|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Warmup-1 | | | | |
| sleepIn | sumDouble | makes10 | notString | frontBack |
| or35 | startHi | hasTeen | mixStart | intMax |
| stringE | lastDigit | everyNth |  |  |

// The parameter weekday is true if it is a weekday,   
 // and the parameter vacation is true if we are on vacation.   
 // We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in.  
 // Ex) sleepIn(true, false) -> false  
 public static boolean sleepIn(boolean weekday, boolean

vacation)  
 {

}  
  
 // Given two int values, return their sum.   
 // Unless the two values are the same, then return double their sum.  
 // Ex) sumDouble(2, 2) -> 8  
 public static int sumDouble(int a, int b)  
 {

}   
  
 // Given 2 ints, a and b, return true if one if them is 10 or if their sum is 10.  
 public static boolean makes10(int a, int b)  
 {

}  
  
 // Given a string, return a new string where "not " has been added to the front.   
 // However, if the string already begins with "not", return the string unchanged.   
 public static String notString(String str)  
 {

}  
 // Given a string, return a new string where the first and last chars have been exchanged.  
 // Ex) frontBack("code") -> "eodc"  
 public static String frontBack(String str)  
 {

}  
 // Return true if the given non-negative number is a multiple of 3 or a multiple of 5.   
 public static boolean or35(int n)  
 {

}  
 // Given a string, return true if the string starts with "hi" and false otherwise.  
 public static boolean startHi(String str)  
 {

}  
  
 // We'll say that a number is "teen" if it is in the range 13..19 inclusive.   
 // Given 3 int values, return true if one or more of them are teen.  
 public static boolean hasTeen(int a, int b, int c)  
 {

}  
   
 // Return true if the given string begins with "mix", except the 'm' can be anything,   
 // so "pix", "9ix" .. all count.  
 public static boolean mixStart(String str)  
 {

}  
  
 // Given three int values, a b c, return the largest.  
 public static int intMax(int a, int b, int c)  
 {

}  
 // Return true if the given string contains between 1 and 3 'e' chars.  
 public static boolean stringE(String str)  
 {

}  
 // Given two non-negative int values,   
 // return true if they have the same last digit, such as with 27 and 57.   
 public static boolean lastDigit(int a, int b)  
 {

}  
  
 // Given a non-empty string and an int N,   
 // return the string made starting with char 0, and then every Nth char of the string.   
 // So if N is 3, use char 0, 3, 6, ... and so on. N is 1 or more.  
 // Ex) everyNth("abcdefg", 3) -> "adg"  
 public static String everyNth(String str, int n)   
 {

}