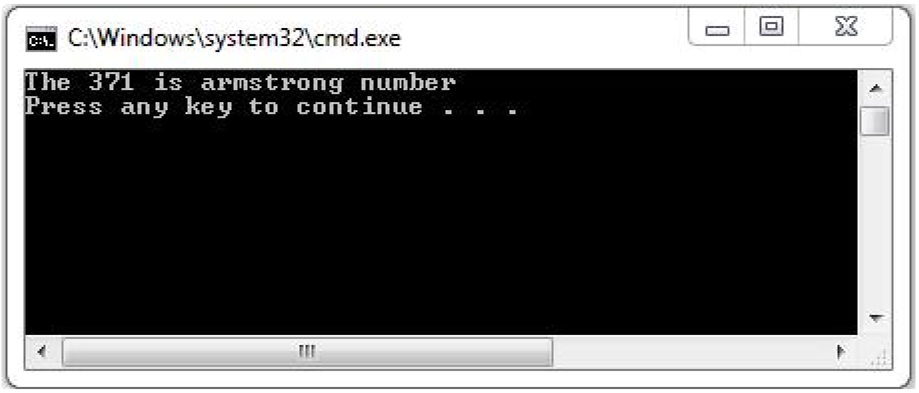
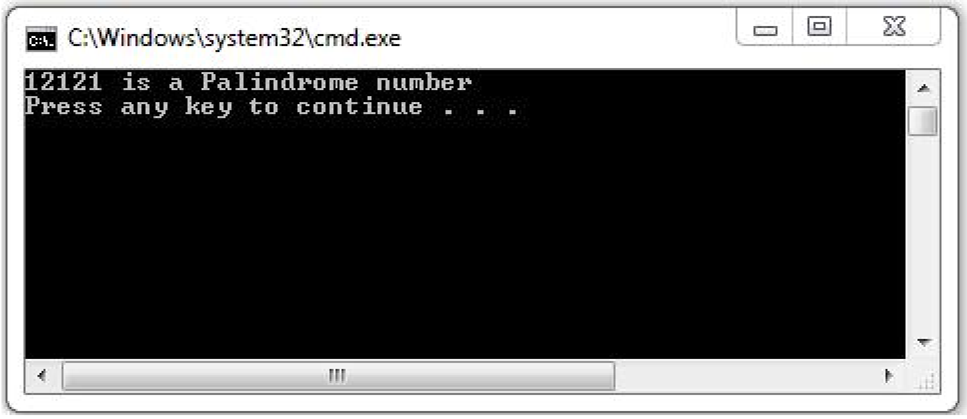
LAB 1 (Programming in Java)

1. Write a Java program to check if the given number is Armstrong number or not.



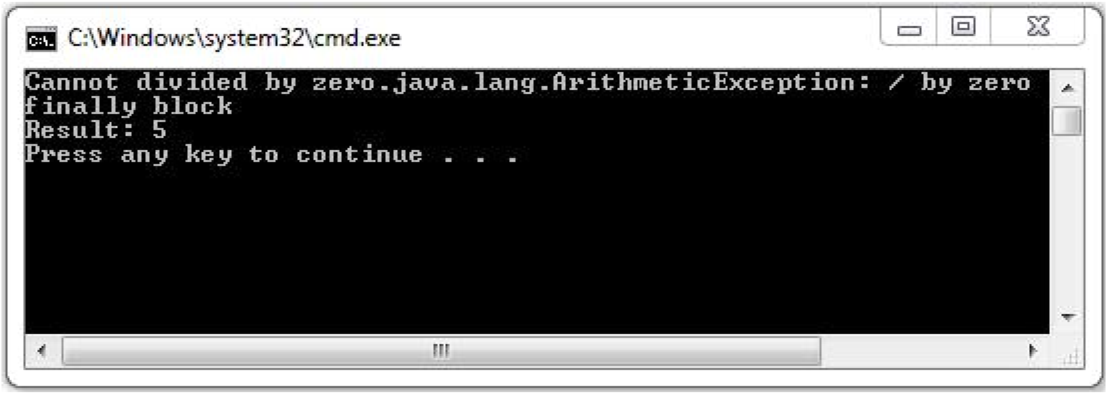
```
The 371 is armstrong number
Press any key to continue . . .
```

2. Write a Java program to check the given number is palindrome or not.



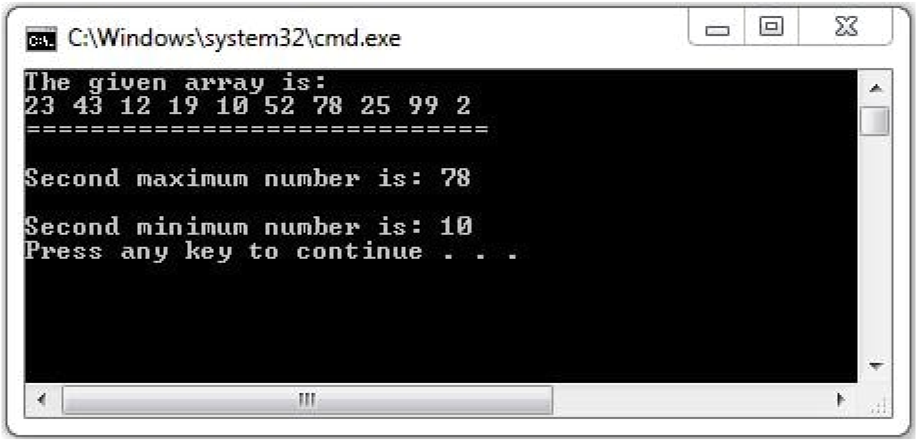
```
12121 is a Palindrome number
Press any key to continue . . .
```

3. Write a Java program that demonstrate the try, catch and finally block to handle exception.



```
Cannot divided by zero.java.lang.ArithmeticException: / by zero
finally block
Result: 5
Press any key to continue . . .
```

4. Write a program to find second minimum and second maximum number from the array.



```
The given array is:
23 43 12 19 10 52 78 25 99 2
=============================

Second maximum number is: 78

Second minimum number is: 10
Press any key to continue . . .
```

5. Write a Java program that creates two threads: one for printing even numbers and another for printing odd numbers. The program should ensure that the threads print the numbers alternatively, starting from 1 up to a given limit (e.g., 20)

   (Hint : You may use the synchronized keyword, wait(), and notify())

6. Write a Java program that defines a Person class with the following attributes: name, age, and gender. Include a constructor to initialize these attributes. Create two instances of the Person class and display their details.

7. Create a base class called Shape with methods calculateArea() and calculatePerimeter(). Derive two classes, Circle and Rectangle, from the Shape class. Implement the necessary methods in each derived class and demonstrate their usage

8. Design a BankAccount class with private attributes accountNumber, accountHolder, and balance. Include methods to deposit and withdraw funds, and a method to display the account details. Ensure proper encapsulation by providing public accessors and mutators

9. Create an interface called Drawable with a method draw(). Implement this interface in two classes, Circle and Square. Create an array of Drawable objects and invoke the draw() method for each object.

10. Design an abstract class called Animal with abstract methods makeSound() and move(). Create two subclasses, Dog and Bird, that extend the Animal class. Implement the abstract methods in each subclass and demonstrate their usage.