# Re.split()

In [1]:

```
regex_pattern = r"[, .]"

import re
print("\n".join(re.split(regex_pattern, input())))
```

```
100,000.000
100
000
000
```

# Group(), Groups() & Groupdict() ¶

In [*]:

```
import re
exp=r"([a-zA-Z0-9])\1+"
o=re.search(exp, input())
if o:
    print(o.group(1))
else:
    print(-1)
```

# Re.start() & Re.end()

In [*]:

```
import re
s=input()
k=input()
pattern=re.compile(k)
m=pattern.search(s)
if not m:
    print('(-1, -1)')
else:
    while m:
        print("({0}, {1})".format(m.start(), m.end()-1))
        m=pattern.search(s,m.start() + 1)
```

# Validating phone number

In [*]:

```
n=int(input())
for i in range(n):
    a=str(input())
    if(len(a)==10):
        if(a.isnumeric()):
            if(a[0]=="?" or a[0]=="8" or a[0]=="9"):
                print("YES")
            else:
                print("NO")

         else:
                print("NO")
    else:
        print("NO")
```

# Validating Roman Numerals

In [*]:

```
th='M{0, 3}'
hu='(C[MD]|D?C{0,3})'
ten='(X[CL]|L?X{0,3})'
dig='(I[VX]|V?I{0,3})'
regex_pattern = r"%s%s%s%s$" % (th,hu,ten,dig)  # Do not delete 'r'.

import re
print(str(bool(re.match(regex_pattern, input()))))
```

# Validating and Parsing Email Adresses

In [*]:

```
import re
n=int(input())
for i in range(n):
    line=input()
    name,email=line.split(" ")
    pattern="<[a-z][a-zA-Z0-9\-\.\_]+@[a-zA-Z]+\.[a-zA-Z]{1,3}"
    if bool(re.match(pattern,email)):
        print(name,email)
```

# HTML Parser - Part 1

In [*]:

```
from html.parser import HTMLParser
class MyHTMLParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print("Start :", tag)
        for attr in attrs:
            print('->', ' > '.join(map(str, attr)))
    def handle_endtag(self, tag):
        print("End :",tag)
    def handle_startendtag(self, tag, attrs):
        print('Empty :', tag)
        for attr in attrs:
            print('->', ' > '.join(map(str, attr)))
html=""
for i in range(int(input())):
    html+=input()
parser=MyHTMLParser()
parser.feed(html)
parser.close()
```

# HTML Parser - Part 2

In [*]:

```
from html.parser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_comment(self, comment):
        if '\n' in comment:
            print('>>> Multi-line Comment')
        else:
            print('>>> Single-line Comment')
        print(comment)

    def handle_data(self, data):
        if data == '\n' : return
        print('>>> Data')
        print(data)

html = ""
for i in range(int(input())):
    html += input().rstrip()
    html += '\n'

parser = MyHTMLParser()
parser.feed(html)
parser.close()
```

# Thanks