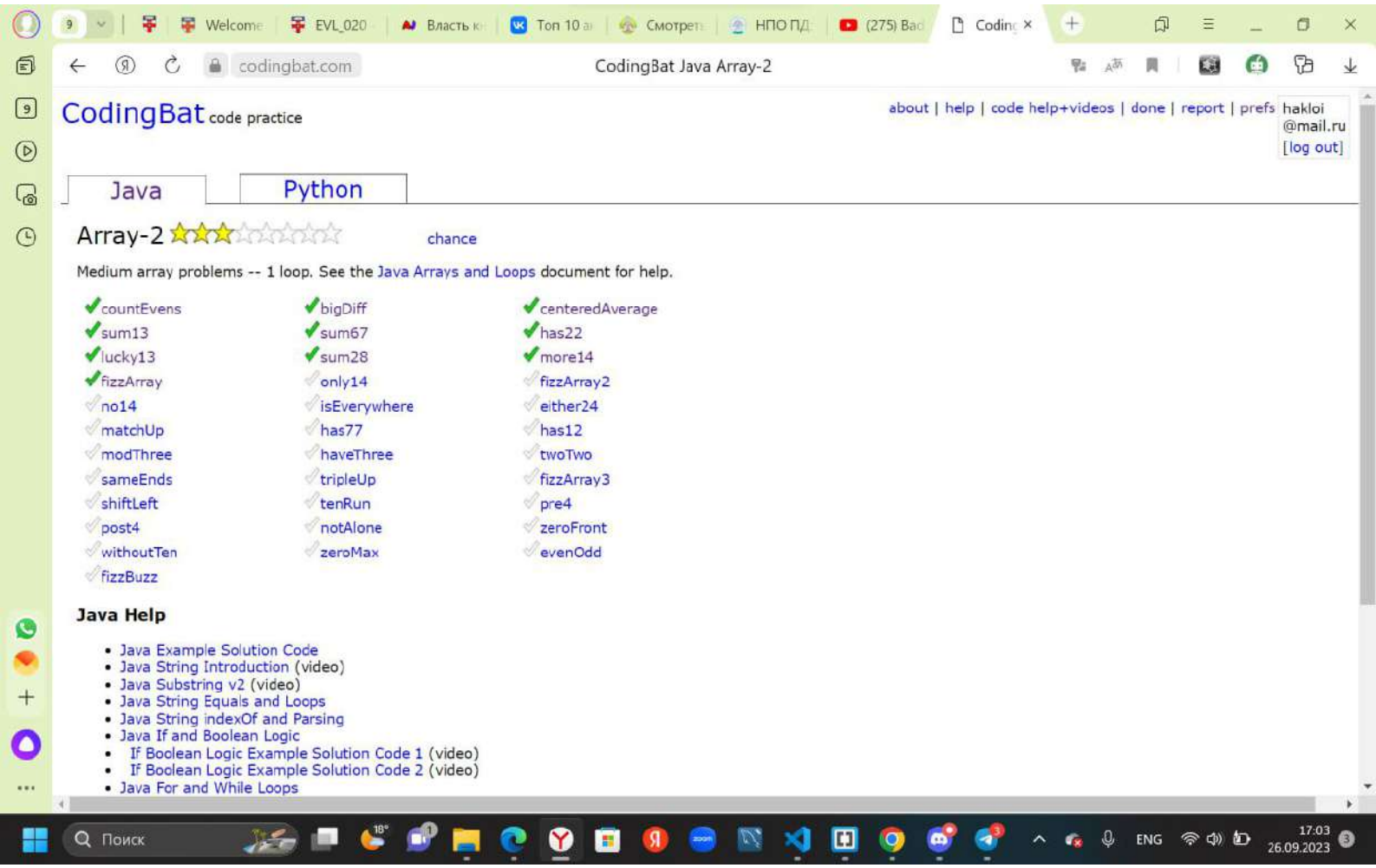
codingbat.com
CodingBat Java Array-2
CodingBat code practice
about | help | code help+videos | done | report | prefs
hakloi
@mail.ru
[log out]
Java
Python
Array-2
chance
Medium array problems -- 1 loop. See the Java Arrays and Loops document for help.
countEvens
bigDiff
centeredAverage
sum13
sum67
has22
lucky13
sum28
more14
fizzArray
only14
fizzArray2
no14
isEverywhere
either24
matchUp
has77
has12
modThree
haveThree
twoTwo
sameEnds
tripleUp
fizzArray3
shiftLeft
tenRun
pre4
post4
notAlone
zeroFront
withoutTen
zeroMax
evenOdd
fizzBuzz
Java Help
Java Example Solution Code
Java String Introduction (video)
Java Substring v2 (video)
Java String Equals and Loops
Java String indexOf and Parsing
Java If and Boolean Logic
If Boolean Logic Example Solution Code 1 (video)
If Boolean Logic Example Solution Code 2 (video)
Java For and While Loops
Поиск
ENG
17:03
26.09.2023

codingbat.com

CodingBat Java Array-2 countEvens

prev | next | chance

Return the number of even ints in the given array. Note: the % "mod" operator computes the remainder, e.g. 5 % 2 is 1.

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

Go ...Save, Compile, Run (ctrl-enter)

```
public int countEvens(int[] nums) {
  int count = 0;
  for (int i = 0; i < nums.length; i++){
    if (nums[i] % 2 == 0) {
      count += 1;
    }
  }
  return count;
}
```

Go

Expected	Run		
countEvens([2, 1, 2, 3, 4]) → 3	3	OK	
countEvens([2, 2, 0]) → 3	3	OK	
countEvens([1, 3, 5]) → 0	0	OK	
countEvens([]) → 0	0	OK	
countEvens([11, 9, 0, 1]) → 1	1	OK	
countEvens([2, 11, 9, 0]) → 2	2	OK	
countEvens([2]) → 1	1	OK	
countEvens([2, 5, 12]) → 2	2	OK	
other tests		OK	

All Correct

next | chance

Java > Array-2

hakloi@mail.ru done page

Your progress graph for this problem

Поиск

ENG 15:19 26.09.2023

codingbat.com

CodingBat Java Array-2 bigDiff

prev | next | chance

Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array. Note: the built-in Math.min(v1, v2) and Math.max(v1, v2) methods return the smaller or larger of two values.

bigDiff([10, 3, 5, 6]) → 7
bigDiff([7, 2, 10, 9]) → 8
bigDiff([2, 10, 7, 2]) → 8

Go ...Save, Compile, Run (ctrl-enter)

```
public int bigDiff(int[] nums) {
  int smallest = nums[0];
  int biggest = nums[0];
  for (int i = 0; i < nums.length; i++){
    if (nums[i] < smallest){
      smallest = nums[i];
    } else if (nums[i] > biggest){
      biggest = nums[i];
    }
  }
  return biggest - smallest;
}
```

Go

Expected	Run		
bigDiff([10, 3, 5, 6]) → 7	7	OK	
bigDiff([7, 2, 10, 9]) → 8	8	OK	
bigDiff([2, 10, 7, 2]) → 8	8	OK	
bigDiff([2, 10]) → 8	8	OK	
bigDiff([10, 2]) → 8	8	OK	
bigDiff([10, 0]) → 10	10	OK	
bigDiff([2, 3]) → 1	1	OK	
bigDiff([2, 2]) → 0	0	OK	
bigDiff([2]) → 0	0	OK	
bigDiff([5, 1, 6, 1, 9, 9]) → 8	8	OK	
bigDiff([7, 6, 8, 5]) → 3	3	OK	
bigDiff([7, 7, 6, 8, 5, 5, 6]) → 3	3	OK	
other tests		OK	

All Correct

next | chance

Java > Array-2

codingbat.com — CodingBat Java Array-2 centeredAverage

the values, except ignoring the largest and smallest values in the array. If there are multiple copies of the smallest value, ignore just one copy, and likewise for the largest value. Use int division to produce the final average. You may assume that the array is length 3 or more.

centeredAverage([1, 2, 3, 4, 100]) → 3
centeredAverage([1, 1, 5, 5, 10, 8, 7]) → 5
centeredAverage([-10, -4, -2, -4, -2, 0]) → -3

Go ...Save, Compile, Run (ctrl-enter)

```
public int centeredAverage(int[] nums) {
  Arrays.sort(nums);
  int count = 0;
  int sum = 0;
  for (int i = 1; i < nums.length - 1; i++){
    sum += nums[i];
    count++;
  }
    return sum / count;
}
```

Go

Editor font size %: 125

Expected	Run		
centeredAverage([1, 2, 3, 4, 100]) → 3	3	OK	
centeredAverage([1, 1, 5, 5, 10, 8, 7]) → 5	5	OK	
centeredAverage([-10, -4, -2, -4, -2, 0]) → -3	-3	OK	
centeredAverage([5, 3, 4, 6, 2]) → 4	4	OK	
centeredAverage([5, 3, 4, 0, 100]) → 4	4	OK	
centeredAverage([100, 0, 5, 3, 4]) → 4	4	OK	
centeredAverage([4, 0, 100]) → 4	4	OK	
centeredAverage([0, 2, 3, 4, 100]) → 3	3	OK	
centeredAverage([1, 1, 100]) → 1	1	OK	
centeredAverage([7, 7, 7]) → 7	7	OK	
centeredAverage([1, 7, 8]) → 7	7	OK	
centeredAverage([1, 1, 99, 99]) → 50	50	OK	
centeredAverage([1000, 0, 1, 99]) → 50	50	OK	
centeredAverage([4, 4, 4, 4, 5]) → 4	4	OK	
centeredAverage([4, 4, 4, 1, 5]) → 4	4	OK	
centeredAverage([6, 4, 8, 12, 3]) → 6	6	OK	
other tests		OK	

All Correct

Поиск ENG 15:44 26.09.2023

codingbat.com

CodingBat Java Array-2 sum13

prev | next | chance

Return the sum of the numbers in the array, returning 0 for an empty array. Except the number 13 is very unlucky, so it does not count and numbers that come immediately after a 13 also do not count.

sum13([1, 2, 2, 1]) → 6
sum13([1, 1]) → 2
sum13([1, 2, 2, 1, 13]) → 6

Go ...Save, Compile, Run (ctrl-enter)

```
public int sum13(int[] nums) {
  int sum = 0;
  for (int i = 0; i < nums.length; i++){
    if (nums[i] != 13){
      sum+=nums[i];
    } else if (nums[i] == 13 && i < nums.length -1 ) {
      nums[i] = 0;
      nums[i+1] = 0;
    }
  }
  return sum;
}
```

Go

Expected	Run		
sum13([1, 2, 2, 1]) → 6	6	OK	
sum13([1, 1]) → 2	2	OK	
sum13([1, 2, 2, 1, 13]) → 6	6	OK	
sum13([1, 2, 13, 2, 1, 13]) → 4	4	OK	
sum13([13, 1, 2, 13, 2, 1, 13]) → 3	3	OK	
sum13([]) → 0	0	OK	
sum13([13]) → 0	0	OK	
sum13([13, 13]) → 0	0	OK	
sum13([13, 0, 13]) → 0	0	OK	
sum13([13, 1, 13]) → 0	0	OK	
sum13([5, 7, 2]) → 14	14	OK	
sum13([5, 13, 2]) → 5	5	OK	
sum13([0]) → 0	0	OK	
sum13([13, 0]) → 0	0	OK	
other tests		OK	

All Correct

codingbat.com — CodingBat Java Array-2 sum67

Array-2 > sum67

prev | next | chance

Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 7 (every 6 will be followed by at least one 7). Return 0 for no numbers.

sum67([1, 2, 2]) → 5
sum67([1, 2, 2, 6, 99, 99, 7]) → 5
sum67([1, 1, 6, 7, 2]) → 4

Go ...Save, Compile, Run (ctrl-enter)

```
public int sum67(int[] nums) {
  int sum = 0;
  boolean breakk = false;

  for (int i = 0; i < nums.length; i++) {
    if (nums[i] == 6){
      breakk = true; //break if nums[i] ==6
    }
    if (breakk == false){ //continue
      sum += nums[i];
    }
    if (nums[i] == 7 && breakk == true){
      breakk = false;
    }
  }
  return sum;
}
```

	Expected	Run		
sum67([1, 2, 2]) → 5	5	OK	
sum67([1, 2, 2, 6, 99, 99, 7]) → 5	5	OK	
sum67([1, 1, 6, 7, 2]) → 4	4	OK	
sum67([1, 6, 2, 2, 7, 1, 6, 99, 99, 7]) → 2	2	OK	
sum67([1, 6, 2, 6, 2, 7, 1, 6, 99, 99, 7]) → 2	2	OK	
sum67([2, 7, 6, 2, 6, 7, 2, 7]) → 18	18	OK	
sum67([2, 7, 6, 2, 6, 2, 7]) → 9	9	OK	
sum67([1, 6, 7, 7]) → 8	8	OK	
sum67([6, 7, 1, 6, 7, 7]) → 8	8	OK	
sum67([6, 8, 1, 6, 7]) → 0	0	OK	
sum67([]) → 0	0	OK	
sum67([6, 7, 11]) → 11	11	OK	
sum67([11, 6, 7, 11]) → 22	22	OK	
sum67([2, 2, 6, 7, 7]) → 11	11	OK	
other tests		OK	

All Correct

Поиск ENG 16:11 26.09.2023

codingbat.com — CodingBat Java Array-2 has22

Given an array of ints, return true if the array contains a 2 next to a 2 somewhere.

has22([1, 2, 2]) → true
has22([1, 2, 1, 2]) → false
has22([2, 1, 2]) → false

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean has22(int[] nums) {
  boolean bool = false;
  for (int i = 0; i < nums.length; i++) {
    if (nums[i] == 2 && i > 0 &&
        nums[i-1] == 2){
      bool = true;
    }
    if (nums[i] == 2 && i < nums.length-1 &&
        nums[i+1] == 2){
      bool = true;
    }
  }
  return bool;
}
```

Expected	Run		
has22([1, 2, 2]) → true	true	OK	
has22([1, 2, 1, 2]) → false	false	OK	
has22([2, 1, 2]) → false	false	OK	
has22([2, 2, 1, 2]) → true	true	OK	
has22([1, 3, 2]) → false	false	OK	
has22([1, 3, 2, 2]) → true	true	OK	
has22([2, 3, 2, 2]) → true	true	OK	
has22([4, 2, 4, 2, 2, 5]) → true	true	OK	
has22([1, 2]) → false	false	OK	
has22([2, 2]) → true	true	OK	
has22([2]) → false	false	OK	
has22([]) → false	false	OK	
has22([3, 3, 2, 2]) → true	true	OK	
has22([5, 2, 5, 2]) → false	false	OK	
other tests		OK	

Go

All Correct

Editor font size %: 125

codingbat.com — CodingBat Java Array-2 lucky13

Array-2 > lucky13

prev | next | chance

Given an array of ints, return true if the array contains no 1's and no 3's.

lucky13([0, 2, 4]) → true
lucky13([1, 2, 3]) → false
lucky13([1, 2, 4]) → false

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean lucky13(int[] nums) {
  boolean bool = true;
  for(int i = 0; i < nums.length; i++){
    if (nums[i] == 1 || nums[i] == 3){
      bool = false;
    }
  }
  return bool;
}
```

Go

Expected	Run		
lucky13([0, 2, 4]) → true	true	OK	
lucky13([1, 2, 3]) → false	false	OK	
lucky13([1, 2, 4]) → false	false	OK	
lucky13([2, 7, 2, 8]) → true	true	OK	
lucky13([2, 7, 1, 8]) → false	false	OK	
lucky13([3, 7, 2, 8]) → false	false	OK	
lucky13([2, 7, 2, 1]) → false	false	OK	
lucky13([1, 2]) → false	false	OK	
lucky13([2, 2]) → true	true	OK	
lucky13([2]) → true	true	OK	
lucky13([3]) → false	false	OK	
lucky13([]) → true	true	OK	
other tests		OK	

All Correct

next | chance

Поиск ENG 16:36 26.09.2023

codingbat.com — CodingBat Java Array-2 sum28

Array-2 > sum28

prev | next | chance

Given an array of ints, return true if the sum of all the 2's in the array is exactly 8.

sum28([2, 3, 2, 2, 4, 2]) → true
sum28([2, 3, 2, 2, 4, 2, 2]) → false
sum28([1, 2, 3, 4]) → false

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean sum28(int[] nums) {
  int sum = 0;
  boolean bool = false;
  for (int i = 0; i < nums.length; i++){
    if (nums[i] == 2){
      sum += 2;
    }
  }
    if (sum == 8 ){
      bool = true;
    }
  return bool;
}
```

Go

Expected	Run		
sum28([2, 3, 2, 2, 4, 2]) → true	true	OK	
sum28([2, 3, 2, 2, 4, 2, 2]) → false	false	OK	
sum28([1, 2, 3, 4]) → false	false	OK	
sum28([2, 2, 2, 2]) → true	true	OK	
sum28([1, 2, 2, 2, 2, 4]) → true	true	OK	
sum28([]) → false	false	OK	
sum28([2]) → false	false	OK	
sum28([8]) → false	false	OK	
sum28([2, 2, 2]) → false	false	OK	
sum28([2, 2, 2, 2, 2]) → false	false	OK	
sum28([1, 2, 2, 1, 2, 2]) → true	true	OK	
sum28([5, 2, 2, 2, 4, 2]) → true	true	OK	
other tests		OK	

All Correct

next | chance

Поиск ENG 16:52 26.09.2023

codingbat.com

CodingBat Java Array-2 more14

Given an array of ints, return true if the number of 1's is greater than the number of 4's

more14([1, 4, 1]) → true
more14([1, 4, 1, 4]) → false
more14([1, 1]) → true

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean more14(int[] nums) {
  int count1 = 0;
  int count4 = 0;
  boolean bool = false;
  for (int i = 0; i < nums.length; i++){
    if (nums[i] == 1){
      count1 += 1;
    } else if(nums[i] == 4){
      count4 += 1;
    }
  }
    if (count1 > count4 ){
      bool = true;
    }
  return bool;
}
```

Go

Editor font size %: 125
Shorter output

Expected	Run		
more14([1, 4, 1]) → true	true	OK	
more14([1, 4, 1, 4]) → false	false	OK	
more14([1, 1]) → true	true	OK	
more14([1, 6, 6]) → true	true	OK	
more14([1]) → true	true	OK	
more14([1, 4]) → false	false	OK	
more14([6, 1, 1]) → true	true	OK	
more14([1, 6, 4]) → false	false	OK	
more14([1, 1, 4, 4, 1]) → true	true	OK	
more14([1, 1, 6, 4, 4, 1]) → true	true	OK	
more14([]) → false	false	OK	
more14([4, 1, 4, 6]) → false	false	OK	
more14([4, 1, 4, 6, 1]) → false	false	OK	
more14([1, 4, 1, 4, 1, 6]) → true	true	OK	
other tests		OK	

All Correct

next | chance

Поиск ENG 16:59 26.09.2023

codingbat.com — CodingBat Java Array-2 fizzArray

Java | Python

Array-2 > fizzArray

prev | next | chance

Given a number n, create and return a new int array of length n, containing the numbers 0, 1, 2, ... n-1. The given n may be 0, in which case just return a length 0 array. You do not need a separate if-statement for the length-0 case; the for-loop should naturally execute 0 times in that case, so it just works. The syntax to make a new int array is: new int[desired_length] (See also: FizzBuzz Code)

fizzArray(4) → [0, 1, 2, 3]
fizzArray(1) → [0]
fizzArray(10) → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Go ...Save, Compile, Run (ctrl-enter)

```
public int[] fizzArray(int n) {
  int[] arr = new int[n];
    for (int i = 0; i < n; i++)
        arr[i] = i;
    return arr;
}
```

Expected	Run		
fizzArray(4) → [0, 1, 2, 3]	[0, 1, 2, 3]	OK	
fizzArray(1) → [0]	[0]	OK	
fizzArray(10) → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	OK	
fizzArray(0) → []	[]	OK	
fizzArray(2) → [0, 1]	[0, 1]	OK	
fizzArray(7) → [0, 1, 2, 3, 4, 5, 6]	[0, 1, 2, 3, 4, 5, 6]	OK	
other tests		OK	

All Correct

next | chance

Java > Array-2

hakloi@mail.ru done page

Your progress graph for this problem

Поиск ENG 17:02 26.09.2023