Given a string ‘str’ of digits, find length of the longest substring of ‘str’, such that the length of the substring is 2k digits and sum of left k digits is equal to the sum of right k digits.

Examples:

```
Input: str = "123123"
Output: 6
The complete string is of even length and sum of first and second
half digits is same

Input: str = "1538023"
Output: 4
The longest substring with same first and second half sum is "5380"
```

A **Simple Solution** is to check every substring of even length. The following is C based implementation of simple approach.

```
// A simple C based program to find length of longest  even length
// substring with same sum of digits in left and right
#include<stdio.h>
#include<string.h>

int findLength(char *str)
{
    int n = strlen(str);
    int maxlen =0;  // Initialize result

    // Choose starting point of every substring
    for (int i=0; i<n; i++)
    {
        // Choose ending point of even length substring
        for (int j =i+1; j<n; j += 2)
        {
            int length = j-i+1;//Find length of current substr

            // Calculate left & right sums for current substr
            int leftsum = 0, rightsum =0;
            for (int k =0; k<length/2; k++)
            {
                leftsum  += (str[i+k]-'0');
                rightsum += (str[i+k+length/2]-'0');
            }

            // Update result if needed
            if (leftsum == rightsum && maxlen < length)
                    maxlen = length;
        }
    }
    return maxlen;
}

// Driver program to test above function
int main(void)
{
    char str[] = "1538023";
    printf("Length of the substring is %d", findLength(str));
    return 0;
}
```

Output:

```
Length of the substring is 4
```

The time complexity of above solution is $O(n^3)$. The above solution can be optimized to work in $O(n^2)$ using **Dynamic Programming**. The idea is to build a 2D table that stores sums of substrings. The following is C based implementation of Dynamic Programming approach.

```
// A C based program that uses Dynamic Programming to find length of the
// longest even substring with same sum of digits in left and right half
#include <stdio.h>
#include <string.h>

int findLength(char *str)
{
    int n = strlen(str);
    int maxlen = 0; // Initialize result

    // A 2D table where sum[i][j] stores sum of digits
    // from str[i] to str[j].  Only filled entries are
    // the entries where j >= i
    int sum[n][n];

    // Fill the diagonal values for sunstrings of length 1
    for (int i =0; i<n; i++)
        sum[i][i] = str[i]-'0';

    // Fill entries for substrings of length 2 to n
    for (int len=2; len<=n; len++)
    {
        // Pick i and j for current substring
        for (int i=0; i<n-len+1; i++)
        {
            int j = i+len-1;
            int k = len/2;

            // Calculate value of sum[i][j]
            sum[i][j] = sum[i][j-k] + sum[j-k+1][j];

            // Update result if 'len' is even, left and right
            // sums are same and len is more than maxlen
            if (len%2 == 0 && sum[i][j-k] == sum[(j-k+1)][j]
                           && len > maxlen)
                maxlen = len;
        }
    }
    return maxlen;
}

// Driver program to test above function
int main(void)
{
    char str[] = "153803";
    printf("Length of the substring is %d", findLength(str));
    return 0;
}
```

Output:

```
Length of the substring is 4
```

Time complexity of the above solution is $O(n^2)$, but it requires $O(n^2)$ extra space.