

Caesar Cipher

★

Problem

Submissions

Leaderboard

Discussions

Editorial

Julius Caesar protected his confidential information by encrypting it using a cipher. [Caesar's cipher](#) shifts each letter by a number of letters. If the shift takes you past the end of the alphabet, just rotate back to the front of the alphabet. In the case of a rotation by 3, w, x, y and z would map to z, a, b and c.

Original alphabet: abcdefghijklmnopqrstuvwxyz
Alphabet rotated +3: defghijklmnopqrstuvwxyzabc

Example

***s* = There's-a-starman-waiting-in-the-sky**

***k* = 3**

The alphabet is rotated by **3**, matching the mapping above. The encrypted string is **Wkhuh'v-d-vwdupdq-zdlwlqj-lq-wkh-vnb.**

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

Function Description

Complete the `caesarCipher` function in the editor below.

`caesarCipher` has the following parameter(s):

- string `s`: cleartext
- int `k`: the alphabet rotation factor

Returns

- string: the encrypted string

Input Format

The first line contains the integer, ***n***, the length of the unencrypted string.

The second line contains the unencrypted string, ***s***.

The third line contains ***k***, the number of letters to rotate the alphabet by.

Constraints

$1 \leq n \leq 100$

$0 \leq k \leq 100$

s is a valid ASCII string without any spaces.

Sample Input

```
11
middle-Outz
2
```

Sample Output

```
okffng-Qwvb
```

Explanation

Original alphabet: abcdefghijklmnopqrstuvwxyz
Alphabet rotated +2: cdefghijklmnopqrstuvwxyzab

```
m -> o
i -> k
d -> f
d -> f
```

Author	vatsalchanana
Difficulty	Easy
Max Score	100
Submitted By	6176

NEED HELP?

- View discussions
- View editorial
- View top submissions

RATE THIS CHALLENGE



MORE DETAILS

- Download problem statement
- Download sample test cases
- Suggest Edits



```
l->n
e->g
- -
O->Q
u->w
t->v
z->b
```

[Change Theme](#)

Language

C



```
1  #include <assert.h>
2  #include <ctype.h>
3  #include <limits.h>
4  #include <math.h>
5  #include <stdbool.h>
6  #include <stddef.h>
7  #include <stdint.h>
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11
12 char* readline();
13 char* ltrim(char*);
14 char* rtrim(char*);
15
16 int parse_int(char*);
17
18 /*
19  * Complete the 'caesarCipher' function below.
20  *
21  * The function is expected to return a STRING.
22  * The function accepts following parameters:
23  * 1. STRING s
24  * 2. INTEGER k
25  */
26
27 /*
28  * To return the string from the function, you should either do static allocation or dynamic
29  * allocation
30  * For example,
31  * char* return_string_using_static_allocation() {
32  *     static char s[] = "static allocation of string";
33  *
34  *     return s;
35  * }
36  *
37  * char* return_string_using_dynamic_allocation() {
38  *     char* s = malloc(100 * sizeof(char));
39  *
40  *     s = "dynamic allocation of string";
41  *
42  *     return s;
43  * }
44  *
45  */
46 char* caesarCipher(char* s, int k) {
47
48 }
49
```

Line: 167 Col: 1

 Upload Code as File

☐ Test against custom input

Run Code

Submit Code

[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)