1. **Write program demonstrates how to use regular expressions in Python to match and search for patterns in text.**

**Program:**

import re

def main():

text = input("Enter some text: ")

pattern = r'\b[Aa]\w+'

matches = re.findall(pattern, text)

print("Words starting with 'A' or 'a':")

for match in matches:

print(match)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. **Implement a basic finite state automaton that recognizes a specific language or pattern. In this example, we'll create a simple automaton to match strings ending with 'ab' using python.**

**Program:**

def match(string):

state = 0

for char in string:

if state == 0 and char == 'a':

state = 1

elif state == 1 and char == 'b':

state = 2

return state == 2

print(match("hello world"))

print(match("ab"))

print(match("aab"))

print(match("abab"))

1. **Write program demonstrates how to perform morphological analysis using the NLTK library in Python.**

**Program:**

import nltk

from nltk.corpus import wordnet

def morphological\_analysis(word):

synsets = wordnet.synsets(word)

if synsets:

for synset in synsets:

print("Word:", synset.name())

print("POS:", synset.pos())

print("Definition:", synset.definition())

print("Examples:", synset.examples())

print()

else:

print("No morphological analysis found for the word:", word)

def main():

word = input("Enter a word to perform morphological analysis: ")

morphological\_analysis(word)

if \_\_name\_\_ == "\_\_main\_\_":

nltk.download('wordnet')

main()

1. **Implement a finite-state machine for morphological parsing. In this example, we'll create a simple machine to generate plural forms of English nouns using python.**

**Program:**

import re

def pluralize(word):

rules = [

['[sxz]$', '$', 'es'],

['[^aeioudgkprt]h$', '$', 'es'],

['(qu|[^aeiou])y$', 'y$', 'ies'],

['$', '$', 's']

]

for rule in rules:

pattern, search, replace = rule

if re.search(pattern, word):

return re.sub(search, replace, word)

print(pluralize("cat"))

print(pluralize("dog"))

print(pluralize("knife"))

print(pluralize("potato"))

1. **Use the Porter Stemmer algorithm to perform word stemming on a list of words using python libraries.**

**Program:**

from nltk.stem import PorterStemmer

def stem\_words(words):

stemmer = PorterStemmer()

stemmed\_words = [stemmer.stem(word) for word in words]

return stemmed\_words

def main():

words = input("Enter a list of words separated by spaces: ").split()

stemmed\_words = stem\_words(words)

print("Original words:", words)

print("Stemmed words:", stemmed\_words)

if \_\_name\_\_ == "\_\_main\_\_":

main()