Capture The Flag 2

Time limit: 5.0s **Memory limit:** 256M

CTF stands for Capture The Flag. After all, what else could CTF stand for?

Many problems have long flavour texts that are 'interesting', 'fun to read', and 'help you understand the problem'. In this problem, we cut out the middleman and provide you with the checker directly.

```
from re import split as resplit
from dmoj.utils.unicode import utf8bytes

p=21423806447394057716114884351768139740157716294272001591783248895827984187811227455598782832
09081418288126625169065584467816404098193951958525734996057373855753255674304647690194552312193
07534330918124782541045006801468006763289073960841802463048790554330204444651346374736096764913
1340702479287807853187 1
q=30507412659150399636286826545062944878827389057946084522984397603991477269113163490818969245
83380838553082395474142528327191134282232074036048874990866164421519050604566682623037600297413
76520087657350441875023796855291451765685626141027315115570074695194069704744678759100368904071
7178650520870330806345 3
n = p * q
assert n >= 10 ** 600

e = 289
goal=46446758267700605789745903310042351975630318786832355444354583002481343641976658724327294
17542529924210796547054319161798532259788013332990821763155222365837938150021441194283882615487
86540956054143469339817419980018119648530068591952848674654330850848174733546612649188888690465
07299032469436209812650554879861032185256180284615041606288524842036317711557683912069331056623
75097201439605660253935931114950939431325590608588889047274915131842926029129925411583691163133
43019099347116814030162632472857170786964636799128565247956976472583568554089356284634683641137
60874474759564293017717718113593372194316

def is_valid(s):
    if len(s) != 13:
        return False

    for c in s:
        if not (32 <= c <= 127):
            return False

    return True

def check(process_output, judge_output, judge_input, **kwargs):
    process_lines = list(filter(None, resplit(b'[\r\n]', utf8bytes(process_output))))

    if len(process_lines) != 1:
        return False

    s = process_lines[0] #s is a byte string

    if not is_valid(s):
        return False

    #Encryption Step
    encoded = 0
```

```
    for c in s:
        encoded *= 256
        encoded += c

    assert encoded <= 256 ** 13 <= 10 ** 32

    encrypted = pow(encoded, e, n)

    return encrypted == goal
```