

# Computer Memory

---

## Description

You have been given a very strange computer. Its memory consists of several bits, **each initially set to 0**, and it can only perform the following type of operation: Choose one of the bits in memory, and choose a value: 0 or 1. All the bits between the selected bit and the last bit in memory, inclusive, will be set to the chosen value. For example, if the memory is "0010", and you choose the second bit and a value of 1, the memory will change to "0111".

You are given a String *mem*. The number of characters in *mem* is equal to the number of bits in the computer's memory. *mem* will contain only the characters '0' (zero) or '1' (one).

Return the minimum number of operations required to set the computer's memory equal to *mem* in **MOST EFFICIENT WAY**.

## Complexity

Complexity of your algorithm should be **less than  $O(N^2)$**

**Function:** **Implement it!**

```
int RequiredFunction(string mem)
```

PROBLEM\_CLASS.cs includes this method.

## Example

1) *mem* = "0011"

Returns: 1

0000 -> 0011

Choose the third bit and a value of 1.

2) *mem* = "000"

Returns: 0

No operations are needed.

3) *mem* = "0100"

Returns: 2

0000 -> 0111 -> 0100

3) *mem* = "111000111"

Returns: 3

000000000 -> 111111111 -> 111000000 -> 111000111

## C# Help

### Getting the size of 1D array

```
int size = array1D.GetLength(0);
```

### Getting the size of 2D array

```
int size1 = array2D.GetLength(0);
```

```
int size2 = array2D.GetLength(1);
```

### Creating 1D array

```
int [] array1D = new int [size]
```

### Creating 2D array

```
int [,] array2D = new int [size1, size2]
```

### Sorting single array

Sort the given array "items" in ascending order

```
Array.Sort(items);
```

### Sorting parallel arrays

Sort the first array "master" and re-order the 2<sup>nd</sup> array "slave" according to this sorting

```
Array.Sort(master, slave);
```