ☑ b. Hashtable
- ☐ c. Arraylist
- ☐ d. HashMap.

Question **5**

Not yet answered

Marked out of 1.00

The interface Comparable contains the method __________

○ a. compare

○ b. compareWith

○ c. toCompare

● d. compareTo

Clear my choice

Question **6**

Not yet answered

Marked out of 1.00

Which of the following Sets maintains the insertion order?

○ a. TreeSet

○ b. ArrayList

● c. LinkedHashSet

○ d. HashSet

Clear my choice

Question 7

Not yet answered

Marked out of 1.00

We can calculate the length of an ArrayList arrList using ________.

● a. arrList.size()

○ b. arrList.length

○ c. arrList.len

○ d. sizeof(arrList)

Clear my choice

Question 8

Not yet answered

Marked out of 1.00

Which class stores elements in ascending order?

○ a. HashSet

○ b. HashMap

○ c. ArrayList

● d. TreeSet

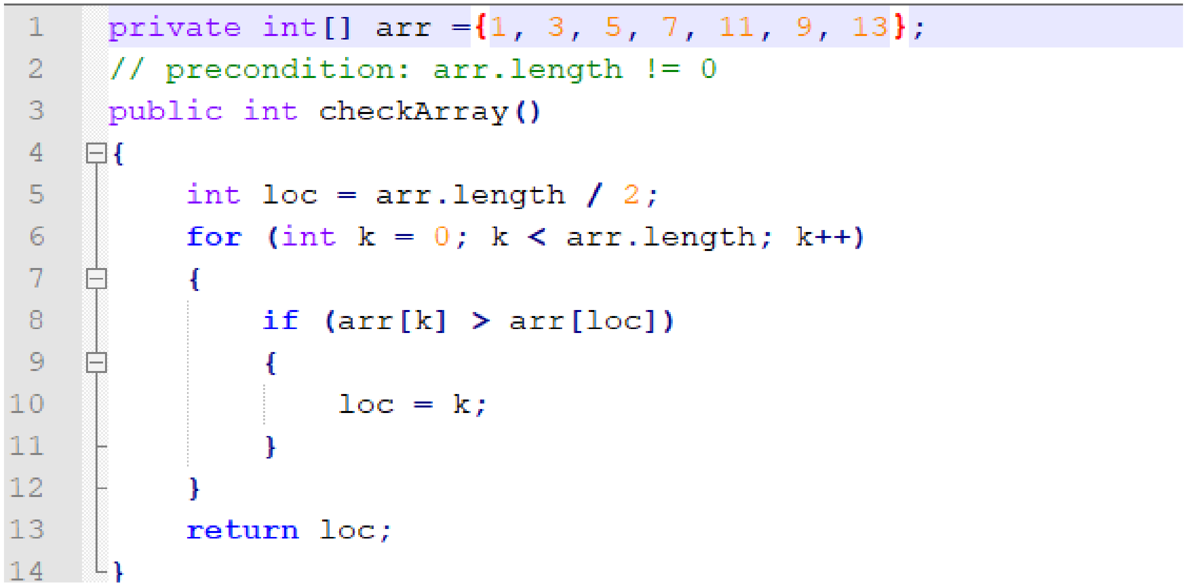
Clear my choice

Question 9

Not yet answered

Marked out of 1.00

Consider the following field arr and method checkArray. Which of the following best describes what checkArray returns?



```
private int[] arr ={1, 3, 5, 7, 11, 9, 13};
// precondition: arr.length != 0
public int checkArray()
{
    int loc = arr.length / 2;
    for (int k = 0; k < arr.length; k++)
    {
        if (arr[k] > arr[loc])
        {
            loc = k;
        }
    }
    return loc;
}
```

● a. Returns the index of the largest value in array arr

○ b. Returns the index of the first element in array arr whose value is greater than arr[loc].

○ c. Returns the index of the last element in array arr whose value is greater than arr[loc].

○ d. Returns the largest value in array arr.

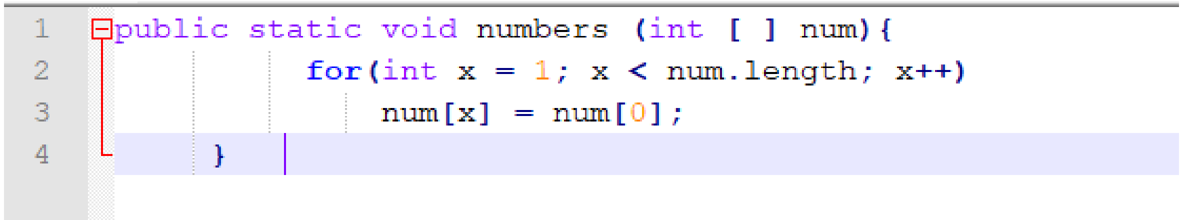
Clear my choice

Question **10**

Not yet answered

Marked out of 1.00

What does the following method do?



```
public static void numbers (int [ ] num){
        for(int x = 1; x < num.length; x++)
            num[x] = num[0];
    }
```

○ a. It copies 0 into every element.

○ b. It copies what is in cell 0 into all other cells.

● c. It changes every element to the value of x

○ d. It puts the array into numerical ascending order.

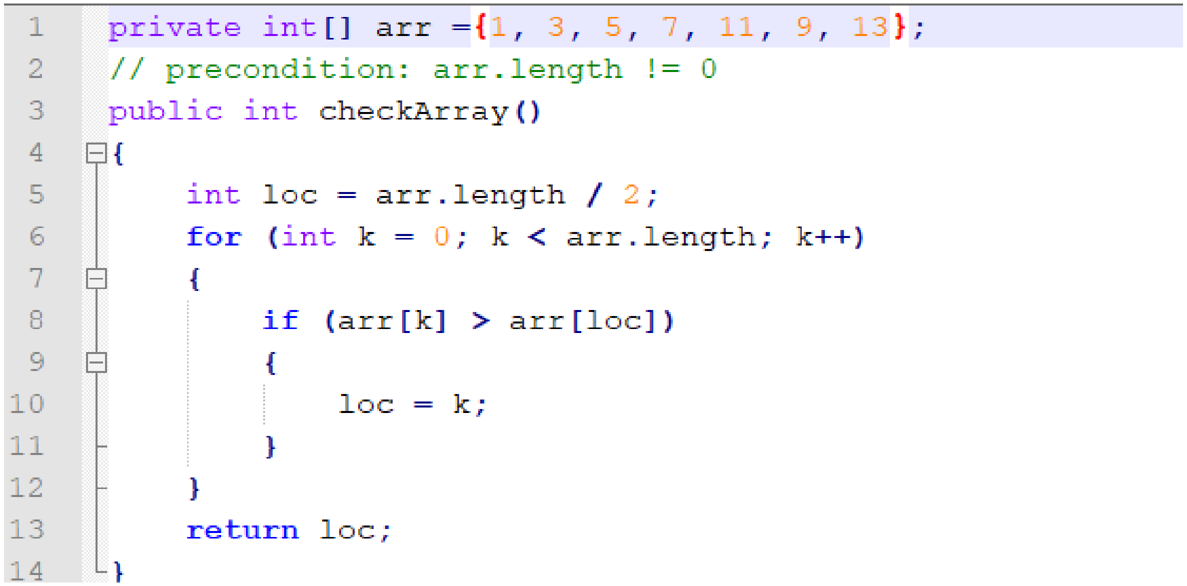
Clear my choice

Question **11**

Not yet answered

Marked out of 1.00

Consider the following field arr and method checkArray. Which of the following best describes what checkArray returns?



```
private int[] arr ={1, 3, 5, 7, 11, 9, 13};
// precondition: arr.length != 0
public int checkArray()
{
    int loc = arr.length / 2;
    for (int k = 0; k < arr.length; k++)
    {
        if (arr[k] > arr[loc])
        {
            loc = k;
        }
    }
    return loc;
}
```

○ a. 13

● b. 3

○ c. 7

○ d. 4

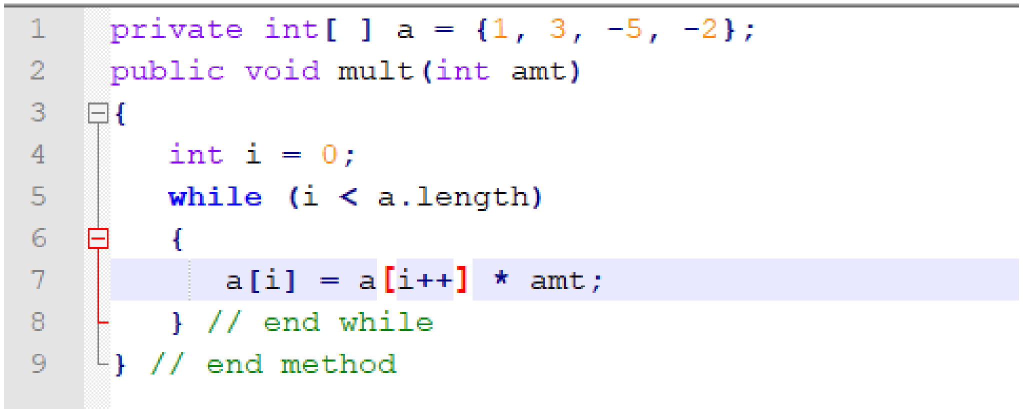
Clear my choice

Question **12**

Not yet answered

Marked out of 1.00

What are the values in a after mult(2) executes?



```
private int[ ] a = {1, 3, -5, -2};
public void mult(int amt)
{
    int i = 0;
    while (i < a.length)
    {
        a[i] = a[i++] * amt;
    } // end while
} // end method
```

○ a. {2, 6, -10, -4}

● b. The code will never stop executing due to an infinite loop

○ c. {1, 3, -5, -2}

○ d. {3, 9, -15, -6}

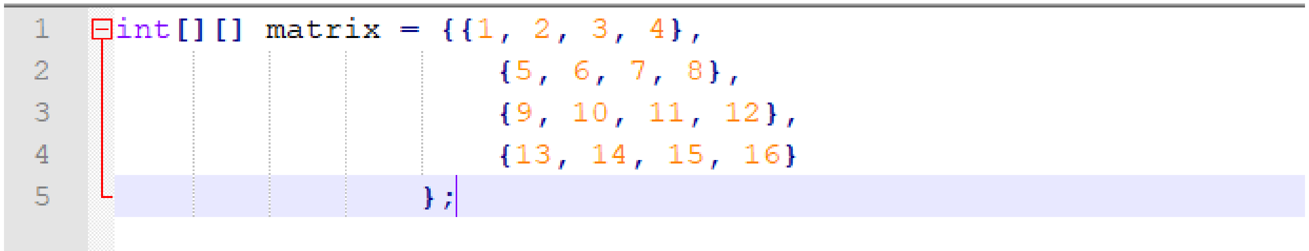
Clear my choice

Question **13**

Not yet answered

Marked out of 1.00

What number is in matrix[2][2]?



```
int[][] matrix = {{1, 2, 3, 4},
                    {5, 6, 7, 8},
                    {9, 10, 11, 12},
                    {13, 14, 15, 16}
                };
```

○ a. 10

○ b. 11

● c. 6

○ d. 7

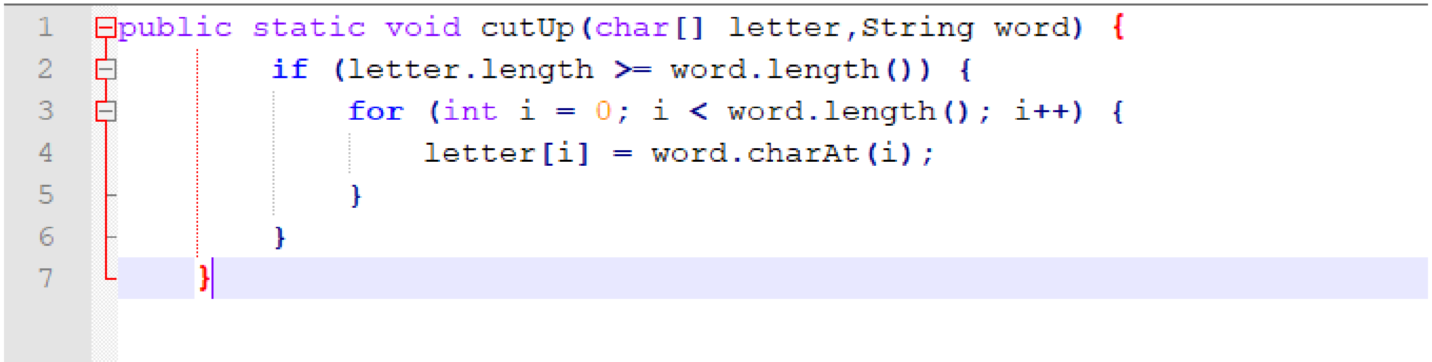
Clear my choice

Question **14**

Not yet answered

Marked out of 1.00

What does this method do?



```
public static void cutUp(char[] letter,String word) {
        if (letter.length >= word.length()) {
            for (int i = 0; i < word.length(); i++) {
                letter[i] = word.charAt(i);
            }
        }
    }
```

● a. It creates a new array that holds the letters from the String.

○ b. It copies letters from the array to the String.

○ c. If there are enough cells in the array, it copies each letter from the String to the array.

○ d. It checks to see if the array contains the same letters as the String.

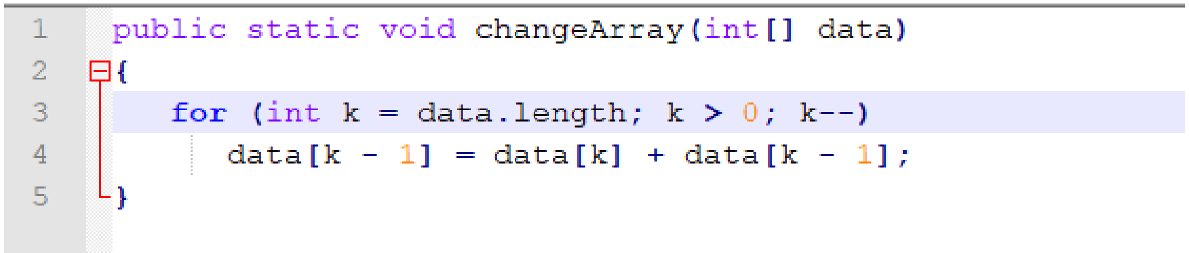
Clear my choice

Question **15**

Not yet answered

Marked out of 1.00

Consider the following method changeArray. An array is created that contains {2, 8, 10, 9, 6} and is passed to changeArray. What are the contents of the array after the changeArray method executes?



```
public static void changeArray(int[] data)
{
    for (int k = data.length; k > 0; k--)
        data[k - 1] = data[k] + data[k - 1];
}
```

○ a. {10, 18, 19, 15, 6}

● b. {35, 33, 25, 15, 6}

○ c. This method results in an IndexOutOfBounds exception.

○ d. {2, 6, 2, -1, -3}

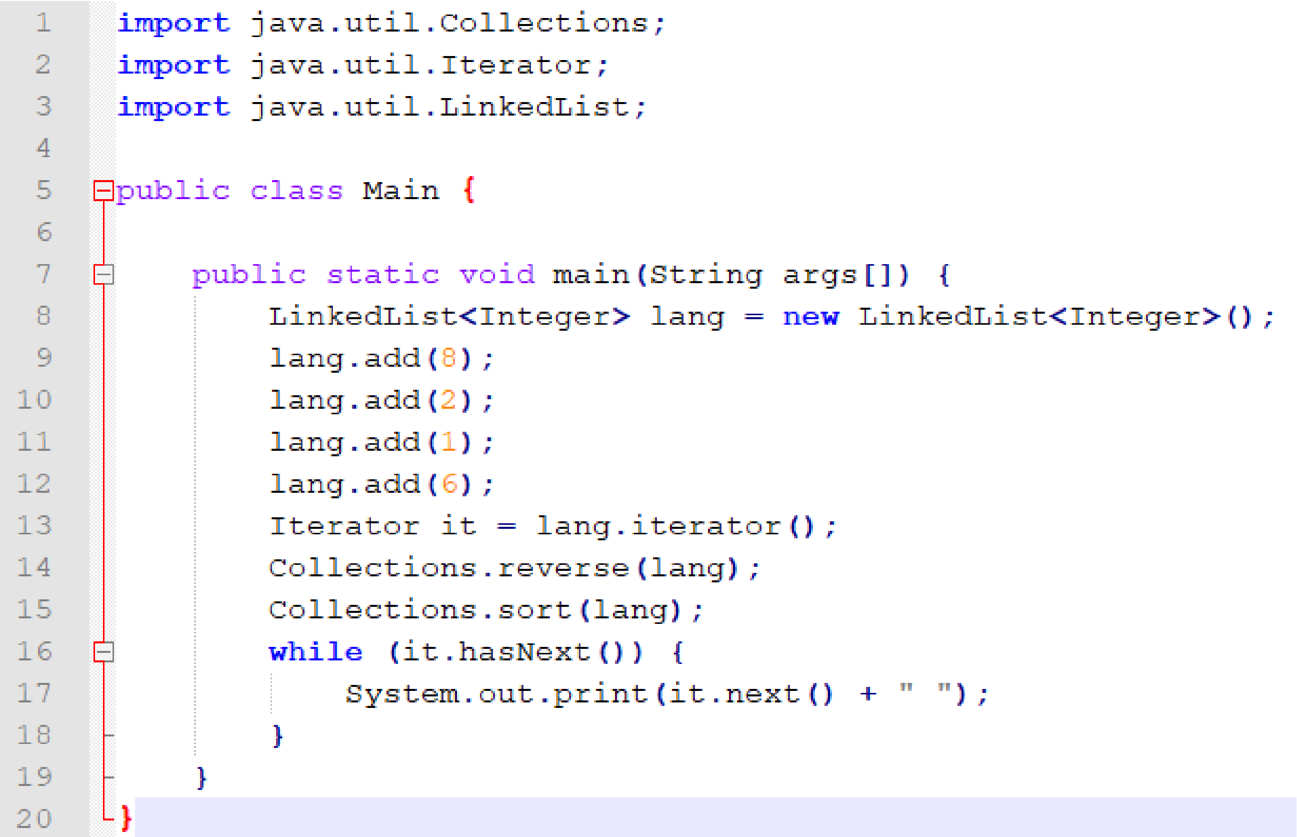
Clear my choice

Question **16**

Not yet answered

Marked out of 1.00

What is the output of the following code?



```
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;

public class Main {

    public static void main(String args[]) {
        LinkedList<Integer> lang = new LinkedList<Integer>();
        lang.add(8);
        lang.add(2);
        lang.add(1);
        lang.add(6);
        Iterator it = lang.iterator();
        Collections.reverse(lang);
        Collections.sort(lang);
        while (it.hasNext()) {
            System.out.print(it.next() + " ");
        }
    }
}
```

○ a. 8 6 2 1

○ b. 6 1 2 8

○ c. 8 2 1 6

● d. 1 2 6 8

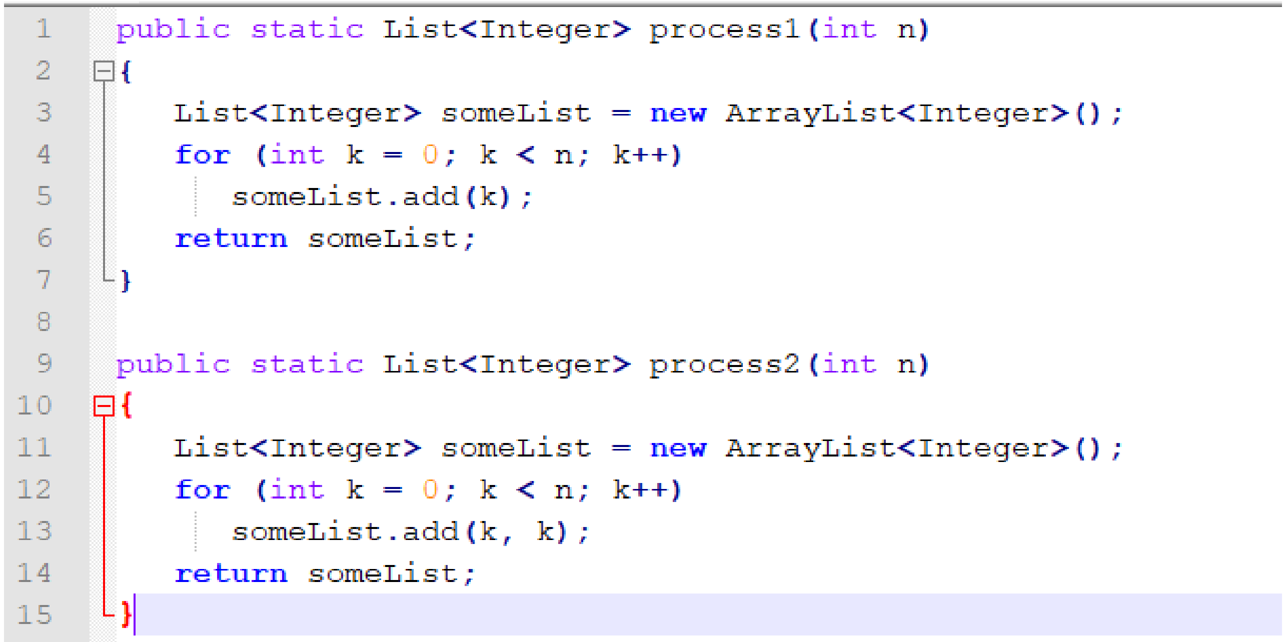
Clear my choice

Question **17**

Not yet answered

Marked out of 1.00

Which of the following best describes the behavior of process1 and process2 (shown below)?



```java
public static List<Integer> process1(int n)
{
    List<Integer> someList = new ArrayList<Integer>();
    for (int k = 0; k < n; k++)
        someList.add(k);
    return someList;
}

public static List<Integer> process2(int n)
{
    List<Integer> someList = new ArrayList<Integer>();
    for (int k = 0; k < n; k++)
        someList.add(k, k);
    return someList;
}
```

○ a. Both methods produce the same result and take the same amount of time.

● b. The two methods produce different results, and process1 is faster than process2.

○ c. The two methods produce different results, and process2 is faster than process1.

○ d. Both methods produce the same result, and process1 is faster than process2.

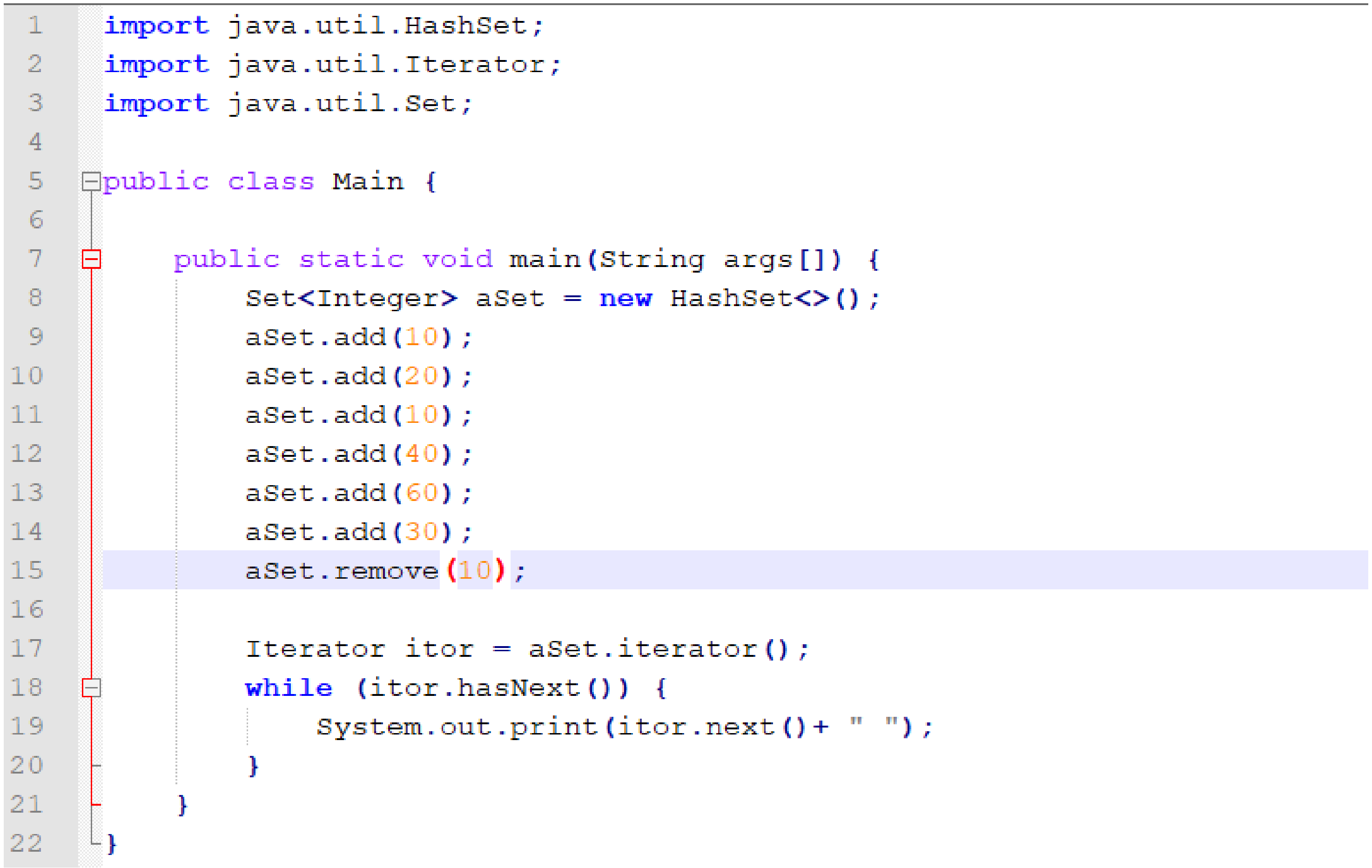
Clear my choice

Question 18

Not yet answered

Marked out of 1.00

What is printed as a result of executing the following code segment?



```java
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class Main {

    public static void main(String args[]) {
        Set<Integer> aSet = new HashSet<>();
        aSet.add(10);
        aSet.add(20);
        aSet.add(10);
        aSet.add(40);
        aSet.add(60);
        aSet.add(30);
        aSet.remove(10);

        Iterator itor = aSet.iterator();
        while (itor.hasNext()) {
            System.out.print(itor.next()+ " ");
        }
    }
}
```

○ a. 10 20 30 40 60

○ b. 20 10 40 60 30

● c. Compiler error

○ d. 20 40 60 30

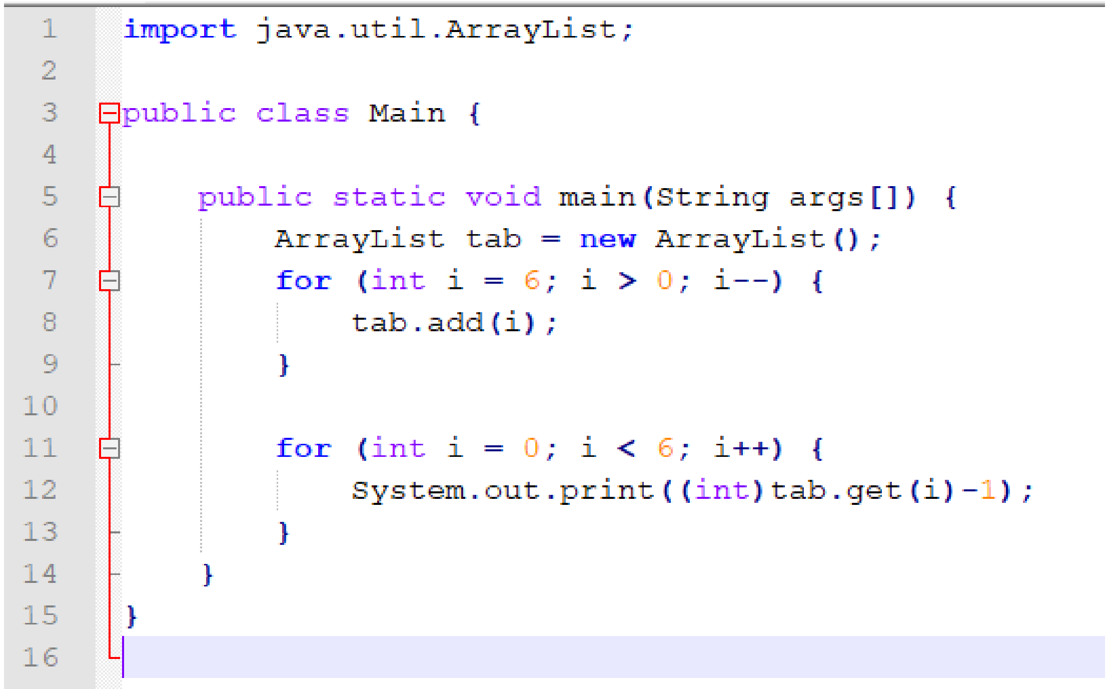
Clear my choice

Question **19**

Not yet answered

Marked out of 1.00

What is the output of the following code?



```
import java.util.ArrayList;

public class Main {

    public static void main(String args[]) {
        ArrayList tab = new ArrayList();
        for (int i = 6; i > 0; i--) {
            tab.add(i);
        }

        for (int i = 0; i < 6; i++) {
            System.out.print((int)tab.get(i)-1);
        }
    }
}

```

○ a. 611110

● b. 543210

○ c. 600001

○ d. 654321

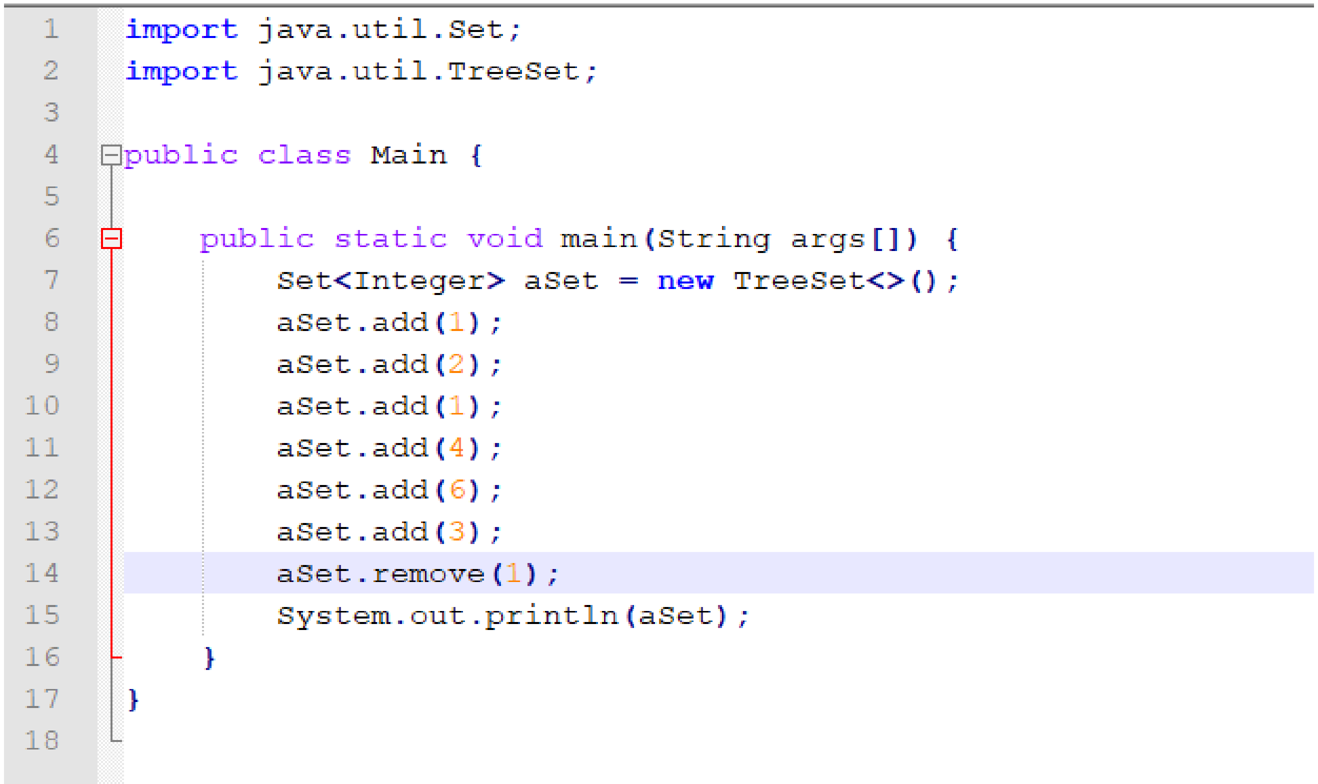
Clear my choice

Question **20**

Not yet answered

Marked out of 1.00

What is printed as a result of executing the following code segment?



```
import java.util.Set;
import java.util.TreeSet;

public class Main {

    public static void main(String args[]) {
        Set<Integer> aSet = new TreeSet<>();
        aSet.add(1);
        aSet.add(2);
        aSet.add(1);
        aSet.add(4);
        aSet.add(6);
        aSet.add(3);
        aSet.remove(1);
        System.out.println(aSet);
    }
}

```

○ a. Compiler error

○ b. [1, 2, 3, 4, 6]

● c. [2, 1, 4, 6, 3]

○ d. [2, 3, 4, 6]

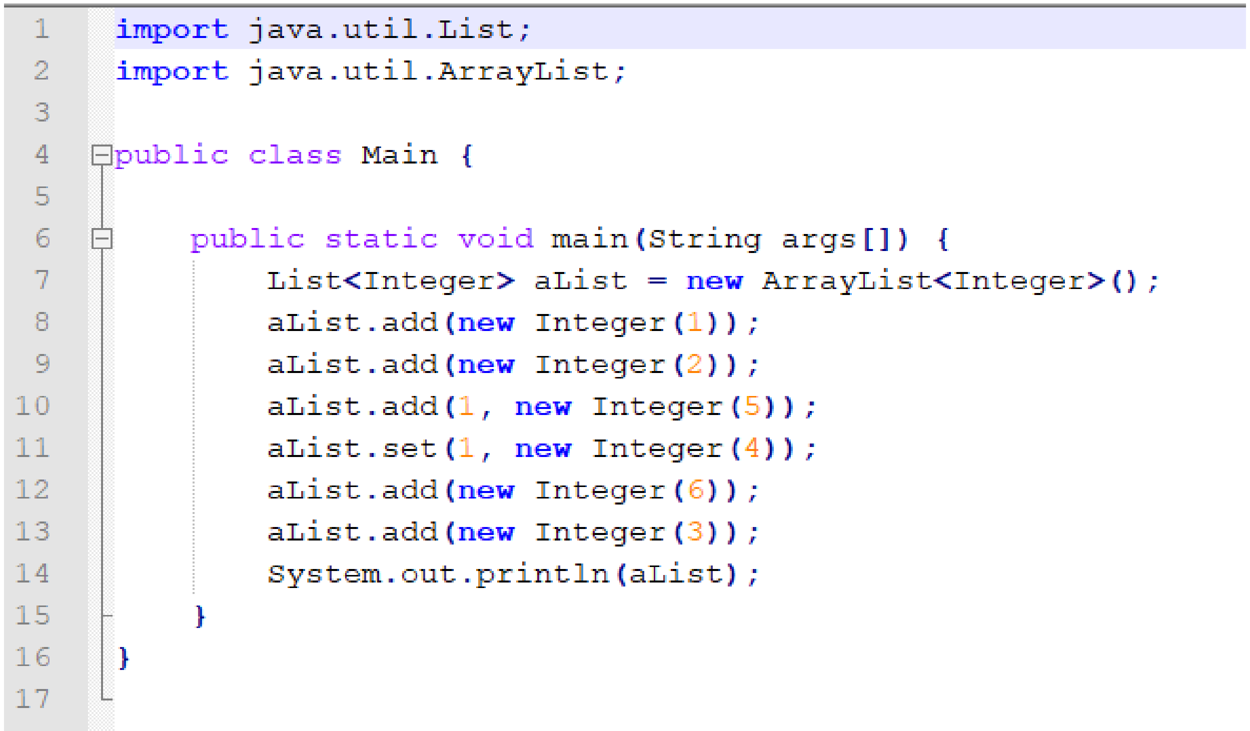
Clear my choice

Question **21**

Not yet answered

Marked out of 1.00

What is printed as a result of executing the following code segment?



```java
import java.util.List;
import java.util.ArrayList;

public class Main {

    public static void main(String args[]) {
        List<Integer> aList = new ArrayList<Integer>();
        aList.add(new Integer(1));
        aList.add(new Integer(2));
        aList.add(1, new Integer(5));
        aList.set(1, new Integer(4));
        aList.add(new Integer(6));
        aList.add(new Integer(3));
        System.out.println(aList);
    }
}

```

● a. [1, 2, 5, 4, 6, 3]

○ b. [1, 4, 2, 6, 3]

○ c. [1, 2, 3, 4, 5, 6]

○ d. Compiler error

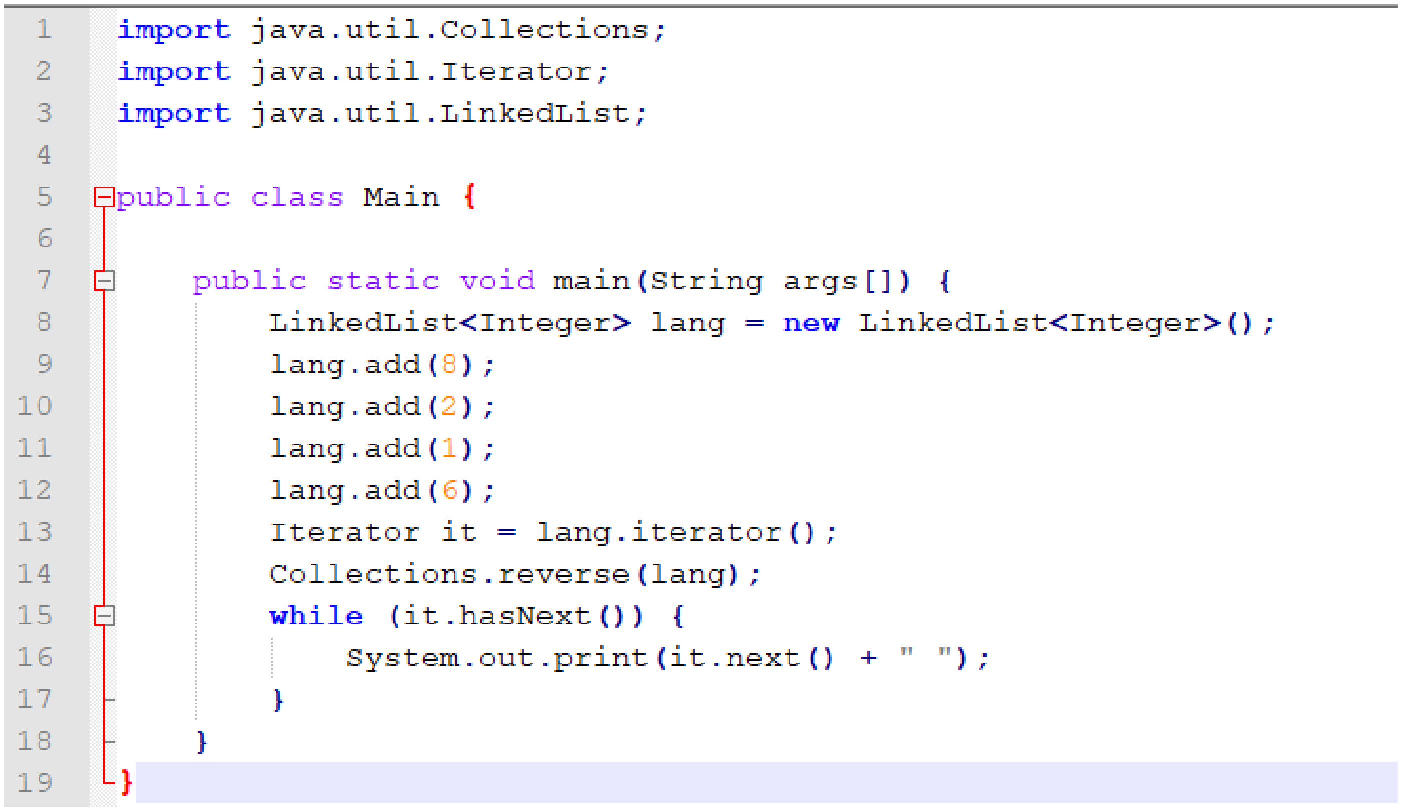
Clear my choice

Question **22**

Not yet answered

Marked out of 1.00

What is the output of the following code?



```
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;

public class Main {

    public static void main(String args[]) {
        LinkedList<Integer> lang = new LinkedList<Integer>();
        lang.add(8);
        lang.add(2);
        lang.add(1);
        lang.add(6);
        Iterator it = lang.iterator();
        Collections.reverse(lang);
        while (it.hasNext()) {
            System.out.print(it.next() + " ");
        }
    }
}
```

○ a. 8 2 1 6

○ b. 8 6 2 1

● c. 1 2 6 8

○ d. 6 1 2 8

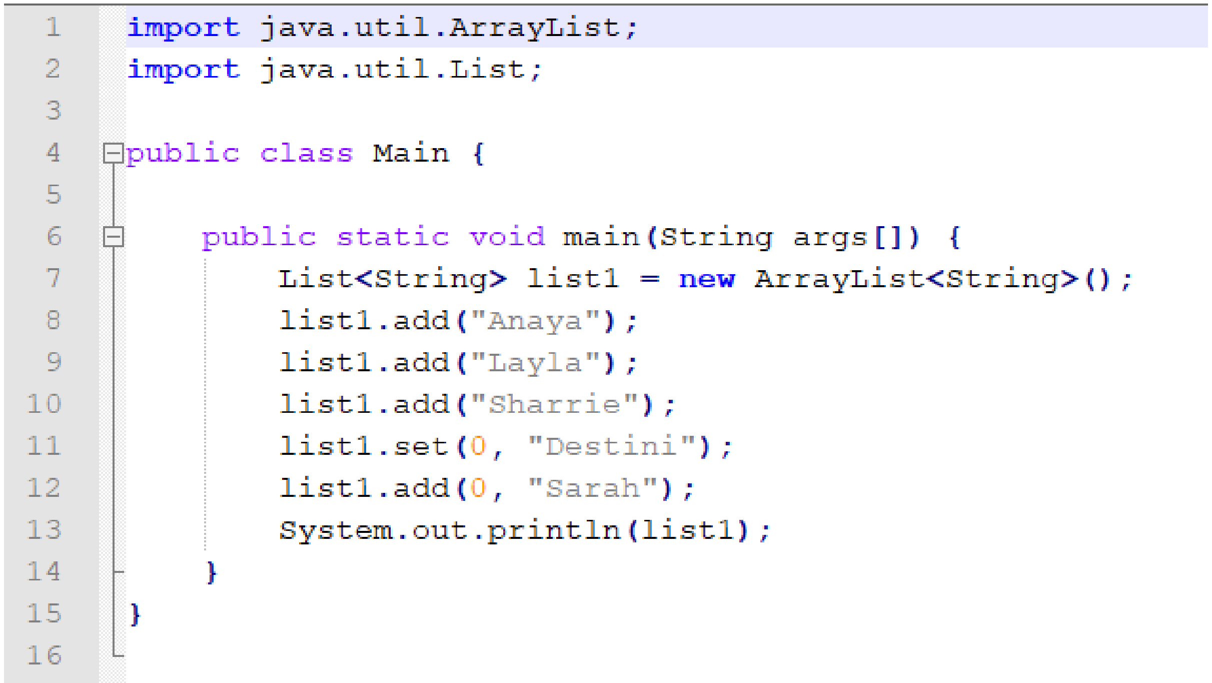
Clear my choice

Question **23**

Not yet answered

Marked out of 1.00

What will print when the following code executes?



```
import java.util.ArrayList;
import java.util.List;

public class Main {

    public static void main(String args[]) {
        List<String> list1 = new ArrayList<String>();
        list1.add("Anaya");
        list1.add("Layla");
        list1.add("Sharrie");
        list1.set(0, "Destini");
        list1.add(0, "Sarah");
        System.out.println(list1);
    }
}

```

○ a. [Destini, Layla, Sharrie, Sarah]

● b. [Sarah, Destini, Anaya, Layla, Sharrie]

○ c. [Sarah, Destini, Layla, Sharrie]

○ d. [Sarah, Layla, Sharrie]

Clear my choice

Question **24**

Not yet answered

Marked out of 1.00

Collection ____________

○ a. implements the Traversable interface

○ b. implements the Serializable interface

○ c. inherits the Iterable interface

● d. inherits the Collections class

Clear my choice

Question **25**

Not yet answered

Marked out of 1.00

Which implementation of Iterator can traverse a collection back and forth?

○ a. MapIterator

● b. ListIterator

○ c. SetIterator

○ d. Iterator

Clear my choice

«

»