Problem No: 01

Given two non-negative int values, return true if they have the same last digit, such as with 27 and 57. Note that the % "mod" operator computes remainders, so 17 % 10 is 7.

lastDigit(7, 17) → true

lastDigit(6, 17) → false

lastDigit(3, 113) → true

Code:

```
public class SameLastDigit {

   public static void main(String[] args) {

       int num1 = 27;

       int num2 = 57;


       boolean result = hasSameLastDigit(num1, num2);


       System.out.println("Do the numbers have the same last digit? " + result);

   }


   public static boolean hasSameLastDigit(int num1, int num2) {

       int lastDigitNum1 = num1 % 10;

       int lastDigitNum2 = num2 % 10;


       return lastDigitNum1 == lastDigitNum2;

   }
```

```
}
```

Problem No: 02

Given one integer n and return true if it is an even number else return false.

Even(6) -> True.

Even(7) -> False.

Even(9) -> False.

Code:

```
public class CheckEven {

   public static void main(String[] args) {

       int num1 = 6;

       int num2 = 7;

       int num3 = 9;



       System.out.println("Is " + num1 + " even? " + isEven(num1));

       System.out.println("Is " + num2 + " even? " + isEven(num2));

       System.out.println("Is " + num3 + " even? " + isEven(num3));

   }



   public static boolean isEven(int num) {

       return num % 2 == 0;

   }

}
```

Problem no: 03

Given two int values, return their sum. Unless the two values are the same, then return double their sum.

sumDouble(1, 2) → 3

sumDouble(3, 2) → 5

sumDouble(2, 2) → 8

Code:

```
public class SumDouble {

    public static void main(String[] args) {

        int num1 = 1;

        int num2 = 2;



        int result = sumDouble(num1, num2);



        System.out.println("The result is: " + result);

    }



    public static int sumDouble(int a, int b) {

        if (a == b) {

            return 2 * (a + b);

        } else {

            return a + b;

        }

    }

}
```

Problem no: 04

Given 2 ints, a and b, return true if one of them is 10 or if their sum is 10.

makes10(9, 10) → true

makes10(9, 9) → false

makes10(1, 9) → true

Code:

```
public class Makes10 {

   public static void main(String[] args) {

       int num1 = 9;

       int num2 = 10;


       boolean result = makes10(num1, num2);


       System.out.println("The result is: " + result);

   }


   public static boolean makes10(int a, int b) {

       return (a == 10 || b == 10 || a + b == 10);

   }

}
```

Problem No: 05

Given 2 int values, return true if either of them is in the range 10..20 inclusive.

in1020(12, 99) → true

in1020(21, 12) → true

in1020(8, 99) → false

Code:

```
public class In1020 {

   public static void main(String[] args) {

       int num1 = 12;

       int num2 = 99;


       boolean result = in1020(num1, num2);


       System.out.println("The result is: " + result);

   }


   public static boolean in1020(int a, int b) {

       return (a >= 10 && a <= 20) || (b >= 10 && b <= 20);

   }

}
```