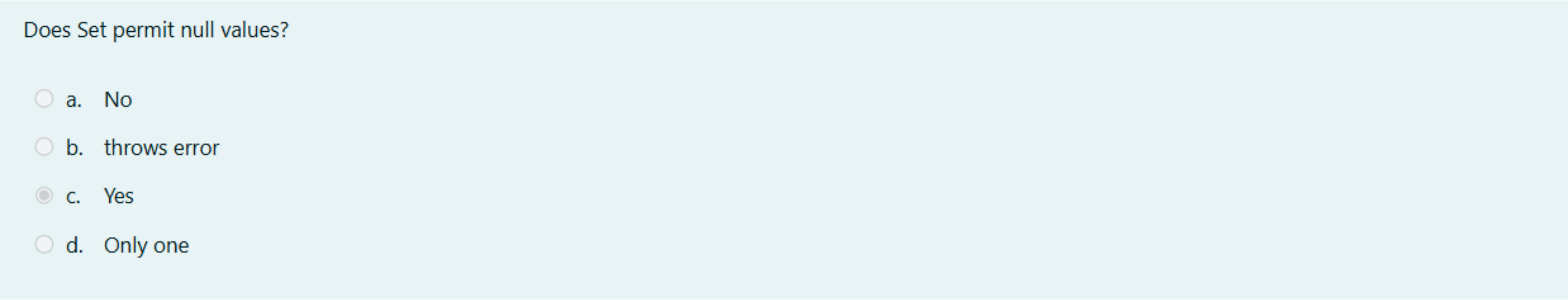
10. COLLECTION- LIST



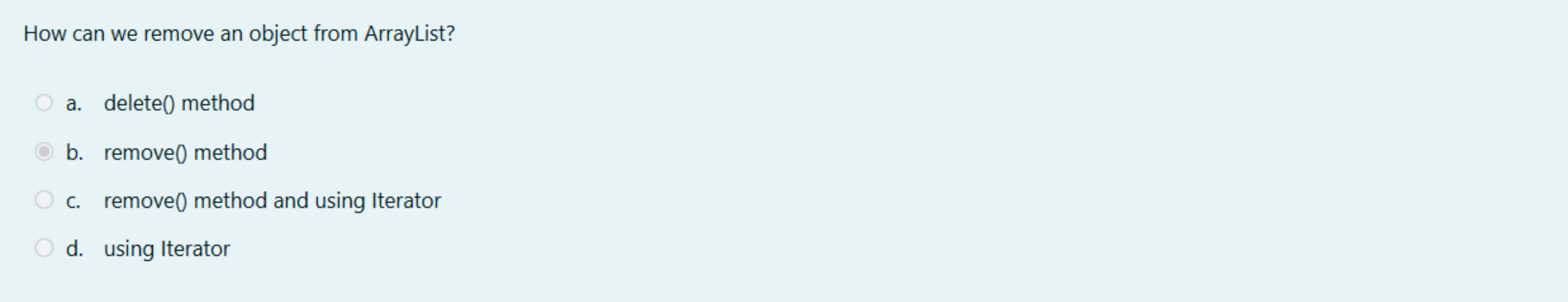
Does Set permit null values?

○ a. No

○ b. throws error

● c. Yes

○ d. Only one



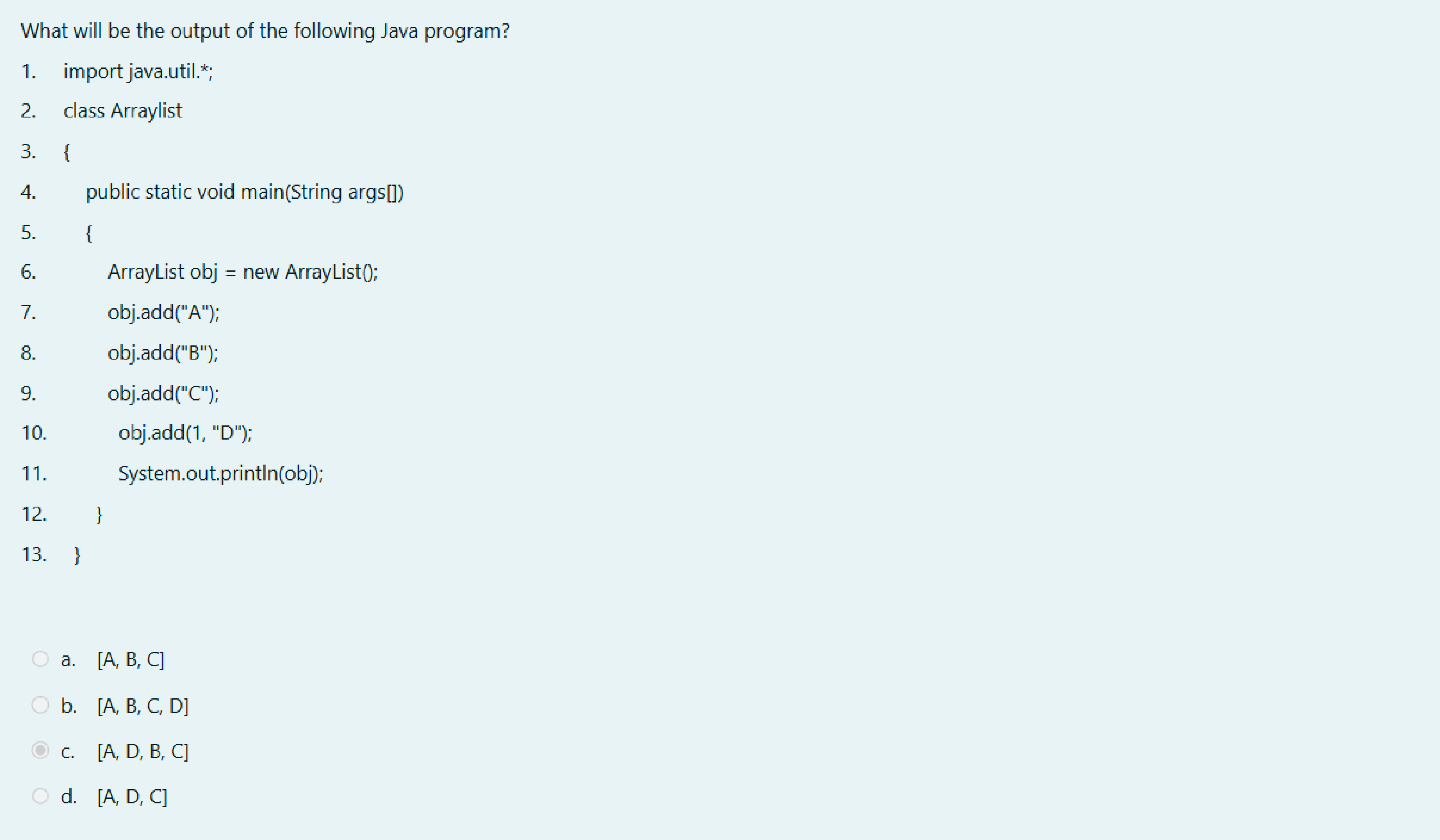
How can we remove an object from ArrayList?

○ a. delete() method

● b. remove() method

○ c. remove() method and using Iterator

○ d. using Iterator



What will be the output of the following Java program?

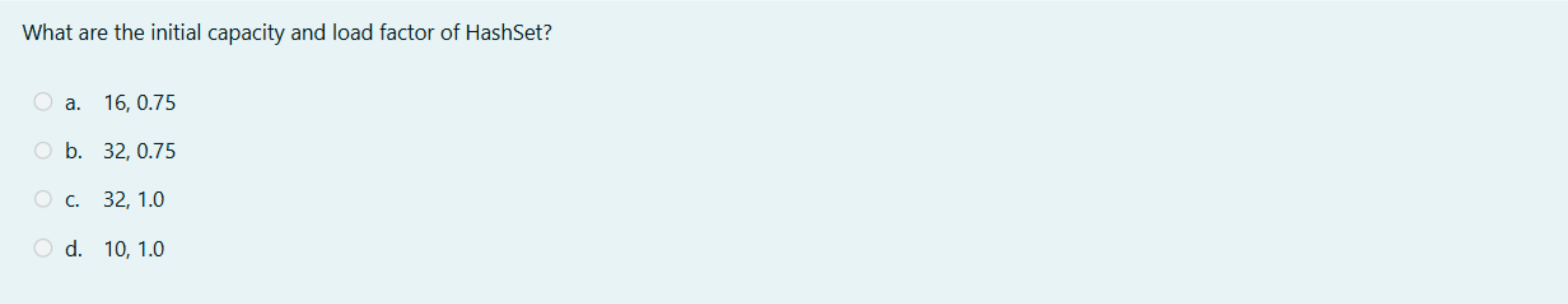
```
import java.util.*;
class Arraylist
{
    public static void main(String args[])
    {
        ArrayList obj = new ArrayList();
        obj.add("A");
        obj.add("B");
        obj.add("C");
          obj.add(1, "D");
          System.out.println(obj);
      }
  }
```

○ a. [A, B, C]

○ b. [A, B, C, D]

● c. [A, D, B, C]

○ d. [A, D, C]



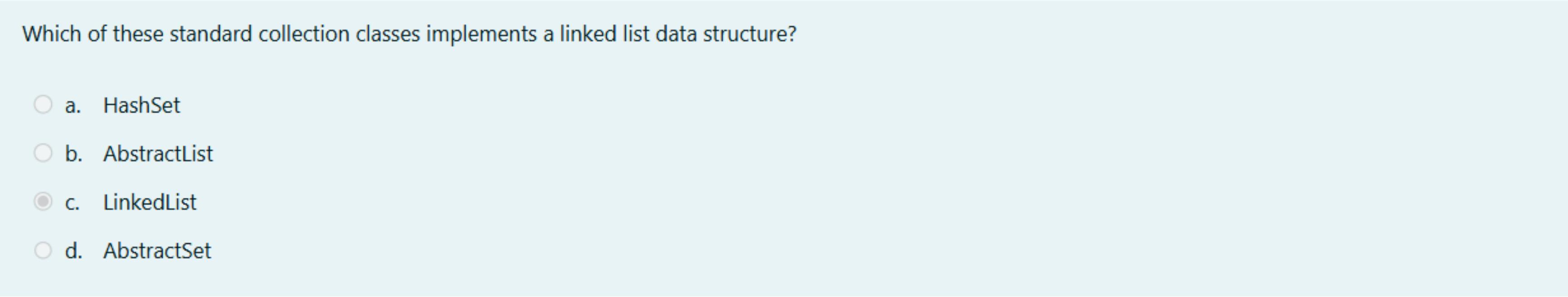
What are the initial capacity and load factor of HashSet?

○ a. 16, 0.75

○ b. 32, 0.75

○ c. 32, 1.0

○ d. 10, 1.0



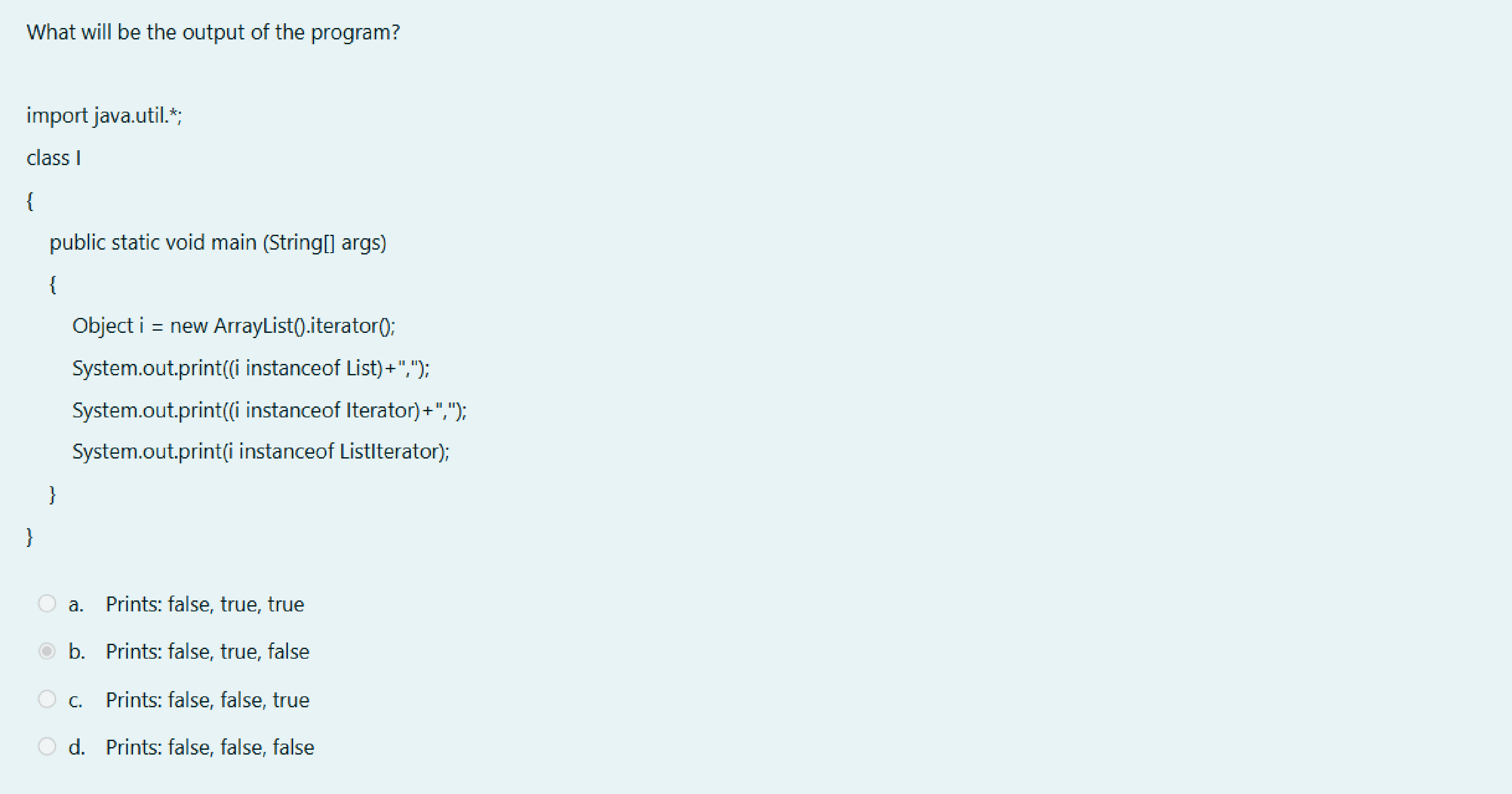
Which of these standard collection classes implements a linked list data structure?

○ a. HashSet

○ b. AbstractList

● c. LinkedList

○ d. AbstractSet



What will be the output of the program?

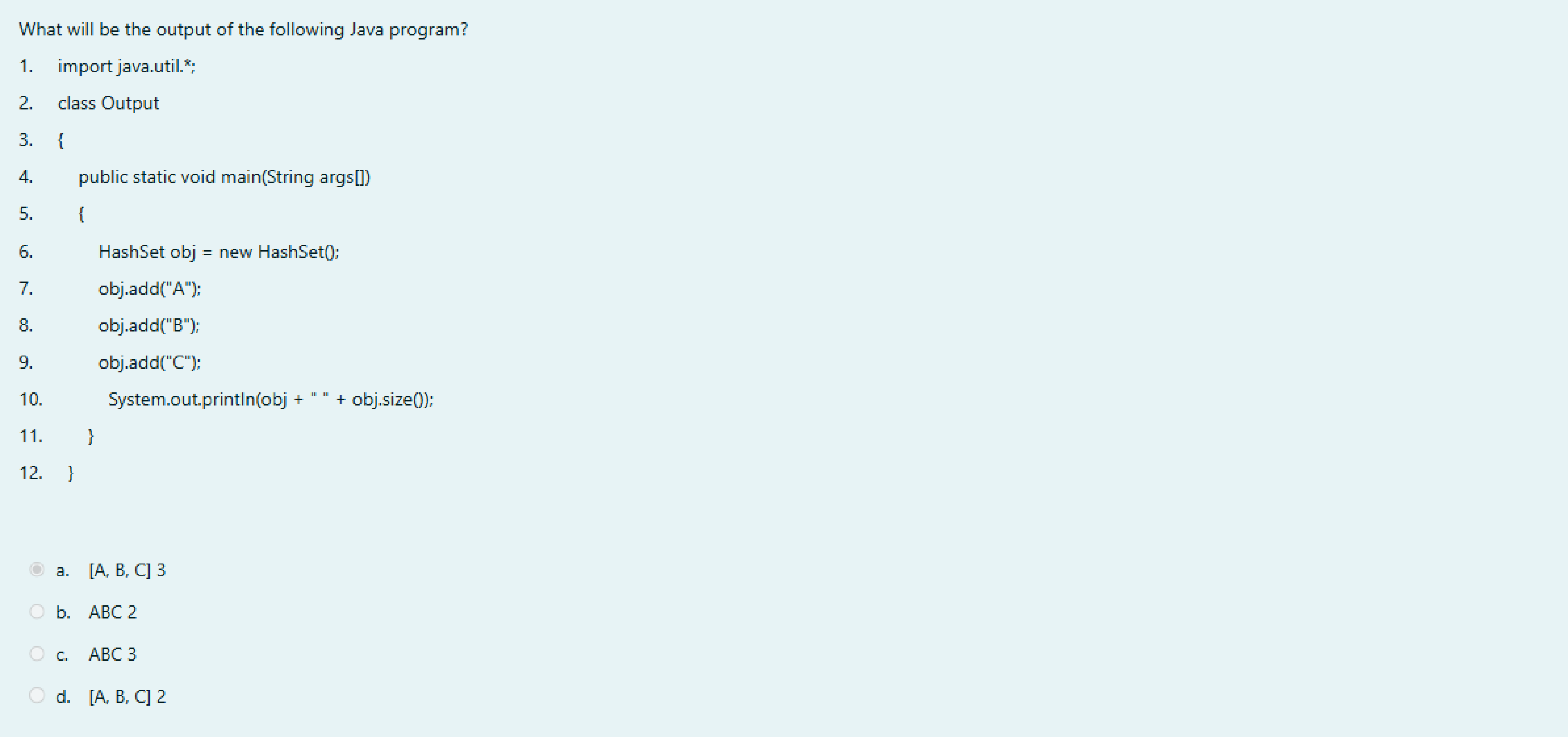
```
import java.util.*;
class I
{
    public static void main (String[] args)
    {
        Object i = new ArrayList().iterator();
        System.out.print((i instanceof List)+",");
        System.out.print((i instanceof Iterator)+",");
        System.out.print(i instanceof ListIterator);
    }
}
```

○ a. Prints: false, true, true

● b. Prints: false, true, false

○ c. Prints: false, false, true

○ d. Prints: false, false, false



What will be the output of the following Java program?

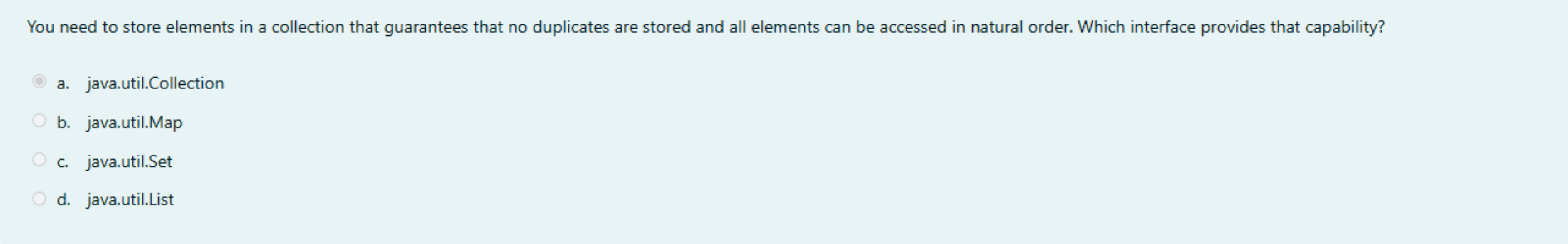
```
import java.util.*;
class Output
{
    public static void main(String args[])
    {
        HashSet obj = new HashSet();
        obj.add("A");
        obj.add("B");
        obj.add("C");
         System.out.println(obj + " " + obj.size());
    }
}
```

● a. [A, B, C] 3

○ b. ABC 2

○ c. ABC 3

○ d. [A, B, C] 2



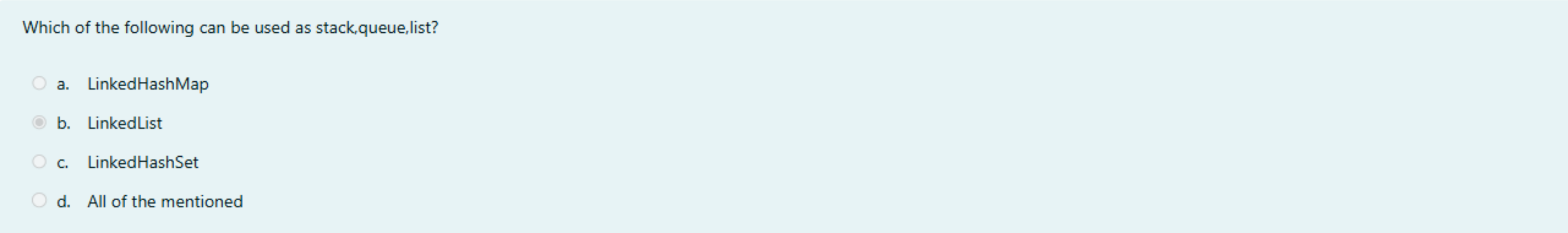
You need to store elements in a collection that guarantees that no duplicates are stored and all elements can be accessed in natural order. Which interface provides that capability?

● a. java.util.Collection

○ b. java.util.Map

○ c. java.util.Set

○ d. java.util.List



Which of the following can be used as stack,queue,list?

○ a. LinkedHashMap

● b. LinkedList

○ c. LinkedHashSet

○ d. All of the mentioned

What will be the output of the following Java program?

```
import java.util.*;
class Output
{
    public static void main(String args[])
    {
        TreeSet t = new TreeSet();
        t.add("3");
        t.add("9");
        t.add("1");
        t.add("4");
        t.add("8");
        System.out.println(t);
    }
}
```

○ a. [9, 8, 4, 3, 1]

○ b. [3, 4, 1, 8, 9]

● c. [1, 3, 4, 8, 9]

○ d. [1, 3, 5, 8, 9]

Which of this method is used to change an element in a LinkedList Object?

○ a. add()

○ b. change()

● c. set()

○ d. redo()

What differentiates a circular linked list from a normal linked list?

○ a. Head node is known in circular linked list

○ b. You may or may not have the 'next' pointer point to null in a circular linked list

○ c. It is faster to traverse the circular linked list

● d. You cannot have the 'next' pointer point to null in a circular linked list

Which of these classes implements Set interface?

○ a. LinkedList

○ b. DynamicList

● c. HashSet

○ d. ArrayList

What is the unique feature of LinkedHashSet?

○ a. The elements in the collection are linked to each other

○ b. It is not a valid class

● c. It maintains the insertion order and guarantees uniqueness

○ d. It provides a way to store key values with uniqueness

What will be the output of the following Java code snippet?

```
import java.util.*;
class Linkedlist
{
    public static void main(String args[])
    {
        LinkedList obj = new LinkedList();
        obj.add("A");
        obj.add("B");
        obj.add("C");
        obj.addFirst("D");
        System.out.println(obj);
    }
}
```

○ a. [A, B, C]

○ b. [D, B, C]

● c. [D, A, B, C]

○ d. [A, B, C, D]

How to remove duplicates from List?

○ a. HashSet<String> listToSet = duplicateList.toSet();

○ b. HashSet<String> listToSet = duplicateList.getSet();

● c. HashSet<String> listToSet = new HashSet<String>(duplicateList);

○ d. HashSet<String> listToSet = Collections.convertToSet(duplicateList);

Which of these method of HashSet class is used to add elements to its object?

● a. add()

○ b. insert()

○ c. Add()

○ d. addFirst()

What will be the output of the following Java program?

```
import java.util.*;
class Output
{
    public static void main(String args[])
    {
        ArrayList obj = new ArrayList();
        obj.add("A");
        obj.add(0, "B");
        System.out.println(obj.size());
    }
}
```

○ a. 1

○ b. 2

○ c. Any Garbage Value

○ d. 0

You need to store elements in a collection that guarantees that no duplicates are stored and all elements can be accessed in natural order. Which interface provides that capability?

○ a. java.util.List

○ b. java.util.Set

○ c. java.util.Map

○ d. java.util.Collection

What is the difference between TreeSet and SortedSet?

○ a. SortedSet is an interface; TreeSet is a concrete class

○ b. TreeSet is more efficient than SortedSet

○ c. SortedSet is more efficient than TreeSet

○ d. TreeSet is an interface; SortedSet is a concrete class

Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

○ a. java.util.HashSet

○ b. java.util.LinkedHashSet

○ c. java.util.ArrayList

○ d. java.util.List

Which of these methods can be used to delete the last element in a LinkedList object?

○ a. remove()

○ b. deleteLast()

○ c. delete()

○ d. removeLast()

What will be the output of the following Java code snippet?

```
public class Test
{
 public static void main(String[] args)
       {
  Set s = new HashSet();
  s.add(new Long(10));
  s.add(new Integer(10));
  for(Object object : s)
             {
      System.out.println("test - "+object);
  }
 }
}
```

○ a. Runtime Exception

○ b. Test – 10

○ c. Test - 10
Test - 10

○ d. Compilation Failure

How to create a TreeSet that stores values in descending order ?

○ a. TreeSet<Integer> set = new TreeSet<>(Collection.orderReverse());

○ b. TreeSet<Integer> set = new TreeSet<>().reverse();

○ c. TreeSet<Integer> set = new TreeSet<>().reverseCollection();

○ d. TreeSet<Integer> set = new TreeSet<>(Collections.reverseOrder());

What is the output of this program?

```
import java.util.*;
  class Output
  {
     public static void main(String args[])
     {
        ArrayList obj = new ArrayList();
        obj.add("A");
        obj.add(0, "B");
        System.out.println(obj.size());
     }
  }
```

○ a. 3

○ b. 1

○ c. 2

○ d. 0

What will be the output of the following Java program?

```
import java.util.*;
class Linkedlist
{
   public static void main(String args[])
   {
      LinkedList obj = new LinkedList();
      obj.add("A");
      obj.add("B");
      obj.add("C");
        obj.removeFirst();
        System.out.println(obj);
     }
  }
```

○ a. [B, C]

○ b. [A, B, C]

○ c. [A, B]

○ d. [A, B, C, D]

Since Set interface in java closely resembles the mathematical set model, which of the following operations in mathematical set model can be implemented by the Set interface as well?

○ a. All of the mentioned

○ b. Union

○ c. difference

○ d. intersection

Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

○ a. java.util.LinkedHashSet

○ b. java.util.List

○ c. java.util.ArrayList

○ d. java.util.HashSet

What implementation of Iterator can traverse a collection in both directions?

○ a. ListIterator

○ b. Iterator

○ c. SetIterator

○ d. MapIterator

What is the functionality of the following piece of code? Select the most appropriate

```
public void function(int data)
{
 int flag = 0;
 if( head != null)
 {
  Node temp = head.getNext();
  while((temp != head) && (!(temp.getItem() == data)))
  {
   temp = temp.getNext();
   flag = 1;
   break;
  }
 }
 if(flag)
  System.out.println("success");
 else
  System.out.println("fail");
}
```

○ a. Print success if a particular element is equal to 1

○ b. Print fail if a particular element is not found

○ c. Print fail if the list is empty

○ d. Print success if a particular element is not found

What is the difference between length() and size() of ArrayList?

○ a. length() is not defined in ArrayList

○ b. length() and size() return the same value

○ c. length() returns the capacity of ArrayList and size() returns the actual number of elements stored in the list

○ d. size() is not defined in ArrayList

What will be the output of the following Java program?

```
import java.util.*;
class Output
{
    public static void main(String args[])
    {
        ArrayList obj = new ArrayList();
        obj.add("A");
        obj.ensureCapacity(3);
        System.out.println(obj.size());
      }
  }
```

○ a. 2

○ b. 4

○ c. 1

○ d. 3

What is the default clone of HashSet?

○ a. Hollow clone

○ b. Deep clone

○ c. Shallow clone

○ d. Plain clone

How to sort elements of ArrayList?

○ a. listObj.sort();

○ b. Collections.sort(listObj);

○ c. Collection.sort(listObj);

○ d. Sorter.sortAsc(listObj);

Which of these method of ArrayList class is used to obtain present size of an object?

○ a. capacity()

○ b. size()

○ c. index()

○ d. length()