Principle 1.2. If all the numbers you enter into MATLAB to do some calculation, are "reasonably large" and the result of this calculation, is one or more numbers which are "close to" eps, it is very likely that the number or numbers should be zero.

As an example, enter:

```
>> deg = pi/180; th = 40; 1 - (cos(th*deg)^2 + sin(th*deg)^2)
```

The result is `1.1102e-16`. Clearly, all the numbers entered into this calculation are “reasonable” and the result is approximately `eps`. Obviously, the result is supposed to be zero since, from the Pythagorean theorem:

$$\cos^2(\theta) + \sin^2(\theta) = 1$$

for all angles θ.

MATLAB tries to calculate the correct result, but it cannot quite. It is up to you to interpret what MATLAB is trying to tell you.

Note: If you obtained zero for the above calculation, try:

```
>> deg = pi/180
>> th = input(‘angle = ‘); 1 - (cosd(th)^2 + sind(th)^2)
```

for various angles. Some of these calculations should be nonzero.
In fact, MATLAB has introduced two functions to reduce round-off errors.

Solution – RegEx:

```
import re

p = re.compile(r'sin\(th\)\^2\+cos\(th\)\^2')
print p.sub('1', '1-sin(th)^2+cos(th)^2')
```

output:

```
vegastrek@Latitude-D630:~/Desktop$ python math.py

1-1
```

Experiment Result:

```
>> my
angle = 1000
result = 0
>> my
angle = 4000
result =    1.11022302462516e-16
>> my
angle = 2000
result = 0
>> my
angle = 3000
result = 0
>> my
angle = 3500
result = 0
>> my
angle = 3700
result =    1.11022302462516e-16
>> my
angle = 3600
result = 0
>> my
angle = 3650
result = 0
>> my
angle = 3675
result = 0
>> my
angle = 3680
result = 0
>> my
angle = 3690
result = 0
>> my
angle = 3695
result =    1.11022302462516e-16
>> my
angle = 3691
result = 0
>> my
angle = 3692
result = 0
>> my
angle = 3693
result = 0
>> my
angle = 3694
result = 0
>> my
angle = 3695
result =    1.11022302462516e-16
```