

Week 4

1. Given the following Java program:

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        List num = new ArrayList(Arrays.asList(23, 16, 14, 33, 19, 6, 1));
        System.out.println("List is "+num);
    }
}
```

a. Give the index values of all the odd numbers assuming zero-based indexing

Ans: 23(0), 16(1), 14(2), 33(3), 19(4), 6(5), 1(6)

b. How many elements would be looked at when the list is traversed (from start to finish) until the value 19 was found?

Ans: The number of elements that would be looked at is 4.

2. Which of the following lists are syntactically correct in Java?
Try them out in to see if you were correct.

a. List num = new ArrayList(Arrays.asList(1, 2, 3, 'four'));

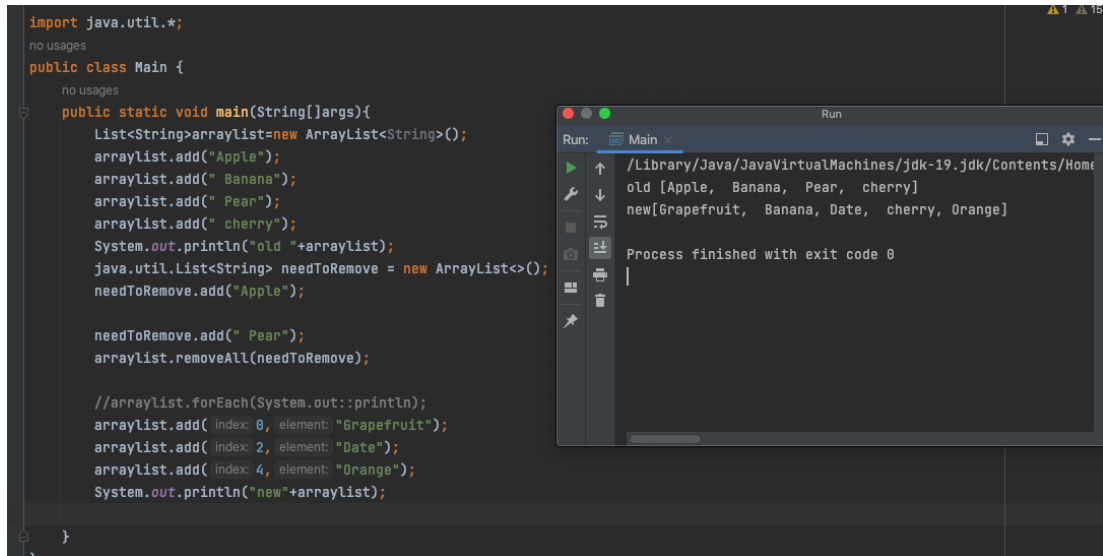
b. List num = new ArrayList(Arrays.asList(1, 2, [3, 4]));

3. Perform a series of list operations on the following list:

List fruit = new ArrayList (Arrays.asList('apple', 'banana', 'pear', 'cherry'));

to produce this updated list:

['Grapefruit', 'banana', 'Date', 'cherry', 'Orange']



The screenshot shows an IDE with a Java file named 'Main.java' and a 'Run' window. The code in 'Main.java' performs the following operations:

```
import java.util.*;

public class Main {
    public static void main(String[] args){
        List<String> arraylist = new ArrayList<String>();
        arraylist.add("Apple");
        arraylist.add(" Banana");
        arraylist.add(" Pear");
        arraylist.add(" cherry");
        System.out.println("old "+arraylist);
        java.util.List<String> needToRemove = new ArrayList<>();
        needToRemove.add("Apple");

        needToRemove.add(" Pear");
        arraylist.removeAll(needToRemove);

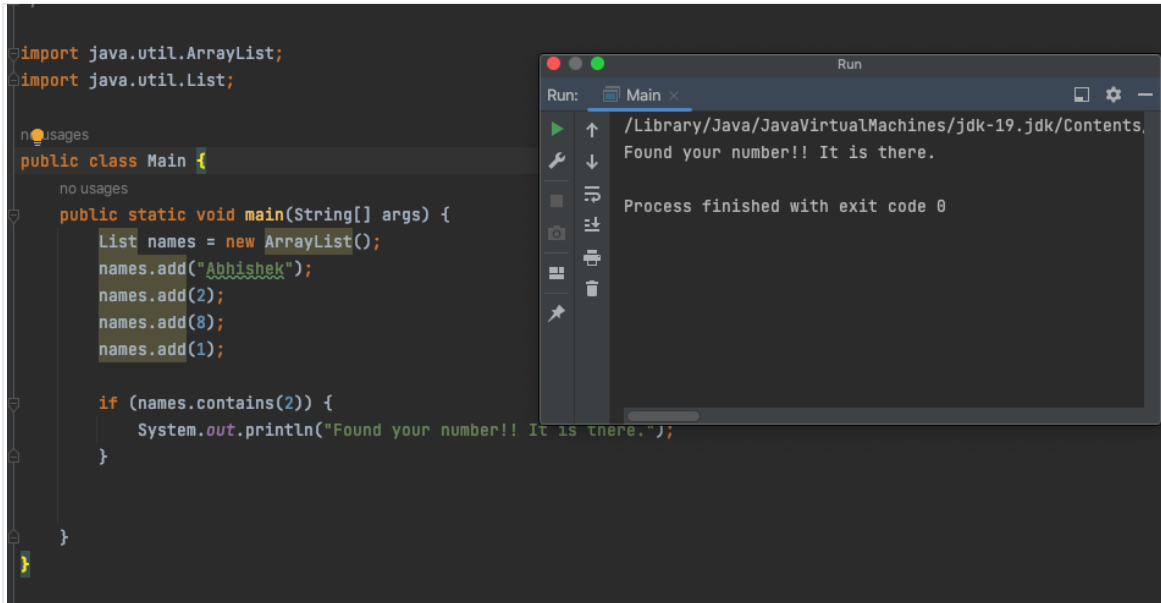
        //arraylist.forEach(System.out::println);
        arraylist.add(index: 0, element: "Grapefruit");
        arraylist.add(index: 2, element: "Date");
        arraylist.add(index: 4, element: "Orange");
        System.out.println("new "+arraylist);
    }
}
```

The 'Run' window shows the output of the program:

```
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home
old [Apple, Banana, Pear, cherry]
new [Grapefruit, Banana, Date, cherry, Orange]
Process finished with exit code 0
```

Group B

1. Write a program to find out whether a given integer is present in an array or not.



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

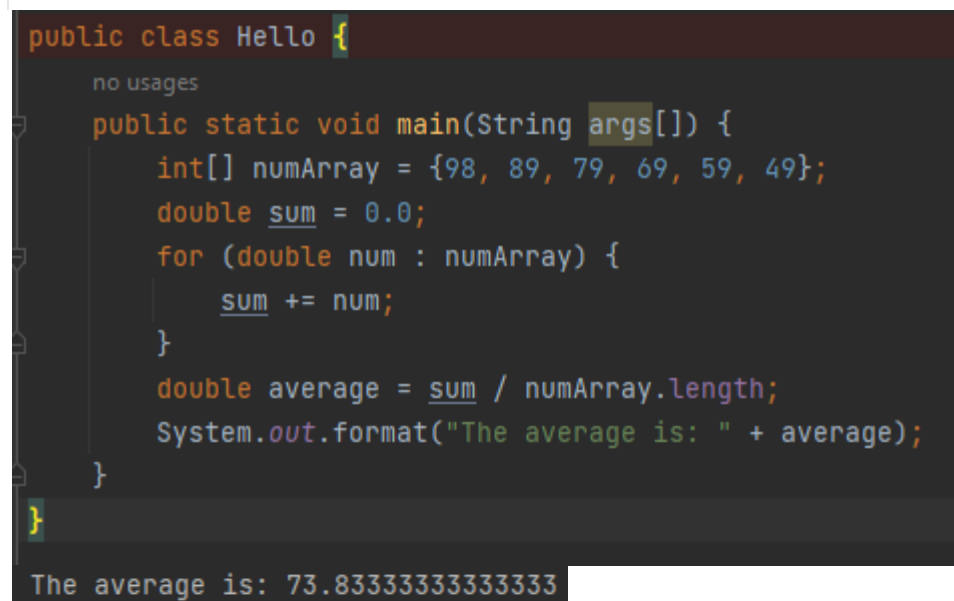
public class Main {
    public static void main(String[] args) {
        List names = new ArrayList();
        names.add("Abhishek");
        names.add(2);
        names.add(8);
        names.add(1);

        if (names.contains(2)) {
            System.out.println("Found your number!! It is there.");
        }
    }
}
```

Run: Main x

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents.
Found your number!! It is there.
Process finished with exit code 0

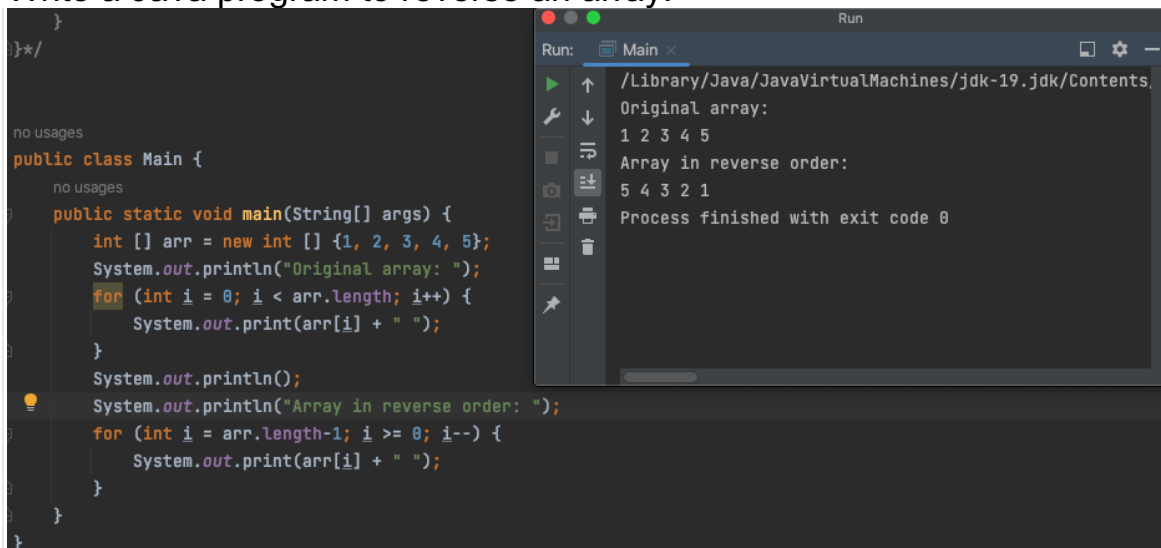
2. Calculate the average marks from an array containing marks of all students in physics using a for-each loop.



```
public class Hello {
    public static void main(String args[]) {
        int[] numArray = {98, 89, 79, 69, 59, 49};
        double sum = 0.0;
        for (double num : numArray) {
            sum += num;
        }
        double average = sum / numArray.length;
        System.out.format("The average is: " + average);
    }
}
```

The average is: 73.83333333333333

3. Write a Java program to reverse an array.



The screenshot shows an IDE with a Java file named `Main.java`. The code defines a `main` method that creates an array `arr` with values `{1, 2, 3, 4, 5}`. It first prints the original array, then iterates from the end of the array to the beginning to print the elements in reverse order. A `Run` window on the right shows the output: "Original array: 1 2 3 4 5" and "Array in reverse order: 5 4 3 2 1".

```
no usages
public class Main {
    no usages
    public static void main(String[] args) {
        int [] arr = new int [] {1, 2, 3, 4, 5};
        System.out.println("Original array: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
        System.out.println("Array in reverse order: ");
        for (int i = arr.length-1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Run: Main x

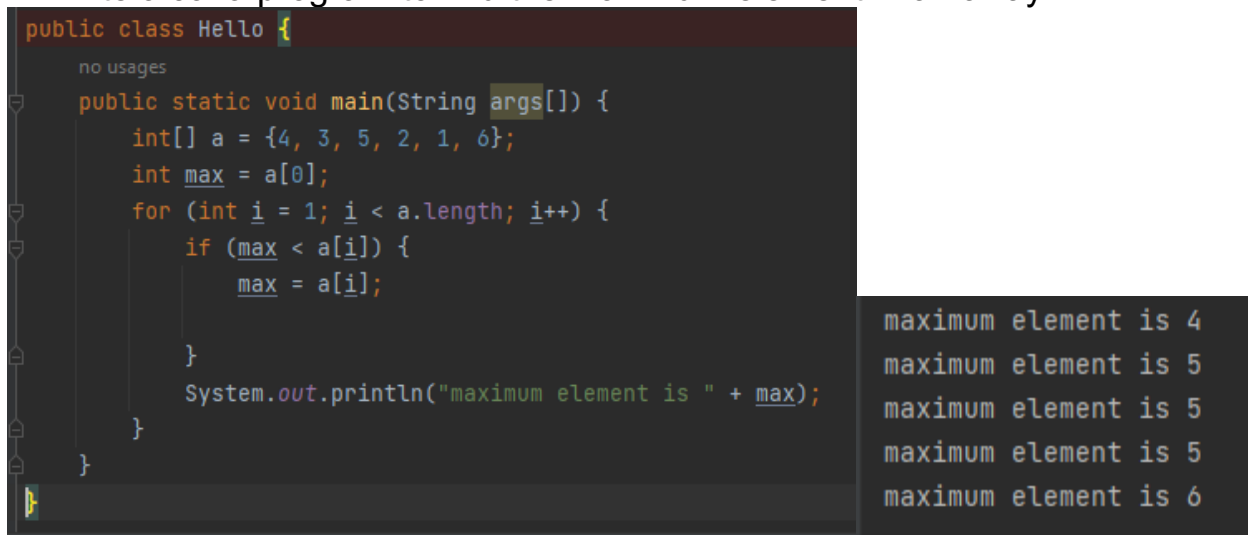
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents, ...

Original array:
1 2 3 4 5

Array in reverse order:
5 4 3 2 1

Process finished with exit code 0

4. Write a Java program to find the maximum element in an array.

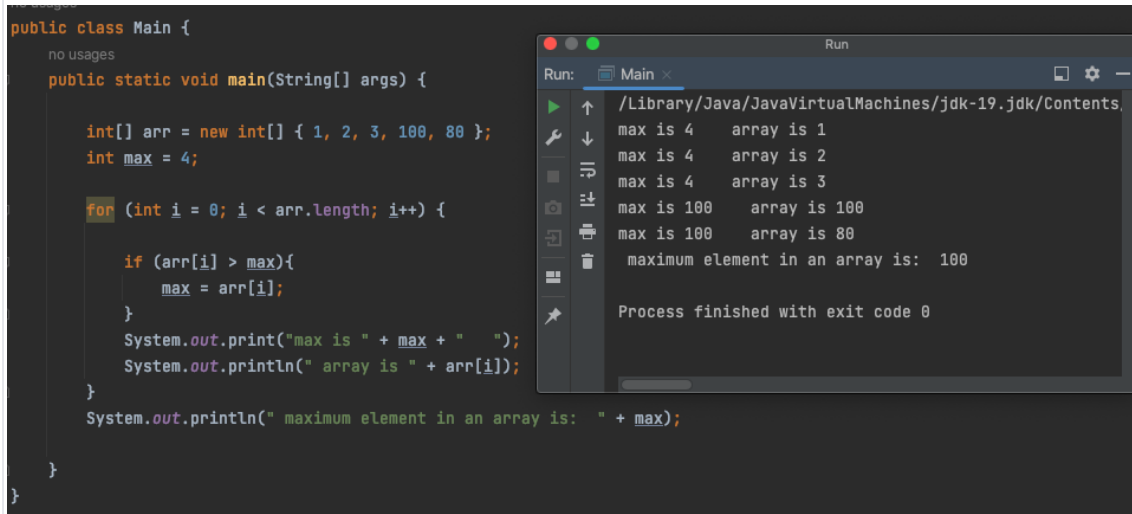


The screenshot shows an IDE with a Java file named `Hello.java`. The code defines a `main` method that creates an array `a` with values `{4, 3, 5, 2, 1, 6}`. It initializes `max` to `a[0]` and iterates through the array, updating `max` whenever a larger element is found. The output shows the maximum element being updated from 4 to 5 and finally to 6.

```
no usages
public class Hello {
    no usages
    public static void main(String args[]) {
        int[] a = {4, 3, 5, 2, 1, 6};
        int max = a[0];
        for (int i = 1; i < a.length; i++) {
            if (max < a[i]) {
                max = a[i];
            }
        }
        System.out.println("maximum element is " + max);
    }
}
```

maximum element is 4
maximum element is 5
maximum element is 5
maximum element is 5
maximum element is 6

5. Write a Java program to find whether an array is sorted or not.



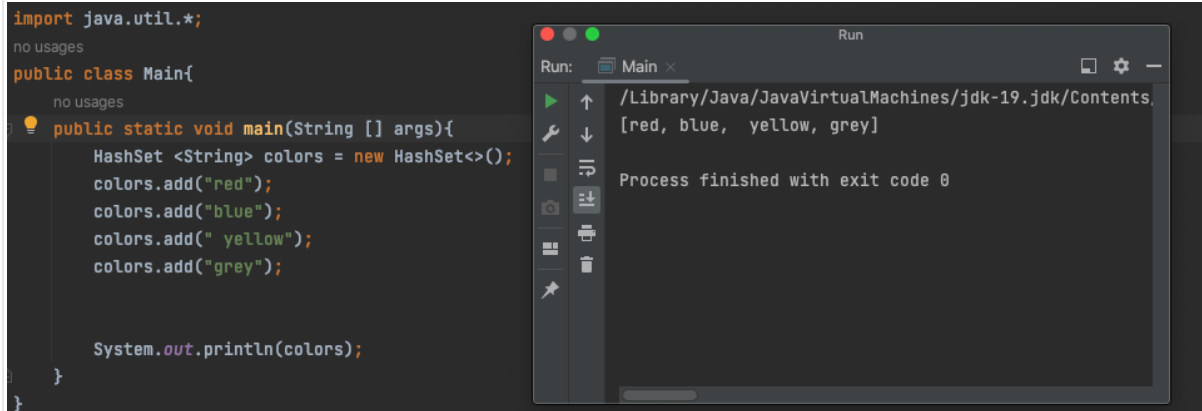
```
public class Main {  
    no usages  
    public static void main(String[] args) {  
  
        int[] arr = new int[] { 1, 2, 3, 100, 80 };  
        int max = 4;  
  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] > max){  
                max = arr[i];  
            }  
            System.out.print("max is " + max + " ");  
            System.out.println(" array is " + arr[i]);  
        }  
        System.out.println(" maximum element in an array is: " + max);  
    }  
}
```

Run: Main x

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents.
max is 4 array is 1
max is 4 array is 2
max is 4 array is 3
max is 100 array is 100
max is 100 array is 80
maximum element in an array is: 100
Process finished with exit code 0

Group C

1. Write a Java program to append the specified element to the end of a hash set.



```
import java.util.*;

public class Main{

    public static void main(String [] args){

        HashSet <String> colors = new HashSet<>();
        colors.add("red");
        colors.add("blue");
        colors.add(" yellow");
        colors.add("grey");

        System.out.println(colors);

    }

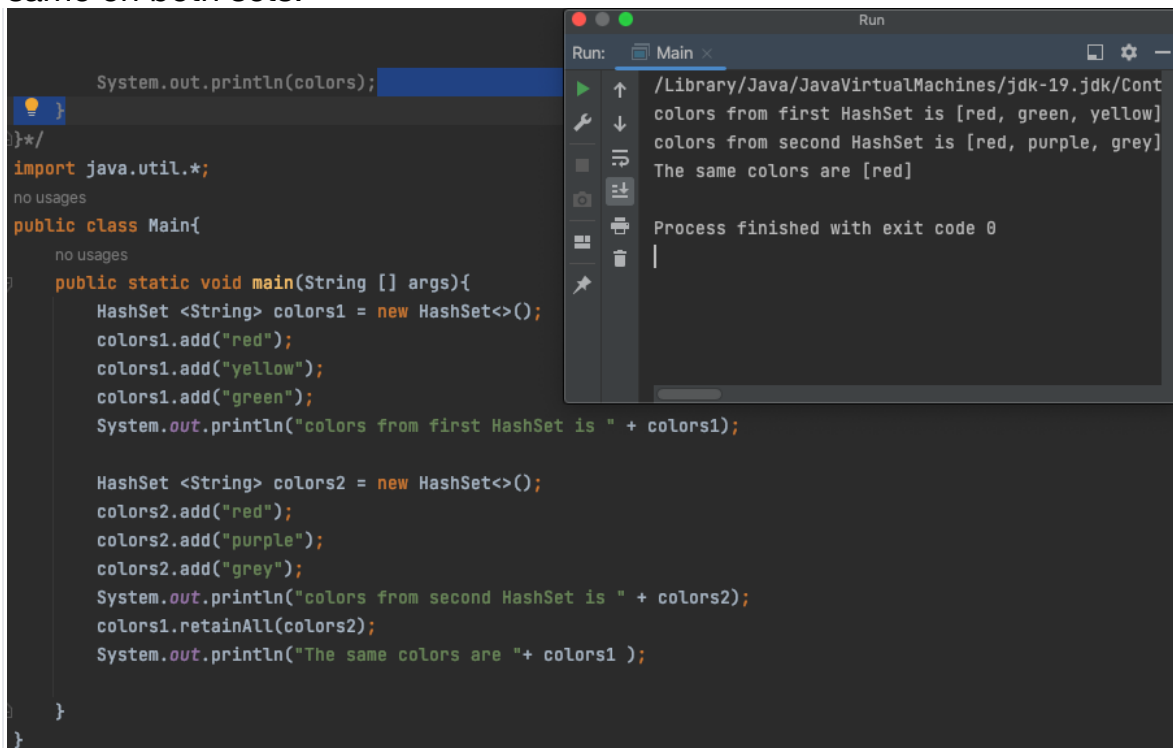
}
```

Run: Main x

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents.
[red, blue, yellow, grey]

Process finished with exit code 0

2. Write a Java program to compare two sets and retain elements which are same on both sets.



```
System.out.println(colors);
}*/
import java.util.*;

public class Main{

    public static void main(String [] args){

        HashSet <String> colors1 = new HashSet<>();
        colors1.add("red");
        colors1.add("yellow");
        colors1.add("green");
        System.out.println("colors from first HashSet is " + colors1);

        HashSet <String> colors2 = new HashSet<>();
        colors2.add("red");
        colors2.add("purple");
        colors2.add("grey");
        System.out.println("colors from second HashSet is " + colors2);
        colors1.retainAll(colors2);
        System.out.println("The same colors are "+ colors1 );

    }

}
```

Run: Main x

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Cont
colors from first HashSet is [red, green, yellow]
colors from second HashSet is [red, purple, grey]
The same colors are [red]

Process finished with exit code 0

3. Write a Java program to count the number of key-value mappings in a hash table

```
import java.util.*;

no usages

public class Hello {

    no usages

    public static void main(String args[]) {

        HashMap<Integer, String> hash_map = new HashMap<Integer, String>();
        hash_map.put(1, "Red");
        hash_map.put(2, "Blue");
        hash_map.put(3, "White");
        hash_map.put(4, "Black");
        hash_map.put(5, "Pink");
        System.out.println("Size of the hash map: " + hash_map.size());

    }

}

Size of the hash map: 5
```

4. Write a Java program to get a collection view of the values contained in this map

```
System.out.println("The same colors are " + colors1);
}
}*/
import java.util.*;
no usages
public class Main {
    no usages
    public static void main(String args[]){
        HashMap<Integer,String> animal = new HashMap<>();
        animal.put(1, "cow");
        animal.put(3, "dog");
        animal.put(8, "cat");
        animal.put(13, "elephant");
        animal.put(2, "hipo");
        System.out.println(animal);
        System.out.println(" Number of key-value mappings in a hashtable is: "+animal.size());
        System.out.println("The value is " + animal.values());
    }
}
```

Run: Main x

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home
{1=cow, 2=hipo, 3=dog, 8=cat, 13=elephant}
Number of key-value mappings in a hashtable is: 5
The value is [cow, hipo, dog, cat, elephant]

Process finished with exit code 0

