Problem J. Lucky Tickets

A ticket is considered lucky if in its n-digit number the sum of the first [n / 2] digits is equal to the sum [n / 2] of the last digits (for odd n, the central digit does not participate in the “luck test” and can be any).
Count the number of lucky tickets with different n-digit numbers (leading zeros in numbers are possible, but a number consisting of all zeros does not exist).

23532 lucky

Input data
The input to the program is a natural number n <16.

4774 lucky

Output
Print the number of n-digit lucky tickets.

Examples of
input data
1

output
9 (1,2,3,4,5,6,7,8,9)

input data
2

output
9 (11,22,33,44,55,66,77,88,99)

input data
3
…
(121,…

n=4

1221 Lucky
1212 Lucky
1213 not Lucky

```
void isItLucky() {
    int n;
    int number;
    int reverse;
    int leftover;
    int count;

    cout << "Enter a number: ";
    cin >> n;
    int degree = 1;

    while (n > 0) {
        n--;
        degree *= 10;
    }
    int start = degree / 10;
    int finish = degree - 1;
    count = 0;

    for (int i =  start; i <= finish; i++) {
        reverse = 0;
        leftover = 0;
        number = i;
        while(number != 0) {
            leftover = number % 10;
            reverse = reverse * 10 + leftover;
            number = number / 10;
        }
        if (reverse == i) {
            cout << i << endl;
            count++;
        }
    }
    cout << "Count: " << count;
}
```

ONLY SIMMETRIC

```
void isItLucky() {
    int n;
    int number;
    int reverse;
    int leftover;
    int count;
    int sumLeft;
    int sumRight;

    cout << "Enter a number: ";
    cin >> n;
    int degree = 1;

    int copyN = n;

    while (n > 0) {
        n--;
        degree *= 10;
    }
    int start = degree / 10;
    int finish = degree - 1;
    count = 0;
    int amount = 0;

    for (int i =  start; i <= finish; i++) {  // i = 988
        number = i;
        count = 0;
        sumLeft = 0;
        sumRight = 0;
        while (count < copyN) {   // example: 988
            if (count < (copyN /2)) { //
                sumRight += number % 10;  //
                number /= 10; //
            }
            else if (count >= (copyN + 1)/ 2) {
                sumLeft += number % 10;
                number /= 10;
            }
            else { // need to add this so it moves forward regardless
                number /= 10;
            }
            count++;
        }
        if (sumLeft == sumRight) {
            amount++;
            cout << i << endl;
        }
    }
    cout << "Amount: " << amount;
}
```