Java PS4

Name: Shikhar Bakhda
UNI: ssb2189

R7.6

a) 25
b) 13
c) 12
d) The code will give an error as index will be out of bounds – no element exists at a[10]. The last value of total before the error was 22.
e) 11
f) 25
g) 12
h) -1

R7.13

a) for(int i : values){ total = total + i; }
b) for(int i : values){ total = total + i; }total = total – values[0];
c)
```
int counter = 0;
for(int i : values){
        if (i == target) {
                return counter;
                }
        else{
                counter = counter + 1;
        }
```

R7.23

Set an integer ‘longestRun’ to 0
Get the array as ‘values’
For each valid index ‘i’ in the range 0 *inclusive* to (length of values) *exclusive*
    Set an integer ‘currentRunIndex’ to the value of ‘i’
    Set an integer ‘currentRun’ to 0
    While values[currentRunIndex] equals values[i]
        Set the value of ‘currentRun’ to (currentRun + 1)
        Set the value of ‘currentRunIndex’ to (currentRunIndex + 1)
        If ‘currentRunIndex’ equals (length of values)
            Break out of the While loop
    If currentRun > longestRun
        Set the value of ‘longestRun’ to the value of ‘currentRun’
Print longestRun

R7.32

a) True
b) False
c) False
d) False
e) False
f) True
g) True

R7.33

a)

```
ArrayList<Integer> array1 = new ArrayList<Integer>; //Initialising array list array1
ArrayList<Integer> array2 = new ArrayList<Integer>; //Initialising array list array2

if(array1.size() != array2.size()) //checking if both array lists have the same size
        return false;  //Returning false if they're not the same size

//This for loop has a variable i that loops from 0 through last element,
// array1.size() - 1
for (int i = 0; i < array1.size(); i++) {
// If the ith element of array1 doesn't equal the ith element of array2, return false
    if (array1.get(i) != array2.get(i))
        return false;

// We only reach here if nothing was returned from the loop above.
//This means that every ith element of array1 equals the ith element of array2.
return true;        //Thus, we return true
```

b)

```
ArrayList<Integer> array1 = new ArrayList<Integer>;  //Initialising array list array1
ArrayList<Integer> array2 = new ArrayList<Integer>(array1); //Copying over array1
                                                            // into array2
```

c)

```
ArrayList<Integer> array129/10/16 = new ArrayList<Integer>;  //Initialising array list
array1

//This for loop has a variable i that loops from 0 through last element,
// array1.size() - 1
for (int i = 0; i < array1.size(); i++)
        array1.set(i, 0); //The set function replaces the element at index i with 0
```

d)

```
ArrayList<Integer> array1 = new ArrayList<Integer>;  //Initialising array list array1
while(array1.size() != 0)
      array1.remove(0); //The remove function removes first element till the array
                        //length is becomes 0
```