FirstLast6

For(I = 0; I < nums.length; i++) {

If(nums[0] == 6 || nums[nums.length-1] == 6) {

Return true;

} else {

Return false;

Same First Last

If length >= 1 and first & last element are equal 🡪 return true

Otherwise 🡪 return false;

Common end

If length a & b >= 1 and a[0] == b[0] || a[length-1] == b[length-1] 🡪 return true

Otherwise 🡪 return false

RotateLeft3

* Create new array of length 3
* Reassign value of each index.
  + Temp[0] = index[1]
  + Temp[1] = index[2]
  + Temp[2] = index[0]
* Return new array

Count evens

countEven = 0;

For(int I = 0; I < nums.length; i++) {

If (nums[i] % 2 == 1) 🡪 return countEven++

Big Diff

Int largest = nums[0];

For(int num : nums){

If(largest < num){

Largest = num;

}

}

Return largest;

}

Sum13

Int sum = 0;

For (int I = 0; I < nums.length; i++) {

If(nums[i] == 13 & nums[i+1]) {

Continue;

Else sum += nums[i]

Centered Average

1. Check if array length >= 3
2. Check the largest by using for each loop
3. Check the largest by using for each loop
4. If the largest has more than one 🡪 ignore one
5. If the smallest has more than one 🡪 ignore one
6. Sum of the left-over index/left over index length

Int largest = nums[0];

Int smallest = nums[0];

Int sum = 0;

Int average = 0;

For(int i = 0; I < nums.length; i++) {

If(largest < num) {

I++;

}

If(smallest > num) {

I++;

}

Sum += nums[i]

}

Average = sum/nums.length

Return average

Final Project

***4***0***0***3***6***0***0***0***0***0***0***0***0***0***1***4

AMEX 🡪 34 OR 37 and has 15 digits

VISA 🡪 starts with 4 and has 13 and 16 digits

MASTER CARD 🡪 starts with 51,52, 53, 54, 55 and has 16 digits

Long creNum = 0;

(I = Length – 2; I >= 0; i--)

User input will be “long” type.

1. If (cardNum.length() = 13 or 15 or 16🡪multiply each index by 2 ( I \*2 )🡪 add multiple result (I + i) ;

1•2 + 0•2 + 0•2 + 0•2 + 0•2 + 6•2 + 0•2 + 4•2

1. If (index I % 2 == 1) 🡪 add with result from number 1.

2 + 0 + 0 + 0 + 0 + 12 + 0 + 8

1. Add products digits together.

2 + 0 + 0 + 0 + 0 + 1 + 2 + 0 + 8 = 13

1. add that sum (13) to the sum of the digits that weren’t multiplied by 2 (starting from the end)

13 + 4 + 0 + 0 + 0 + 0 + 0 + 3 + 0 = 20

1. the last digit in that sum (20) is a 0, so John’s card is legit

for(long I = 0; I < cardNum.length(); i++) {

if(cardNum(i) % 2 == 0) {

System.out.println(

Char[] strNumber = str.toCharArray();

If(int number.length() == 15 && number.substring(number.length)-1 && number.substring(0,2) == 34 || number.substring(0,1) == 37) {

System.out.println(“AMEX”)

If(number.length() == 16 && number.substring(number.length)-1 == 0 && number.substring(0,1) == 4 || number.substring(number.length) == 13