IUPAC

Write a program called iupac.py that translates an IUPAC-encoded (https://www.bioinformatics.org/sms/iupac.html) string of DNA into a regular expression that will match all the possible strings of DNA that match.

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
U	Uracil
R	A/G
Y	C/T
S	G/C
W	A/T
K	G/T
M	A/C
B	C/G/T
D	A/G/T
H	A/C/T
V	A/C/G
N	any base
./-	gap

For instance, the pattern AYG would match both ACG and ATG, so the regular expression would be ^A[CT]G$. We can use the REPL to verify that this works:

```
>>> import re
>>> re.search('^A[CT]G$', 'ACG')
<re.Match object; span=(0, 3), match='ACG'>
>>> re.search('^A[CT]G$', 'ATG')
<re.Match object; span=(0, 3), match='ATG'>
>>> 'OK' if re.search('^A[CT]G$', 'ACG') else 'NO'
'OK'
```

Your program should echo the given pattern and a translation to a regular expression. Then iterate through a sorted list of all possible combinations of the bases to test your regular expression, printing “OK” if there is a match and “NO” if not.

```
$ ./iupac.py AYG
pattern = "AYG"
regex   = "^A[CT]G$"
ACG OK
```

```
ATG OK
$ ./iupac.py MRY
pattern = "MRY"
regex   = "^[AC][AG][CT]$"
AAC OK
AAT OK
AGC OK
AGT OK
CAC OK
CAT OK
CGC OK
CGT OK
```