Задачи на закрепления всех основных конструкций — 10

1. 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.

```
sleepIn(false, false) → true
sleepIn(true, false) → false
sleepIn(false, true) → true
```

2. Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

```
diff21(19) → 2
diff21(10) → 11
diff21(21) → 0
```

3. Given an int n, return true if it is within 10 of 100 or 200. Note: Math.abs(num) computes the absolute value of a number.

```
nearHundred(93) → true
nearHundred(90) → true
nearHundred(89) → false
```

4. Given a non-empty string and an int n, return a new string where the char at index n has been removed. The value of n will be a valid index of a char in the original string (i.e. n will be in the range 0..str.length()-1 inclusive).

```
missingChar("kitten", 1) → "ktten"
missingChar("kitten", 0) → "itten"
missingChar("kitten", 4) → "kittn"
```

5. Given a string, take the last char and return a new string with the last char added at the front and back, so "cat" yields "tcatt". The original string will be length 1 or more.

```
backAround("cat") → "tcatt"
backAround("Hello") → "oHelloo"
backAround("a") → "aaa"
```

6. Given a string, return true if the string starts with "hi" and false otherwise.

```
startHi("hi there") → true
startHi("hi") → true
startHi("hello hi") → false
```

7. We'll say that a number is "teen" if it is in the range 13..19 inclusive. Given 3 int values, return true if 1 or more of them are teen.

```
hasTeen(13, 20, 10) → true
hasTeen(20, 19, 10) → true
```

```
hasTeen(20, 10, 13) → true
```

8. Return true if the given string begins with "mix", except the 'm' can be anything, so "pix", "9ix" .. all count.

```
mixStart("mix snacks") → true
mixStart("pix snacks") → true
mixStart("piz snacks") → false
```

9. Given 2 int values, return whichever value is nearest to the value 10, or return 0 in the event of a tie. Note that Math.abs(n) returns the absolute value of a number.

```
close10(8, 13) → 8
close10(13, 8) → 8
close10(13, 7) → 0
```

10. Return true if the given string contains between 1 and 3 'e' chars.

```
stringE("Hello") → true
stringE("Heelle") → true
stringE("Heelele") → false
```

11. 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.

```
everyNth("Miracle", 2) → "Mrce"
everyNth("abcdefg", 2) → "aceg"
everyNth("abcdefg", 3) → "adg"
```

12. We have two monkeys, a and b, and the parameters aSmile and bSmile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return true if we are in trouble.

```
monkeyTrouble(true, true) → true
monkeyTrouble(false, false) → true
monkeyTrouble(true, false) → false
```

13. We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return true if we are in trouble.

```
parrotTrouble(true, 6) → true
parrotTrouble(true, 7) → false
parrotTrouble(false, 6) → false
```

14. Given 2 int values, return true if one is negative and one is positive. Except if the parameter "negative" is true, then return true only if both are negative.

```
posNeg(1, -1, false) → true
posNeg(-1, 1, false) → true
```

```
posNeg(-4, -5, true) → true
```

15. Given a string, return a new string where the first and last chars have been exchanged.

```
frontBack("code") → "eodc"
frontBack("a") → "a"
frontBack("ab") → "ba"
```

16. Return true if the given non-negative number is a multiple of 3 or a multiple of 5. Use the % "mod" operator -- see Introduction to Mod

```
or35(3) → true
or35(10) → true
or35(8) → false
```

17. Given two temperatures, return true if one is less than 0 and the other is greater than 100.

```
icyHot(120, -1) → true
icyHot(-1, 120) → true
icyHot(2, 120) → false
```

18. We'll say that a number is "teen" if it is in the range 13..19 inclusive. Given 2 int values, return true if one or the other is teen, but not both.

```
loneTeen(13, 99) → true
loneTeen(21, 19) → true
loneTeen(13, 13) → false
```

19. Given a string, return a string made of the first 2 chars (if present), however include first char only if it is 'o' and include the second only if it is 'z', so "ozymandias" yields "oz".

```
startOz("ozymandias") → "oz"
startOz("bzoo") → "z"
startOz("oxx") → "o"
```

20. Given 2 int values, return true if they are both in the range 30..40 inclusive, or they are both in the range 40..50 inclusive.

```
in3050(30, 31) → true
in3050(30, 41) → false
in3050(40, 50) → true
```

21. 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
```

Разминка номер два

1. Given a string and a non-negative int n, return a larger string that is n copies of the original string.

```
stringTimes("Hi", 2) → "HiHi"
stringTimes("Hi", 3) → "HiHiHi"
stringTimes("Hi", 1) → "Hi"
```

2. Given a string, return true if the first instance of "x" in the string is immediately followed by another "x".

```
doubleX("axxbb") → true
doubleX("axaxax") → false
doubleX("xxxxx") → true
```

3. Given a string, return the count of the number of times that a substring length 2 appears in the string and also as the last 2 chars of the string, so "hixxxhi" yields 1 (we won't count the end substring).

```
last2("hixxhi") → 1
last2("xaxxaxaxx") → 1
last2("axxxaaxx") → 2
```

4. Given a string, return a string made of the chars at indexes 0,1, 4,5, 8,9 ... so "kittens" yields "kien".

```
altPairs("kitten") → "kien"
altPairs("Chocolate") → "Chole"
altPairs("CodingHorror") → "Congrr"
```

5. Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front;

```
frontTimes("Chocolate", 2) → "ChoCho"
frontTimes("Chocolate", 3) → "ChoChoCho"
frontTimes("Abc", 3) → "AbcAbcAbc"
```

6. Given a string, return a new string made of every other char starting with the first, so "Hello" yields "Hlo".

```
stringBits("Hello") → "Hlo"
stringBits("Hi") → "H"
stringBits("Heeololeo") → "Hello"
```

7. Given 2 strings, a and b, return the number of the positions where they contain the same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa", and "az" substrings appear in the same place in both strings.

```
stringMatch("xxcaazz", "xxbaaz") → 3
stringMatch("abc", "abc") → 2
stringMatch("abc", "axc") → 0
```

8. Suppose the string "yak" is unlucky. Given a string, return a version where all the "yak" are removed, but the "a" can be any char. The "yak" strings will not overlap.

```
stringYak("yakpak") → "pak"
stringYak("pakyak") → "pak"
stringYak("yak123ya") → "123ya"
```

9. Count the number of "xx" in the given string. We'll say that overlapping is allowed, so "xxx" contains 2 "xx".

```
countXX("abcxx") → 1
countXX("xxx") → 2
countXX("xxxx") → 3
```

10. Given a non-empty string like "Code" return a string like "CCoCodCode".

```
stringSplosion("Code) → "CCoCodCode"
stringSplosion("abc") → "aababc"
stringSplosion("ab") → "aab"
```

11. Given a string, return a version where all the "x" have been removed. Except an "x" at the very start or end should not be removed.

```
stringX("xxHxix") → "xHix"
stringX("abxxxcd") → "abcd"
stringX("xabxxxcdx") → "xabcdx"
```


FizzBuzz - Задача, которая очень часто встречается на собеседовании

- Program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

```
fizzBuzz() → "Fizz"
fizzBuzz() → "Buzz"
fizzBuzz() → "FizzBuzz"
```