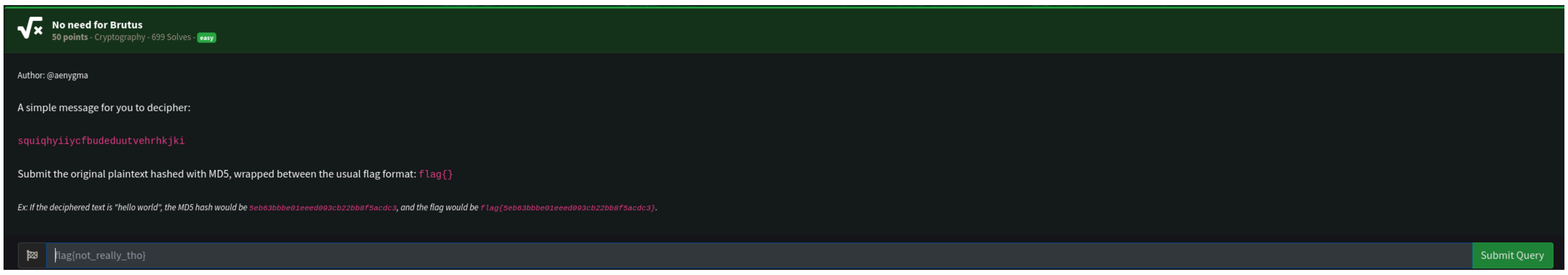
No Need For Brutus



No need for Brutus
50 points - Cryptography - 699 Solves - easy

Author: @aenygma

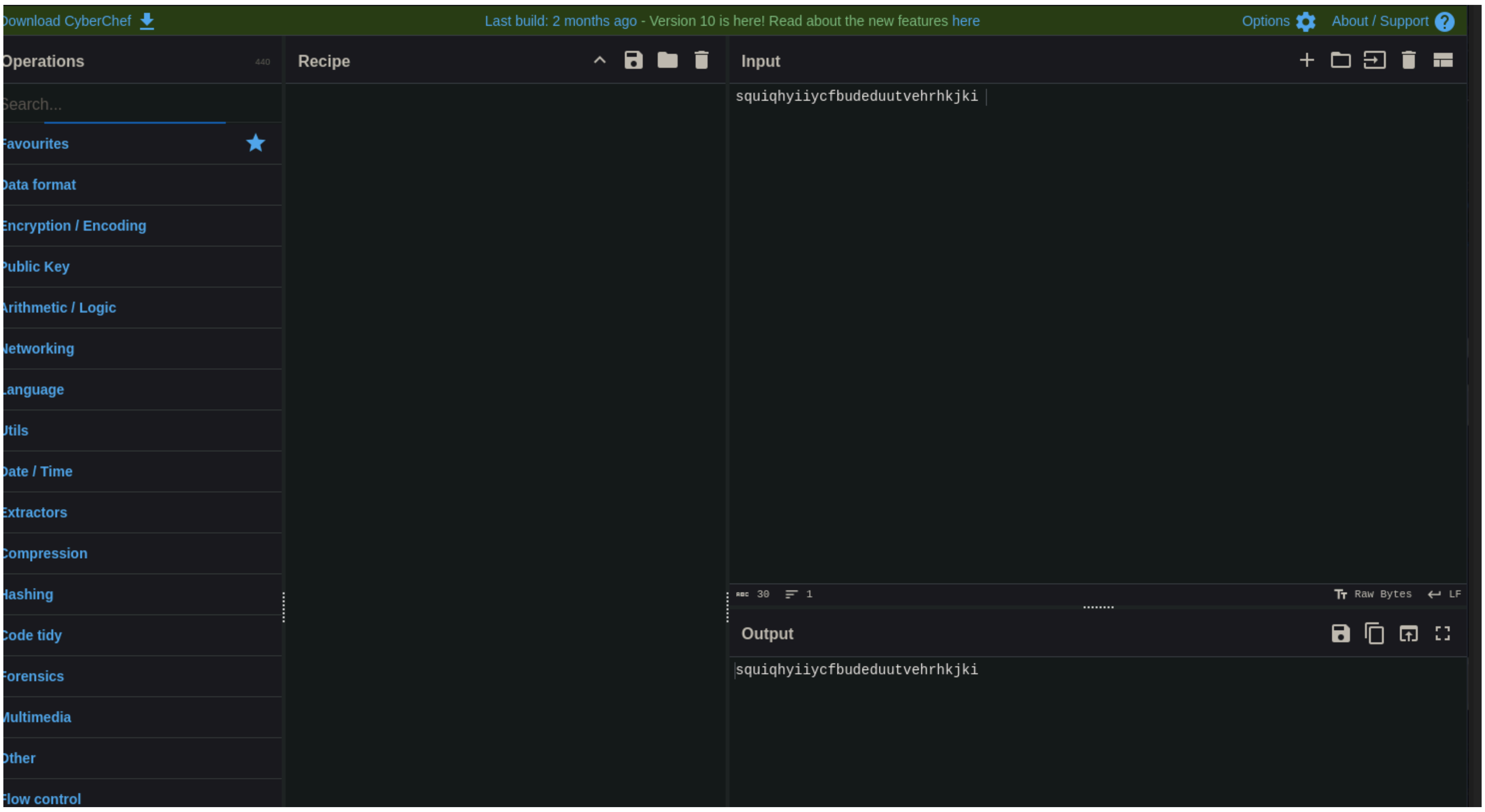
A simple message for you to decipher:

`squiqhyiiycfbudeduutvehrhkjki`

Submit the original plaintext hashed with MD5, wrapped between the usual flag format: `flag{}`

Ex: If the deciphered text is "hello world", the MD5 hash would be `5eb63bbbe01eeed093cb22bb8f5acdc3`, and the flag would be `flag{5eb63bbbe01eeed093cb22bb8f5acdc3}`.

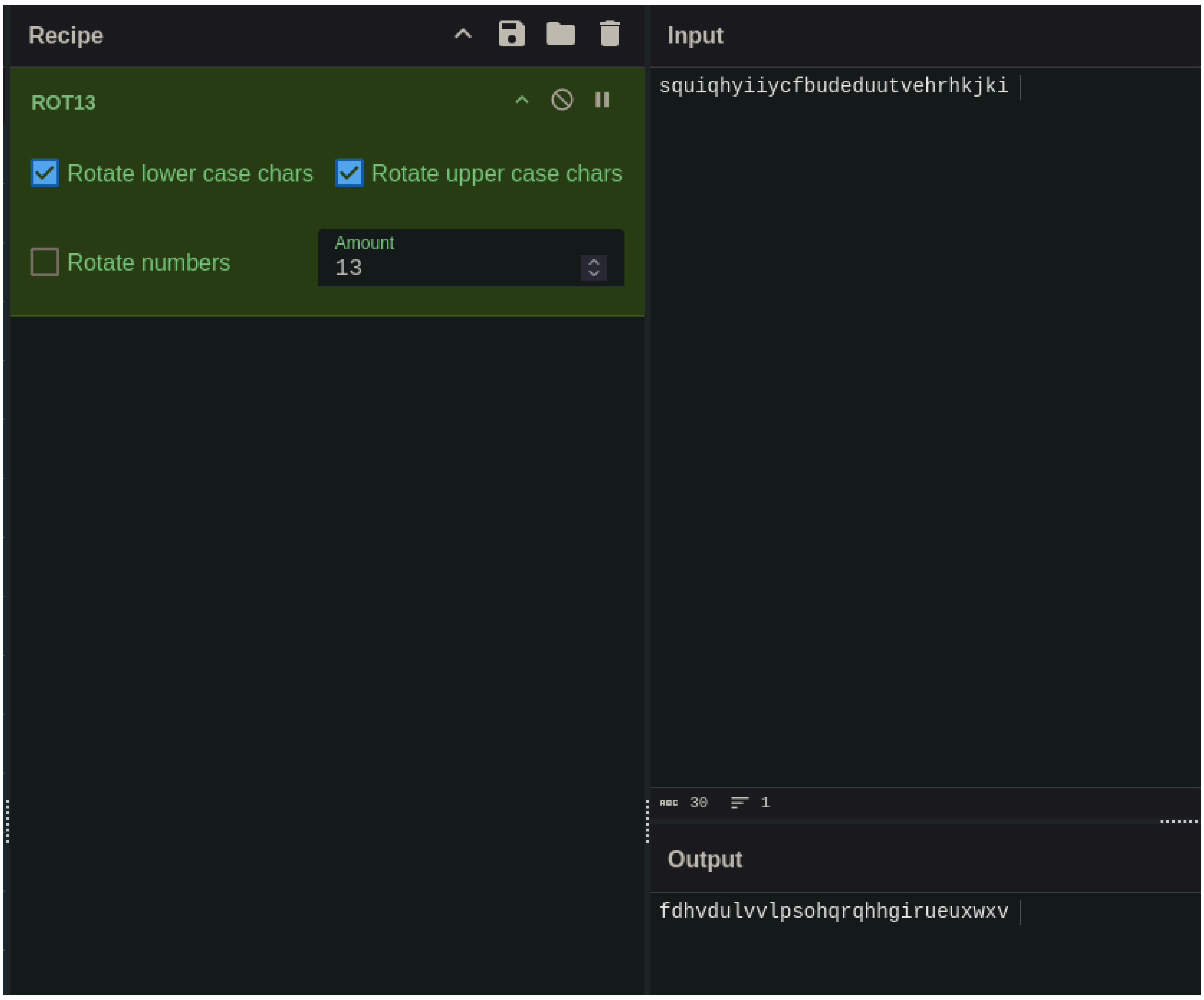
We are given a string to decode for the message. Then we need to use MD5 to make it a hash. This will give us the contents of the flag. Lets take this over to cyberchef.



```
squiqhyiiycfbudeduutvehrhkjki
```

```
squiqhyiiycfbudeduutvehrhkjki
```

Upon looking at the string it seems to have letters moved around to misconstrue the real message. I try ROT-13 and ROT-47, but did not get correct results.



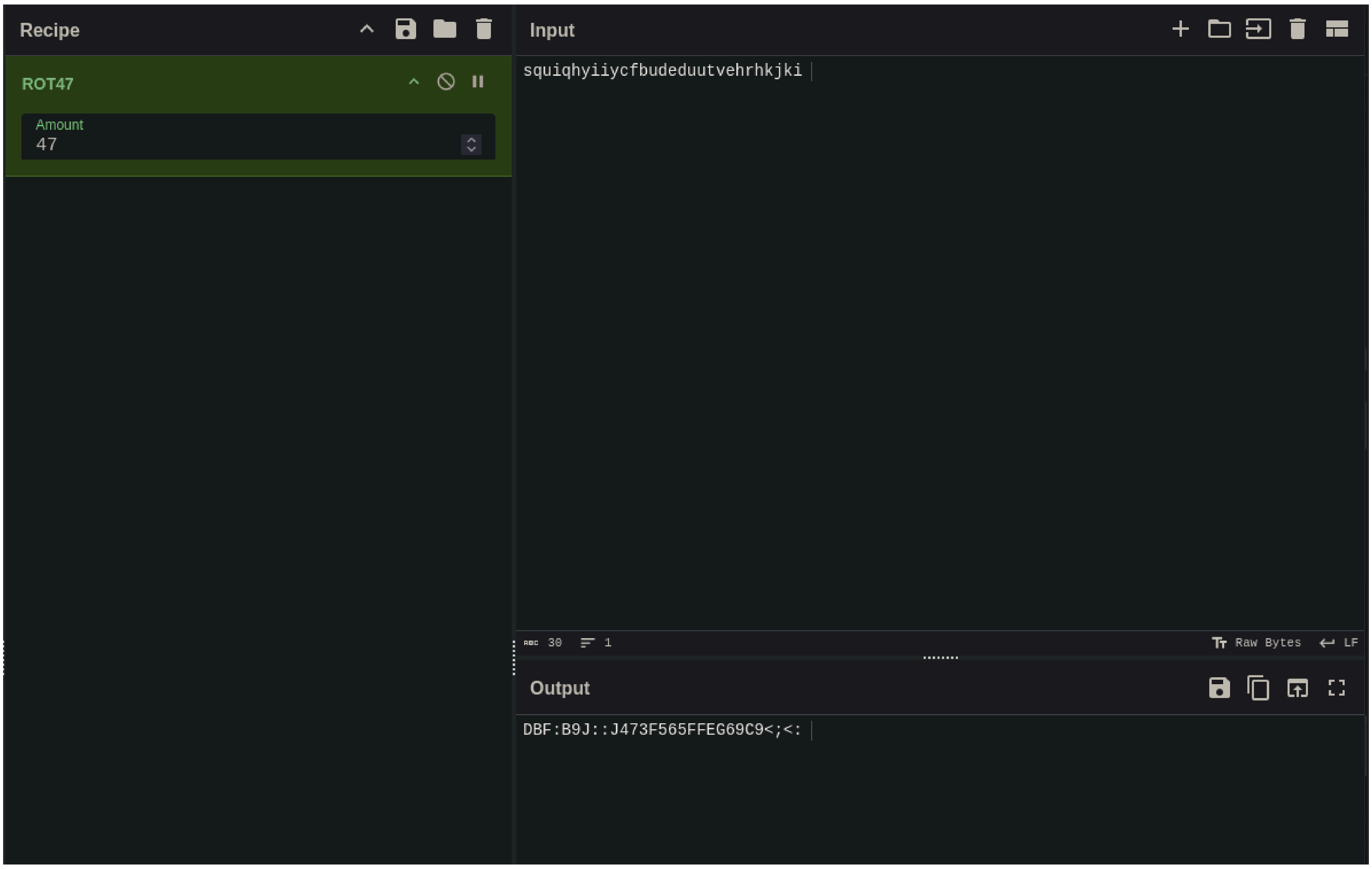
☑ Rotate lower case chars ☑ Rotate upper case chars

□ Rotate numbers

Amount
13

```
squiqhyiiycfbudeduutvehrhkjki
```

```
fdhvdulvvlpsohqrqhhgirueuxwxv
```

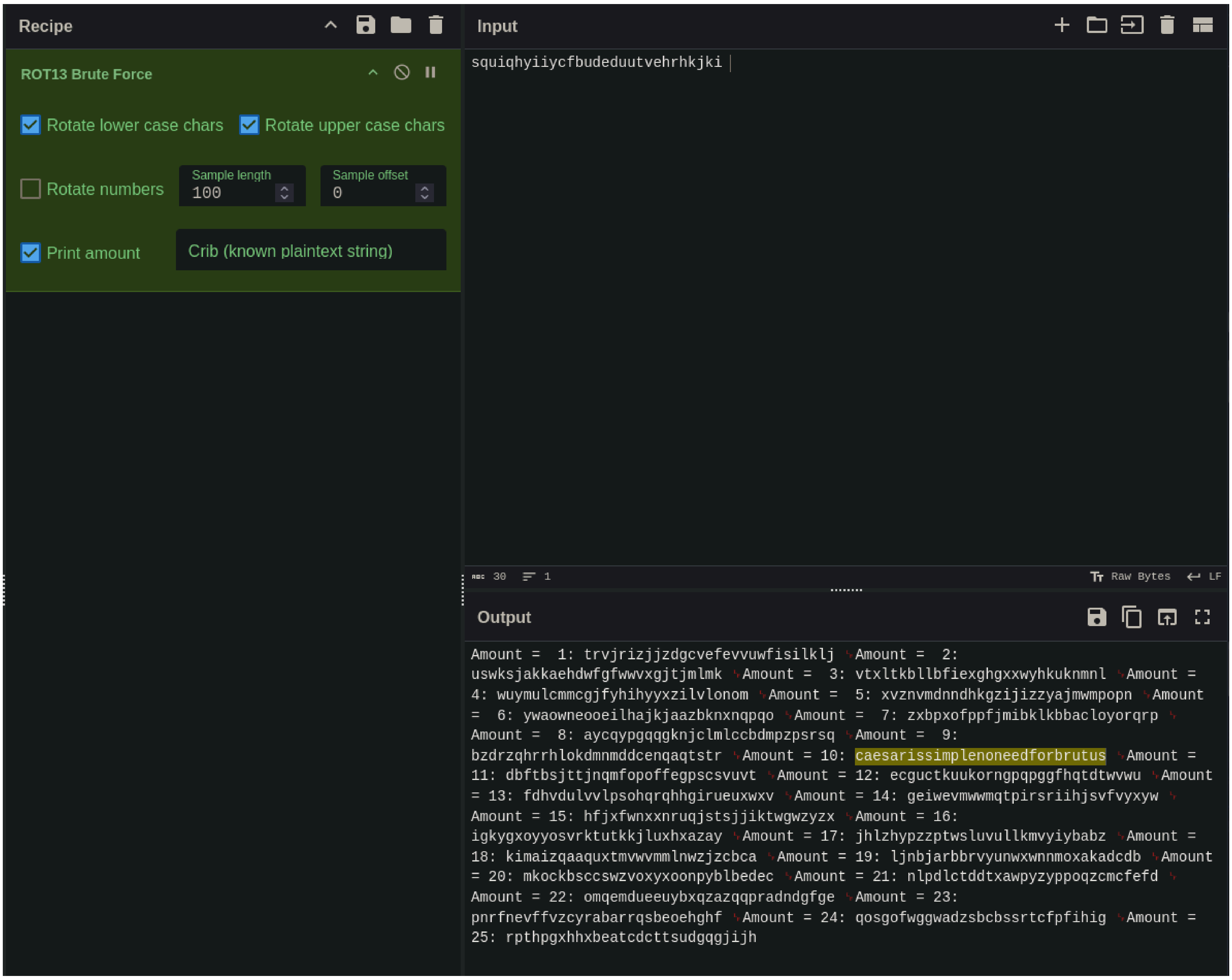


Amount
47

```
squiqhyiiycfbudeduutvehrhkjki
```

```
DBF:B9J::J473F565FFEG69C9<;<:
```

So I try "ROT13 Brute Force" recipe. I get a readable message that I can try as the flag. It needed an amount of 10 revelations of ROT13.



ROT13 Brute Force

☑ Rotate lower case chars ☑ Rotate upper case chars

☐ Rotate numbers

Sample length
100

Sample offset
0

☑ Print amount

Crib (known plaintext string)

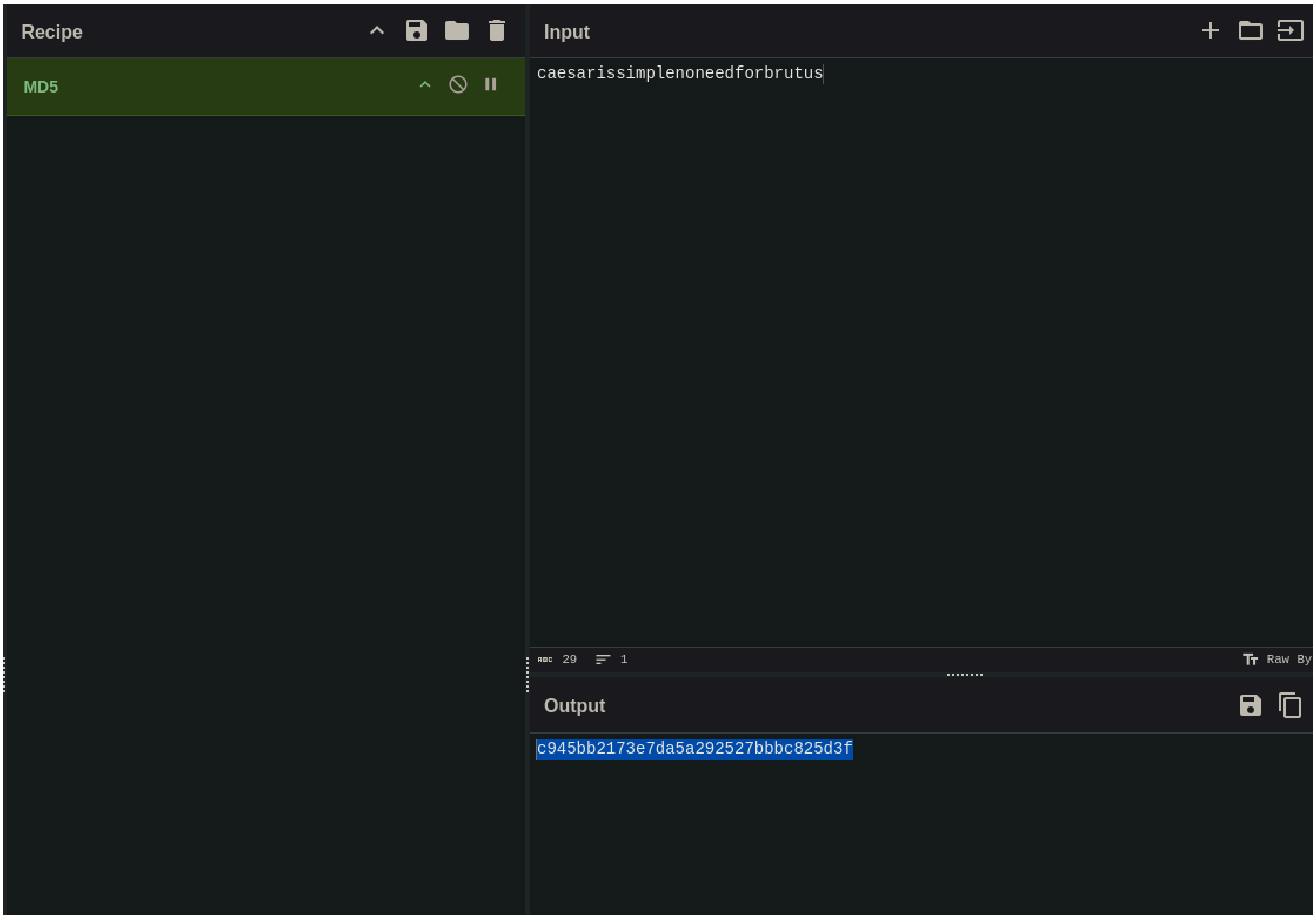
Input

```
squiqhyiiycfbudeduutvehrhkjki
```

Output

```
Amount =  1: trvjrizjjzdgcvefevvuwfisilklj ␊Amount =  2:
uswksjakkaehdwfgfwwvxgjtjmlmk ␊Amount =  3: vtxltkbllbfiexghgxxwyhkuknmnl ␊Amount =
4: wuymulcmmcgjfyhihyyxzilvlonom ␊Amount =  5: xvznvmdnndhkgzijizzyajmwmpopn ␊Amount
=  6: ywaowneooeilhajkjaazbknxnqpqo ␊Amount =  7: zxbpxofppfjmibklkbbacloyorqrp ␊
Amount =  8: aycqypgqqgknjclmlccbdmpzpsrsq ␊Amount =  9:
bzdrzqhrrhlokdmnmddcenqaqtstr ␊Amount = 10: caesarissimplenoneedforbrutus ␊Amount =
11: dbftbsjttjnqmfopoffegpscsvuvt ␊Amount = 12: ecguctkuukorngpqpggfhqtdtwvwu ␊Amount
= 13: fdhvdulvvlpsohqrqhhgirueuxwxv ␊Amount = 14: geiwevmwwmqtpirsriihjsvfvyxyw ␊
Amount = 15: hfjxfwnxxnruqjstsjjiktwgwzyzx ␊Amount = 16:
igkygxoyyosvrktutkkjluxhxazay ␊Amount = 17: jhlzhypzzptwsluvullkmvyiybabz ␊Amount =
18: kimaizqaaquxtmvwvmmlnwzjzcbca ␊Amount = 19: ljnbjarbbrvyunwxwnnmoxakadcdb ␊Amount
= 20: mkockbsccswzvoxyxoonpyblbedec ␊Amount = 21: nlpdlctddtxawpyzyppoqzcmcfefd ␊
Amount = 22: omqemdueeuybxqzazqqpradndgfge ␊Amount = 23:
pnrfnevffvzcyrabarrqsbeoehghf ␊Amount = 24: qosgofwggwadzsbcbssrtcfpfihig ␊Amount =
25: rpthpgxhhxbeatcdcttsudgqgjijh
```

So now we will take the decoded string and use MD5 to make it a hash.



```
caesarissimplenoneedforbrutus
```

29 1

```
c945bb2173e7da5a292527bbbc825d3f
```

Now we can encapsulate it with the "Flag{}" format.

}