Caesar Cipher: Encryption

by vatsalchanana

Problem Submissions Leaderboard Discussions Editorial

Julius Caesar protected his confidential information by encrypting it in a cipher. Caesar's cipher rotated every letter in a string by a fixed number, K, making it unreadable by his enemies. Given a string, S, and a number, K, encrypt S and print the resulting string.

Note: The cipher *only* encrypts letters; symbols, such as `-`, remain unencrypted.

Input Format

The first line contains an integer, N, which is the length of the unencrypted string.
The second line contains the unencrypted string, S.
The third line contains the integer encryption key, K, which is the number of letters to rotate.

Constraints
$1 \leq N \leq 100$
$0 \leq K \leq 100$
S is a valid ASCII string and doesn't contain any spaces.

Output Format

For each test case, print the encoded string.

Sample Input

```
11
middle-Outz
2
```

Sample Output

```
okffng-Qwvb
```

Explanation

Each unencrypted letter is replaced with the letter occurring K spaces after it when listed alphabetically. Think of the alphabet as being both case-sensitive and circular; if K rotates past the end of the alphabet, it loops back to the beginning (i.e.: the letter after z is a, and the letter after Z is A).

Selected Examples:
m (ASCII 109) becomes o (ASCII 111).
i (ASCII 105) becomes k (ASCII 107).
$-$ remains the same, as symbols are not encoded.
O (ASCII 79) becomes Q (ASCII 81).
z (ASCII 122) becomes b (ASCII 98); because z is the last letter of the alphabet, a (ASCII 97) is the next letter after it in lower-case rotation.

Submissions: 3583
Max Score: 40
Difficulty: Easy

Rate This Challenge:
★★★★★ Thanks!
More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static string Caesar(string text, int shift)
        {
            string alfab = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz";
            string mayus = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ";
            shift = shift % 26;
            string ans = "";
            for (int i = 0; i < text.Length; i++)
            {
                // ans += (alfab.IndexOf(text[i]) + shift).ToString().ToUpper();
                //ans += alfab[ text.IndexOf(i)+shift];
                if (char.IsLetter(text[i]))
                {
                    if (char.IsUpper(text[i]))
                    {
                        int indmayus = mayus.IndexOf(text[i]);
                        ans += mayus[indmayus + shift].ToString();
                    }
                    else
                    {
                        int indalfab = alfab.IndexOf(text[i]);
                        ans += alfab[indalfab + shift].ToString();
                    }
                }
                else
                {
                    ans += text[i];
                }
            }
            return ans;
        }
        static void Main(string[] args)
        {
            //int n = 11;
            //string s= "middle-Outz";
            //int k = 2;
            int n = int.Parse(Console.ReadLine());
            string s = Console.ReadLine();
            int k = int.Parse(Console.ReadLine());
            Console.WriteLine(Caesar(s, k));
            // Console.ReadLine();
        }
}

```

Line: 47 Col: 10

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5

✔ Test Case #6 ✔ Test Case #7 ✔ Test Case #8

✔ Test Case #9 ✔ Test Case #10

Next Challenge