Given 3 int values, a b c, return their sum. However, if one of the values is 13 then it does not count towards the sum and values to its right do not count. So for example, if b is 13, then both b and c do not count.

luckySum(1, 2, 3) → 6
luckySum(1, 2, 13) → 3
luckySum(1, 13, 3) → 1

```
public class LuckySum {

  public static int luckySum(int a, int b, int c) {

    if (a == 13) {

      return 0;

    } else if (b == 13) {

      return a;

    } else if (c == 13) {

      return a + b;

    } else {

      return a + b + c;

    }

  }


  public static void main(String[] args) {

    System.out.println(luckySum(1, 2, 3));

    System.out.println(luckySum(1, 2, 13));

    System.out.println(luckySum(1, 13, 3));

  }

}
```

2] **Given 3 int values, a b c, return their sum. However, if any of the values is a teen -- in the range 13..19 inclusive -- then that value counts as 0, except 15 and 16 do not count as a teens. Write a separate helper "public int fixTeen(int n) {"that takes in an int value and returns that value fixed for**

the teen rule. In this way, you avoid repeating the teen code 3 times (i.e. "decomposition"). Define the helper below and at the same indent level as the main noTeenSum().

noTeenSum(1, 2, 3) → 6
noTeenSum(2, 13, 1) → 3
noTeenSum(2, 1, 14) → 3

```
public class GreenLotteryTicket {
   public static void main(String[] args) {
      System.out.println(greenTicket(1, 2, 3));
      System.out.println(greenTicket(2, 2, 2));
      System.out.println(greenTicket(1, 1, 2));
   }

   public static int greenTicket(int a, int b, int c) {
      if (a == b && b == c) {
         return 20;
      } else if (a == b || a == c || b == c) {
         return 10;
      } else {
         return 0;
      }
   }
}
```

3] **You have a green lottery ticket, with ints a, b, and c on it. If the numbers are all different from each other, the result is 0. If all of the numbers are the same, the result is 20. If two of the numbers are the same, the result is 10.**

greenTicket(1, 2, 3) → 0
greenTicket(2, 2, 2) → 20
greenTicket(1, 1, 2) → 10

```
public class GreenLotteryTicket {
   public static void main(String[] args) {
      System.out.println(greenTicket(1, 2, 3));
      System.out.println(greenTicket(2, 2, 2));
      System.out.println(greenTicket(1, 1, 2));
   }
```

```
    public static int greenTicket(int a, int b, int c) {
      if (a == b && b == c) {
        return 20;
      } else if (a == b || a == c || b == c) {
        return 10;
      } else {
        return 0;
      }
    }
}
```

4] **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

```
public class LastDigit {
    public static void main(String[] args) {
        System.out.println(lastDigit(7, 17));
        System.out.println(lastDigit(6, 17));
        System.out.println(lastDigit(3, 113));
    }

    public static boolean lastDigit(int a, int b) {
        return a % 10 == b % 10;
    }
}
```

5] **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.

```
public class Even {
   public static void main(String[] args) {
      System.out.println(isEven(6));
      System.out.println(isEven(7));
      System.out.println(isEven(9));
   }

   public static boolean isEven(int n) {
      return n % 2 == 0;
   }
}
```

6] **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

```
Public class SumDouble {
   Public static void main(String[] args) {
      System.out.println(sumDouble(1, 2));
      System.out.println(sumDouble(3, 2));
      System.out.println(sumDouble(2, 2));
   }

   Public static int sumDouble(int a, int b) {
```

```
        If (a == b) {
            Return 2 * (a + b);
        } else {
            Return a + b;
        }
    }
}
```

7] **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

```
Public class Makes10 {
    Public static void main(String[] args) {
        System.out.println(makes10(9, 10));
        System.out.println(makes10(9, 9));
        System.out.println(makes10(1, 9));
    }

    Public static boolean makes10(int a, int b) {
        Return (a == 10 || b == 10) || (a + b == 10);
    }
}
```

8] **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

```
Public class In1020 {
   Public static void main(String[] args) {
      System.out.println(in1020(12, 99));
      System.out.println(in1020(21, 12));
      System.out.println(in1020(8, 99));
   }

   Public static boolean in1020(int a, int b) {
      Return (a >= 10 && a <= 20) || (b >= 10 && b <= 20);
   }
}
```