



Part 1:

A palindrome is a sequence of characters that reads the same backward as forward. For example, each of the following five-digit integers is a palindrome: 12321, 55555, 45554 and 11611. Write an application that reads in a five-digit integer and determines whether it is a palindrome. If the number is not five digits long, display an error message and allow the user to enter a new value.

Sample Output

```
Enter a 5-digit number: 1234
Number must be 5 digits
Enter a 5-digit number: 123456
Number must be 5 digits
Enter a 5-digit number: 54345
54345 is a palindrome!!!
```

```
Palindromes.java PalindromeTest.java x
1 package lab4;
2 public class PalindromeTest {
3
4     public static void main(String[] args)
5     {
6         Palindrome application = new Palindrome();
7         application.checkPalindrome();
8     }
9 }
10

Console x Problems Debug Shell Terminal
<terminated> PalindromeTest [Java Application] C:\Users\Hayan\.p2\pool\plugins\org.
Enter a 5 digit number: 44444
44444 is a Palindrome
```

```

3
4 public class Palindrome
5 {
6     public void checkPalindrome()
7     {
8         Scanner input = new Scanner( System.in );
9         int number;
10        int digit1 = 0;
11        int digit2 = 0;
12        int digit4 = 0;
13        int digit5 = 0;
14
15        System.out.print("Enter a 5 digit number: ");
16        number = input.nextInt();
17
18        int digits = Integer.toString(number).length();
19
20        while ( digits != 5)
21        {
22            System.out.println("Number must be 5 digits long, try again");
23            System.out.print("Enter a 5 digit number: ");
24            number = input.nextInt();
25            digits = Integer.toString(number).length();
26        }
27
28        digit1 = number/10000;
29        digit2 = (number%10000)/1000;
30        digit4 = (((number%10000)%1000)%100)/10;
31        digit5 = (((number%10000)%1000)%100)%10;
32
33        if (digit1 == digit5 && digit2 == digit4)
34            System.out.printf("%d is a Palindrome", number);
35        else
36            System.out.printf("%d is not a palindrome", number);
37    }
38 }

```

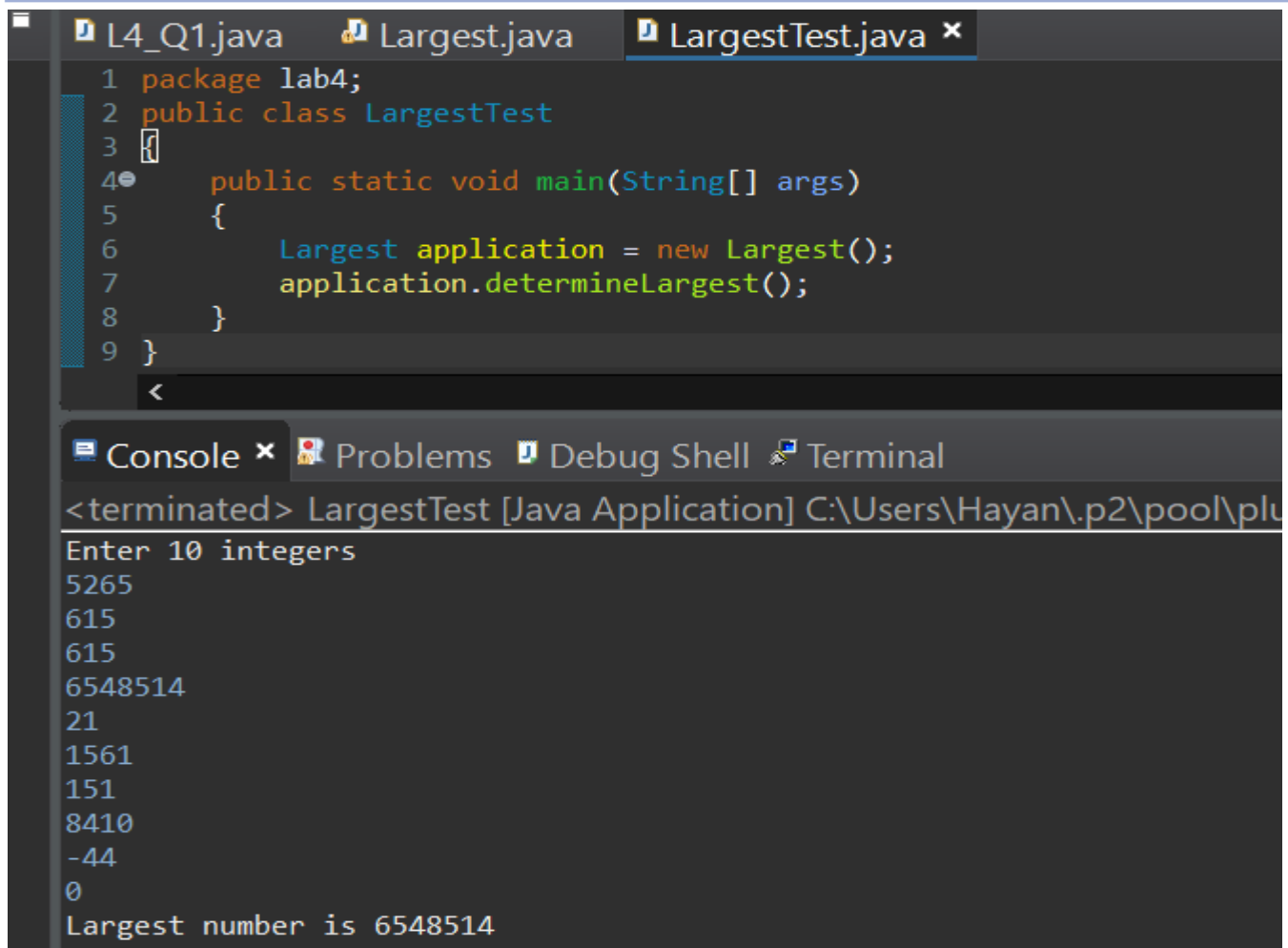
Part 2:

Write a Java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:

- a) counter: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
- b) number: The integer most recently input by the user.
- c) largest: The largest number found so far.

Sample Output

```
Enter number: 56
Enter number: -10
Enter number: 200
Enter number: 25
Enter number: 8
Enter number: 500
Enter number: -20
Enter number: 678
Enter number: 345
Enter number: 45
Largest number is 678
```



The screenshot shows an IDE with three tabs: L4_Q1.java, Largest.java, and LargestTest.java. The LargestTest.java tab is active, displaying the following code:

```
1 package lab4;
2 public class LargestTest
3 {
4     public static void main(String[] args)
5     {
6         Largest application = new Largest();
7         application.determineLargest();
8     }
9 }
```

Below the code editor, the Console window is visible, showing the execution output:

```
<terminated> LargestTest [Java Application] C:\Users\Hayan\.p2\pool\plu
Enter 10 integers
5265
615
615
6548514
21
1561
151
8410
-44
0
Largest number is 6548514
```

```
1 package lab4;
2 import java.util.Scanner;
3 public class Largest
4 {
5     public void determineLargest()
6     {
7         Scanner input = new Scanner(System.in);
8         int largest = 0;
9         int number = 0;
10        int counter = 0;
11
12        System.out.println("Enter 10 integers");
13        while (counter<10)
14        {
15            number = input.nextInt();
16            if (number > largest)
17                largest = number;
18            counter++;
19        }
20        System.out.printf( "Largest number is %d\n", largest );
21    }
22 }
```