SPOJ Problem Set (classical)

899. Ws Cipher

Problem code: WSCIPHER

Weird Wally's Wireless Widgets, Inc. manufactures an eclectic assortment of small, wireless, network capable devices, ranging from dog collars, to pencils, to fishing bobbers. All these devices have very small memories. Encryption algorithms like Rijndael, the candidate for the Advanced Encryption Standard (AES) are demonstrably secure but they don't fit in such a tiny memory. In order to provide some security for transmissions to and from the devices, WWWW uses the following algorithm, which you are to implement.

Encrypting a message requires three integer keys, k_1, k_2, and k_3. The letters [a-i] form one group, [j-r] a second group, and everything else ([s-z] and underscore) the third group. Within each group the letters are rotated *left* by k_i positions in the message. Each group is rotated independently of the other two. Decrypting the message means doing a *right* rotation by k_i positions within each group.

Consider the message `the_quick_brown_fox` encrypted with k_i values of 2, 3 and 1. The encrypted string is `_icuo_bfnwhoq_kxert`. The figure below shows the decrypting right rotations for one character in each of the three character groups.

[IMAGE]

Looking at all the letters in the group [a-i] we see {`i,c,b,f,h,e`} appear at positions {2,3,7,8,11,17} within the encrypted message. After a right rotation of k_1=2, these positions contain the letters {`h,e,i,c,b,f`}. The table below shows the intermediate strings that come from doing all the rotations in the first group, then all rotations in the second group, then all the rotations in the third group. Rotating letters in one group will not change any letters in any of the other groups.

	[a-i], $k_1 = 2$	**[j-r], $k_2 = 3$**	**[s-z] and _, $k_3 = 1$**
Encrypted:	`_icuo_bfnwhoq_kxert`	`_heuo_icnwboq_kxfrt`	`_heuq_ickwbro_nxfot`
Decrypted:	`_heuo_icnwboq_kxfrt`	`_heuq_ickwbro_nxfot`	`the_quick_brown_fox`
Changes:	` ^^   ^^  ^     ^`	`    ^   ^  ^^ ^  ^`	`^  ^ ^   ^   ^ ^  ^`

All input strings contain only lowercase letters and underscores(_). Each string will be at most 80 characters long. The k_i are all positive integers in the range 1-100.

Input consists of information for one or more encrypted messages. Each problem begins with one line containing k_1, k_2, and k_3 followed by a line containing the encrypted message. The end of the input is signalled by a line with all key values of 0.

For each encrypted message, the output is a single line containing the decrypted string.

Input:

```
2 3 1
_icuo_bfnwhoq_kxert
1 1 1
bcalmkyzx
3 7 4
wcb_mxfep_dorul_eov_qtkrhe_ozany_dgtoh_u_eji
2 4 3
cjvdksaltbmu
0 0 0
```

Output:

```
the_quick_brown_fox
abcklmxyz
the_quick_brown_fox_jumped_over_the_lazy_dog
ajsbktcludmv
```

Added by:	Wanderley Guimaraes
Date:	2006-06-09
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	ACM Mid Central Regionals 2001