

Параметр weekday имеет значение true если это рабочий день недели, и параметр vacation имеет значение true если у нас каникулы. Мы спим, если это не рабочий день или у нас каникулы. Вернуть значение true, когда мы спим. Например:

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

Написать код функции:

```
public boolean sleepIn(boolean weekday, boolean vacation) {  
  
}
```

Дано целое число n, вернуть абсолютную разницу между n и 21, но вернуть удвоенную разницу, если n больше 21. Например:

```
diff21(19) → 2  
diff21(10) → 11  
diff21(21) → 0  
diff21(25) → 8
```

Написать код функции:

```
public int diff21(int n) {  
  
}
```

Дана строка, вернуть новую строку, где последние 3 символа в верхнем регистре (заглавные). Если строка длины меньше 3, перевести это в заглавные.

```
endUp("Hello") → "HeLLO"  
endUp("hi there") → "hi thERE"  
endUp("hi") → "HI"
```

Написать код функции:

```
public String endUp(String str) {  
  
}
```

Даны три целых числа a b c, вернуть наибольшее.

```
intMax(1, 2, 3) → 3  
intMax(1, 3, 2) → 3  
intMax(3, 2, 1) → 3
```

```
public int intMax(int a, int b, int c) {  
  
}
```

Даны два массива целых чисел. Вернуть true если у них одинаковый первый или последний элемент. Оба массива длины 1 или более.

commonEnd([1, 2, 3], [7, 3]) → true
commonEnd([1, 2, 3], [7, 3, 2]) → false
commonEnd([1, 2, 3], [1, 3]) → true

```
public boolean commonEnd(int[] a, int[] b) {  
  
}
```

Дан массив целых чисел. Вернуть сумму двух первых элементов массива. Если длина массива меньше двух, вернуть сумму всех элементов. Вернуть 0, если длина массива равна нулю.

sum2([1, 2, 3]) → 3
sum2([1, 1]) → 2
sum2([1, 1, 1, 1]) → 2

```
public int sum2(int[] nums) {  
  
}
```

Вернуть количество четных чисел в массиве целых чисел. Напомню, что оператор % "mod" вычисляет остаток от деления, e.g. 5 % 2 равно 1.

countEvens([2, 1, 2, 3, 4]) → 3
countEvens([2, 2, 0]) → 3
countEvens([1, 3, 5]) → 0

```
public int countEvens(int[] nums) {  
  
}
```

Вернуть версию входного массива, где каждое нулевое значение заменено максимальным нечетным значением, справа от нуля. Если нечетных чисел справа от нуля нет, то оставляем ноль.

zeroMax([0, 5, 0, 3]) → [5, 5, 3, 3]
zeroMax([0, 4, 0, 3]) → [3, 4, 3, 3]
zeroMax([0, 1, 0]) → [1, 1, 0]

```
public int[] zeroMax(int[] nums) {}
```

Вернуть массив, смещенный влево на один индекс. То есть для {6, 2, 5, 3} вернуть {2, 5, 3, 6}. Можно вернуть измененный данный массив, а можно вернуть новый.

shiftLeft([6, 2, 5, 3]) → [2, 5, 3, 6]

shiftLeft([1, 2]) → [2, 1]

shiftLeft([1]) → [1]

```
public int[] shiftLeft(int[] nums) {
```

```
}
```

Даны две строки, вернуть их конкатенацию, но без первого символа в каждой. Строки ненулевой длины.

nonStart("Hello", "There") → "ellohere"

nonStart("java", "code") → "avaode"

nonStart("shotl", "java") → "hotlava"

```
public String nonStart(String a, String b) {
```

```
}
```

Дана строка четной длины. Вернуть строку из двух центральных символов, например строка "string" превратиться в "ri". Длина входной строки минимум 2.

middleTwo("string") → "ri"

middleTwo("code") → "od"

middleTwo("Practice") → "ct"

```
public String middleTwo(String str) {
```

```
}
```