**Q. Difference of the sum of alternate digits (D)**

Given a positive integer 'x' (with even number of digits in it), write an algorithm and the subsequent code to compute the difference between the sum of the digits occurring in the alternate positions (starting from the first position) and the sum of the digits occurring in the alternate positions, starting from the last rightmost position of 'x'

For example, consider the number 8975. The sum of the digits that occur in the alternate positions from the first position is 8+7=15. The sum of the digits that occur in the alternate positions, starting from the rightmost position is 5+9 = 14. Difference between the two sums is 1 (=15-14). Similarly, for the number 5798, the difference between two sums, is 1.

Note: Read the input as a number and do entire processing as a number

C++ compilers can compile C code also

Input format:

First line contains the positive integer

Output format:

First line should contain the difference between the sum of the digits occurring in the alternate positions (starting from the first position) and the sum of the digits occurring in the alternate positions (starting from the last rightmost position).

**Python Program**

n = int(input())  
l1 = []; l2 = []  
dummy = n  
counter = 0  
while dummy > 0:  
 dummy //= 10  
 counter += 1  
dummy = n; i = 0  
if counter%2==0:  
 while dummy > 0:  
 if i%2==0:  
 l1.append(dummy%10)  
 else:  
 l2.append(dummy%10)  
 i += 1  
 dummy //= 10  
print(abs(sum(l1) - sum(l2)))