Python Programming

EDUCATION FOR PEACE

(ESTD-1995)

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

REGULAR EXPRESSIONS - 4

Guido Van Rossum

Learning Mantra

If you really strong in the basics, then remaining things will become so easy.

Agenda

1. **Example applications using Regular expressions**

7. Example applications using Regular expressions

App 1: Write a Regular Expression to represent all ‘P++’ language (My own language) identifiers.

Rules:

1. The allowed characters are a-z,A-Z,0-9,#.
2. The first character should be a lower case alphabet symbol from a to k.
3. The second character should be a digit divisible by 3.
4. The length of identifier should be at least 2.

Regular Expression:

[a-k][3069][a-zA-Z0-9#]*

1. Write a python program to check whether the given string is ‘P++’ language identifier or not?

```
import re
s = input('Enter Identifier to validate :')
m = re.fullmatch('[a-k][3069][a-zA-Z0-9#]*',s)
if m!= None:
    print(s,'is valid P++ Identifier')
else:
    print(s,'is not P++ Identifier')
```

Output:

```
Enter Identifier to validate :a3
a3 is valid P++ Identifier
```

Eg:

```
import re
s = input('Enter Identifier to validate :')
m = re.fullmatch('[a-k][3069][a-zA-Z0-9#]*',s)
if m!= None:
    print(s,'is valid P++ Identifier')
else:
    print(s,'is not P++ Identifier')
```

Output:

```
Enter Identifier to validate :z3k5
z3k5 is not P++ Identifier
```

Eg:

```
import re
s = input('Enter Identifier to validate :')
m = re.fullmatch('[a-k][3069][a-zA-Z0-9#]*',s)
if m!= None:
    print(s,'is valid P++ Identifier')
else:
    print(s,'is not P++ Identifier')
```

Output:

```
Enter Identifier to validate :a9@
a9@ is not P++ Identifier
```

Eg:

```
import re
s = input('Enter Identifier to validate :')
m = re.fullmatch('[a-k][3069][a-zA-Z0-9#]*',s)
if m!= None:
    print(s,'is valid P++ Identifier')
else:
    print(s,'is not P++ Identifier')
```

Output:

```
Enter Identifier to validate :f3jkgjidu
f3jkgjidu is valid P++ Identifier
```

Eg:

```
import re
s = input('Enter Identifier to validate :')
m = re.fullmatch('[a-k][3069][a-zA-Z0-9#]*',s)
if m!= None:
    print(s,'is valid P++ Identifier')
else:
    print(s,'is not P++ Identifier')
```

Output:

```
Enter Identifier to validate :a6kk9z##
a6kk9z## is valid P++ Identifier
```

App 2: Write a Regular Expression to represent all 10 digit mobile numbers.

Rules:

1. Every number should contains exactly 10 digits.

2. The first digit should be 7 or 8 or 9

Regular Expression

[7-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]

or

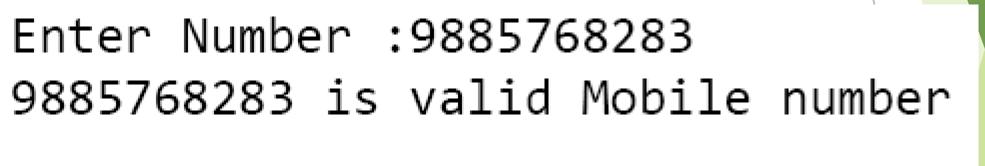
[7-9][0-9]{9}

or

[7-9]\d{9}

Write a Python Program to check whether the given number is valid mobile number or not?

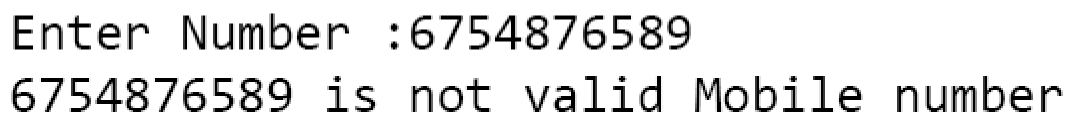
```
import re
s = input('Enter Number :')
m = re.fullmatch('[7-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :9885768283
9885768283 is valid Mobile number
```

Eg:

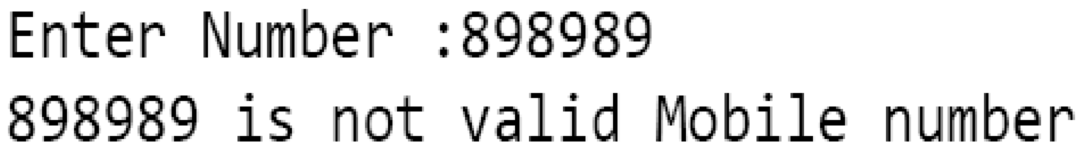
```
import re
s = input('Enter Number :')
m = re.fullmatch('[7-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :6754876589
6754876589 is not valid Mobile number
```

Eg:

```
import re
s = input('Enter Number :')
m = re.fullmatch('[7-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :898989
898989 is not valid Mobile number
```

App 3: Write a regular expression to find the valid mobile number based on the following rules.

The mobile number may contain 10 digit or 11 digit or 12 digit or 13 digit also:

10 : 6 to 9, 9 digits ===> [6-9][0-9]{9}

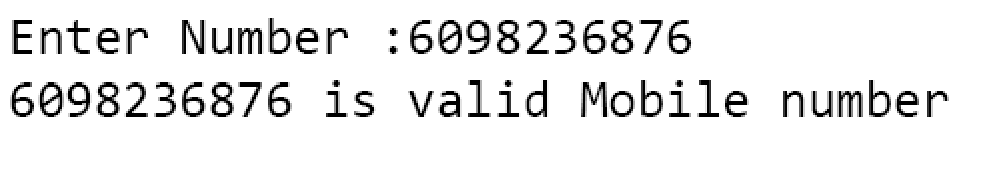
11: The first digit is 0 ===>0[[6-9][0-9]{9}

12: The first 2 digits should 91 ===>[9][1][6-9][0-9]{9}

13: The first 3 digits should be +91 ===>[+][9][1][6-9][0-9]{9}

Write a Python Program to check whether the given number is valid mobile number or not?

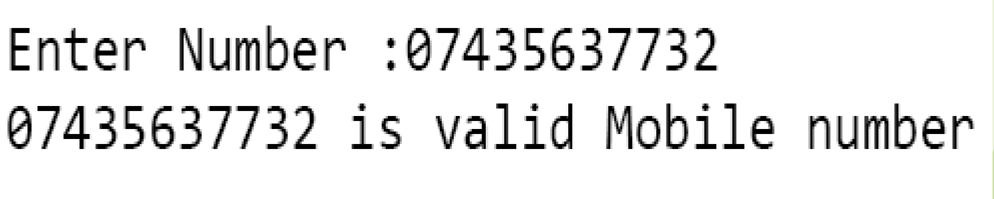
```
import re
s = input('Enter Number :')
m = re.fullmatch('[6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :6098236876
6098236876 is valid Mobile number
```

Eg:

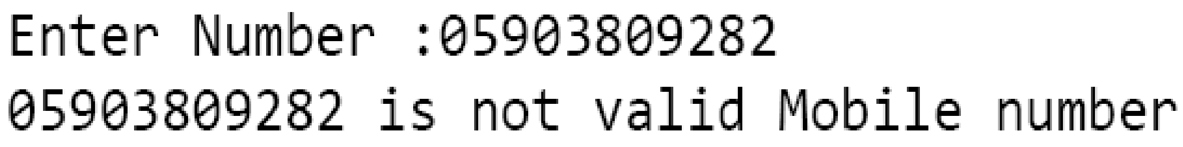
```
import re
s = input('Enter Number :')
m = re.fullmatch('[0][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :07435637732
07435637732 is valid Mobile number
```

Eg:

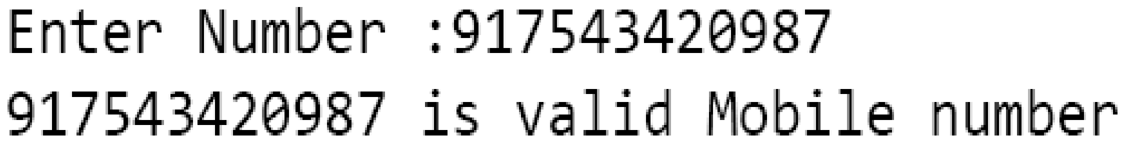
```
import re
s = input('Enter Number :')
m = re.fullmatch('[0][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :05903809282
05903809282 is not valid Mobile number
```

Eg:

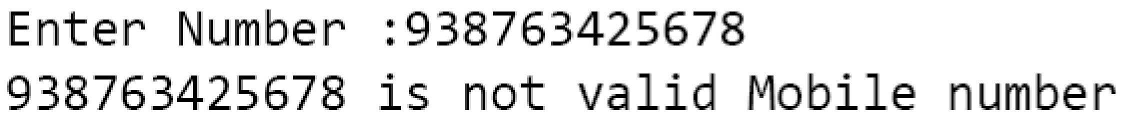
```
import re
s = input('Enter Number :')
m = re.fullmatch('[9][1][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :917543420987
917543420987 is valid Mobile number
```

Eg:

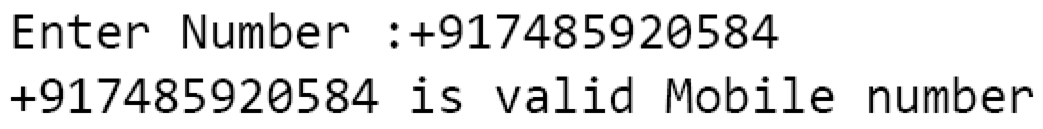
```
import re
s = input('Enter Number :')
m = re.fullmatch('[9][1][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :938763425678
938763425678 is not valid Mobile number
```

Eg:

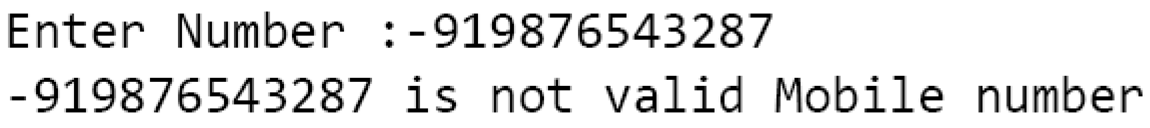
```
import re
s = input('Enter Number :')
m = re.fullmatch('[+][9][1][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :+917485920584
+917485920584 is valid Mobile number
```

Eg:

```
import re
s = input('Enter Number :')
m = re.fullmatch('[+][9][1][6-9][0-9]{9}',s)
if m!= None:
    print(s,'is valid Mobile number')
else:
    print(s,'is not valid Mobile number')
```



```
Enter Number :-919876543287
-919876543287 is not valid Mobile number
```

App 4: Write a python program to extract all mobile numbers present in input.txt file where numbers are mixed with normal text data.

Note: Executed in **Editplus** Editor

Assume that, *input.txt* file contains the following information.

```
C:\Pythonpractice>type input.txt
Hello this is ABC with mobile number 9885768283
hello this is DEF with mobile number 8787878787
hello this is GHI with mobile number 9101919178
999999999999 and 963374944994
```

Eg:

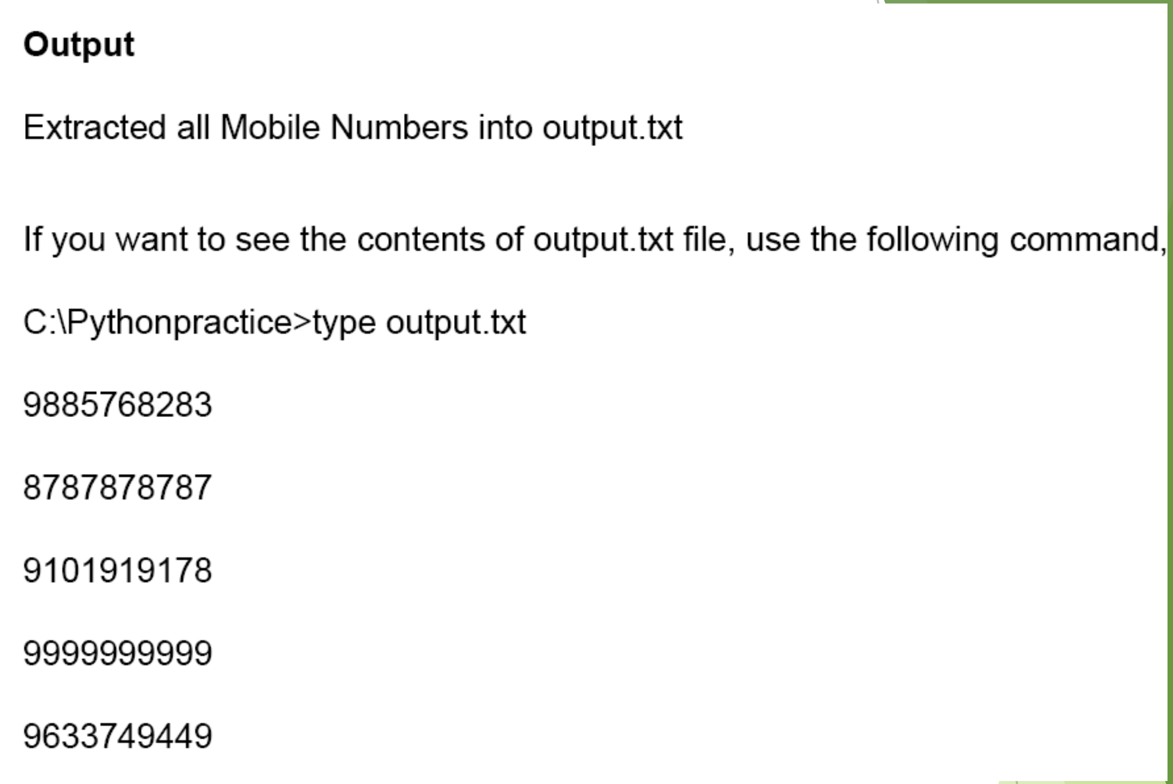
```
import re
f1=open("input.txt","r")
f2=open("output.txt","w")
for line in f1:
    print(line)
```

Output

```
Hello this is ABC with mobile number 9885768283
hello this is DEF with mobile number 8787878787
hello this is GHI with mobile number 9101919178
```

Eg:

```
import re
f1=open("input.txt","r")
f2=open("output.txt","w")
for line in f1:
    list=re.findall("[7-9]\d{9}",line)
    for n in list:
        f2.write(n+"\n")
print("Extracted all Mobile Numbers into output.txt")
f1.close()
f2.close()
```



Output

```
Extracted all Mobile Numbers into output.txt
```

If you want to see the contents of output.txt file, use the following command,

```
C:\Pythonpractice>type output.txt
9885768283
8787878787
9101919178
9999999999
9633749449
```

Any question?

If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You