Harsh Kasliwal
2003085

EXPERIMENT 8

Q1.

1.	Python program to demonstrate use of regular expression ● Create a **phone list** using file **(surname name number)** ● Find all the entries of phone book with surname as **"Rao"** and first name starting with **'J' or 'K'.**

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.
Regular expressions are widely used in UNIX world.The Python module re provides full support for Perl-like regular expressions in Python.
The re module raises the exception re.error if an error occurs while compiling or using a regular expression.
We would cover two important functions, which would be used to handle regular expressions. But a small thing first:
There are various characters, which would have special meaning when they are used in regular expression. To avoid any confusion while dealing with regular expressions, we would use Raw Strings as r'expression'.
Syntax : re.search(pattern, string, flags=0)
Pattern : This is the regular expression to be matched.
String : This is the string, which would be searched to match the pattern at the beginning of string.
Flags : You can specify different flags using bitwise OR (|).

Sets

A set is a set of characters inside a pair of square brackets `[]` with a special meaning:

Set	Description
[arn]	Returns a match where one of the specified characters (`a` , `r` , or `n`) are present
[a-n]	Returns a match for any lower case character, alphabetically between `a` and `n`
[^arn]	Returns a match for any character EXCEPT `a` , `r` , and `n`
[0123]	Returns a match where any of the specified digits (`0` , `1` , `2` , or `3`) are present
[0-9]	Returns a match for any digit between `0` and `9`
[0-5][0-9]	Returns a match for any two-digit numbers from `00` and `59`
[a-zA-Z]	Returns a match for any character alphabetically between `a` and `z` , lower case OR upper case
[+]	In sets, `+` , `*` , `.` , `|` , `()` , `$` , `{}` has no special meaning, so `[+]` means: return a match for any `+` character in the string

Python program to demonstrate use of regex with a **phone list** using file **(surname name number)**

CODE

```
import re
lst = []

with open('phonebook.txt', 'r') as my_file:
    for entry in my_file:
        lst.append(entry.split())

print("The entries in the phonebook are : ")
for contact in lst:
    print(contact)
print("The entries with rao as surname are : ")
for contact in lst:
    check = re.search('rao', contact[0])
    if check:
        print(contact)

print("The entries with first name starting with J or K are :")
for contact in lst:
    check = re.search('(^j)|(^k)', contact[1].lower())
    if check:
        print(contact)
```

OUTPUT

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Harsh\SEM 4\PYTHON\Assingment\EXP 8> python -u "d:\Harsh\SEM 4\PYTHON\Assingment\EXP 8\P1.py"
The entries in the phonebook are :
['kasliwal', 'harsh', '8698515006']
['rao', 'ramesh', '92424141']
['rao', 'mahesh', '23415125']
['lasi', 'darshan', '78924120']
['jagat', 'bhagat', '471894']
['kashish', 'kala', '3091820']
The entries with rao as surname are :
['rao', 'ramesh', '92424141']
['rao', 'mahesh', '23415125']
The entries with first name starting with J or K are :
['kashish', 'kala', '3091820']
PS D:\Harsh\SEM 4\PYTHON\Assingment\EXP 8>
```

Q2.

2.	Python program to demonstrate use of regular expression ● Create string with name of cities in india separated by spaces. ● Find all cities ending with “ai” ● Find all cities starting with “Mu” or “Ma” ● **print name of cities with ‘u’ as second letter and ‘a’ as second last letter**

CODE:

```
import re

s = '''Mumbai Surat Pune Delhi Chennai Indore Hyderabad Jaipur Kanpur
Banglore Ranchi Amritsar Kolkata Manglore Mathura'''
cities = s.split()
print("Cities ending in ai are : ")
for city in cities:
    check = re.search('ai$', city)
    if check:
        print(city)

print("Cities starting with 'Mu' or 'Ma' are : ")
for city in cities:
    check = re.search('(^mu)|(^ma)', city.lower())
    if check:
        print(city)

print("Cities with 'u' as second letter and 'a' as second last letter are : ")
for city in cities:
    check = re.search('(^.u).*(a.$)', city.lower())
    if check:
        print(city)
```

OUTPUT

```
PS D:\Harsh\SEM 4\PYTHON\Assingment\EXP 8> python -u "d:\Harsh\SEM 4\PYTHON\Assingment\EXP 8\P2.py"
Cities ending in ai are :
Mumbai
Dubai
Chennai
Dalai
Madukarai
Cities starting with 'Mu' or 'Ma' are :
Mumbai
Manglore
Madukarai
Cities with 'u' as second letter and 'a' as second last letter are :
Mumbai
Dubai
Aurangabad
Surat
PS D:\Harsh\SEM 4\PYTHON\Assingment\EXP 8>
```