AP-1: scoresAverage

https://drive.google.com/open?id=1F2L2AWj_Sxxr4pkwhTySwPWtfVbHE51S
Link to screen recording

```
public int scoresAverage(int[] scores) {
  int firstHalf = average(scores, 0, scores.length/2);
  int secondHalf = average(scores, scores.length/2, scores.length);
  if(firstHalf>secondHalf)
    return firstHalf;
  else
    return secondHalf;
}
public int average(int[] scores, int start, int end)
{
  int count = 0;
  int total = 0;
  int average = 0;
  for(int i = start; i < end; i++)
  {
    total += scores[i];
    count++;
  }
  average = total/count;
  return average;
}
```

AP-1 > scoresAverage

Given an array of scores, compute the int average of the first half and the second half, and return whichever is larger. We'll say that the second half begins at index length/2. The array length will be at least 2. To practice decomposition, write a separate helper method `int average(int[] scores, int start, int end) {` which computes the average of the elements between indexes start..end. Call your helper method twice to implement scoresAverage(). Write your helper method after your scoresAverage() method in the JavaBat text area. Normally you would compute averages with doubles, but here we use ints so the expected results are exact.

scoresAverage([2, 2, 4, 4]) → 4
scoresAverage([4, 4, 4, 2, 2, 2]) → 4
scoresAverage([3, 4, 5, 1, 2, 3]) → 4

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

```java
public int scoresAverage(int[] scores) {
  int firstHalf = average(scores, 0, scores.length/2);
  int secondHalf = average(scores, scores.length/2, scores.length);
  if(firstHalf>secondHalf)
    return firstHalf;
  else
    return secondHalf;
}
public int average(int[] scores, int start, int end)
{
  int count = 0;
  int total = 0;
  int average = 0;
  for(int i = start; i < end; i++)
  {
    total += scores[i];
    count++;
  }
  average = total/count;
  return average;
}
```

Go

Expected	Run		
scoresAverage([2, 2, 4, 4]) → 4	4	OK	🟩
scoresAverage([4, 4, 4, 2, 2, 2]) → 4	4	OK	🟩
scoresAverage([3, 4, 5, 1, 2, 3]) → 4	4	OK	🟩
scoresAverage([5, 6]) → 6	6	OK	🟩
scoresAverage([5, 4]) → 5	5	OK	🟩
scoresAverage([5, 4, 5, 6, 2, 1, 2, 3]) → 5	5	OK	🟩
other tests		OK	🟩

✓ All Correct

next | chance

Java > AP-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

AP-1: dividesSelf

https://drive.google.com/open?id=1fLOApHD7zE7I0xssejkNRbRdoe6R9RTU
Link to recording

```
public boolean dividesSelf(int n) {
  String str = Integer.toString(n);

  for(int i = 0; i < str.length(); i++)
  {
    int a = Character.getNumericValue(str.charAt(i));
    if(a == 0)
      return false;
    if(n % a != 0)
      return false;
  }
  return true;
}
```

CodingBat code practice

Java | Python

AP-1 > dividesSelf

prev | next | chance

We'll say that a positive int divides itself if every digit in the number divides into the number evenly. So for example 128 divides itself since 1, 2, and 8 all divide into 128 evenly. We'll say that 0 does not divide into anything evenly, so no number with a 0 digit divides itself. Note: use % to get the rightmost digit, and / to discard the rightmost digit.

dividesSelf(128) → true
dividesSelf(12) → true
dividesSelf(120) → false

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

```java
public boolean dividesSelf(int n) {
  String str = Integer.toString(n);

  for(int i = 0; i < str.length(); i++)
  {
    int a = Character.getNumericValue(str.charAt(i));
    if(a == 0)
      return false;
    if(n % a != 0)
      return false;
  }
  return true;
}
```

Go

Editor font size %: 100

Shorter output

Expected	Run		
dividesSelf(128) → true	true	OK	🟩
dividesSelf(12) → true	true	OK	🟩
dividesSelf(120) → false	false	OK	🟩
dividesSelf(122) → true	true	OK	🟩
dividesSelf(13) → false	false	OK	🟩
dividesSelf(32) → false	false	OK	🟩
dividesSelf(22) → true	true	OK	🟩
dividesSelf(42) → false	false	OK	🟩
dividesSelf(212) → true	true	OK	🟩
dividesSelf(213) → false	false	OK	🟩
dividesSelf(162) → true	true	OK	🟩
other tests		OK	🟩

✓ All Correct

next | chance

Java > AP-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem