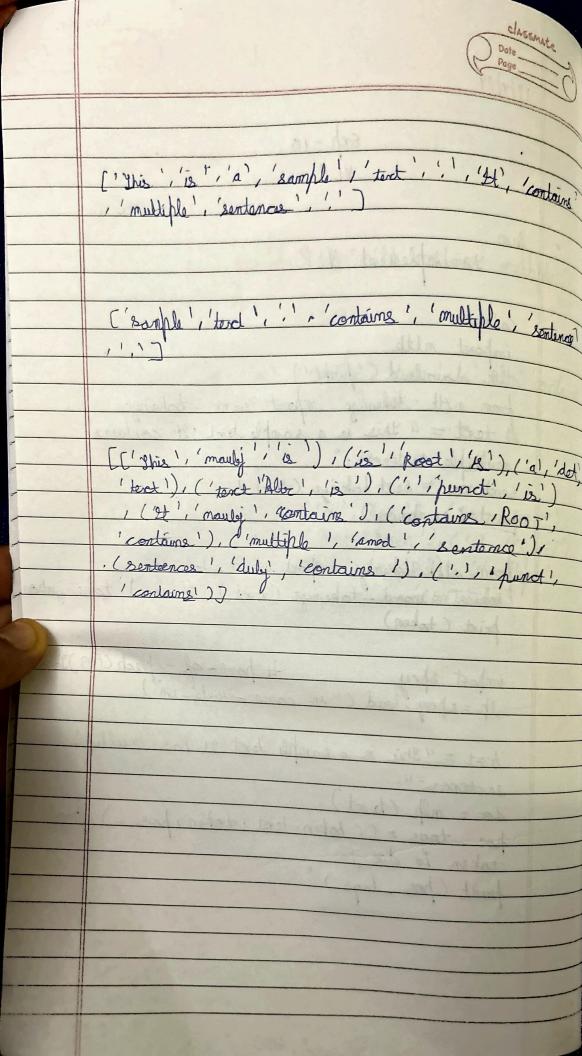
	Page O
	The state of the s
	Output:
	['This is a sample best', 'It contains multiple
	['This 1, 'a', 1 santa, 'tord', '1) '9+1' contains
	[('This ', 'PRON'), ('is', 'Aux'). ('al, 'DET'). (Sample, 'NOUN'), (Haret ', Moun'), ('). 'PUNET') ('9t', 'PRON), ('Contains', 'VERB'), ('multiple', 'ADT', ('sontences!, 'Noun), ('!, 'PUNTT')]
to of the	and the same that the same tha

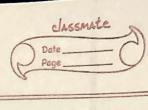
Classmate

Date ______
Page _____ 17/4/24 Exp-10 To implement NLP impost meth noth download ("pinkt") nltk download (pinkt)

from nttk tokenize impost sont - tokenize

A tort = " This is a sample text It contains multiple sentences," sentances = sent hoteringe (tent) point (sontinces) from mltk tokening import word - tokeninge tokens = word - tokeninge (tout) # word tokeningation import spry Hard ("en-coro - wele- Dri) terret = "This is a sample teret gt has multiple doc = mlp (text) pos - torge = [(token · text · token · pos -) for print (hos bags)





from nell. stom import wordnot le monatizos
melk. download ('usodnot') lemmatizer = Hord Atlemmatizer ()

lemmar = [lemmatizer & lemmatize (Hober) for boton in

tokene] point (lemmas) note download ('stopwoods') Stop woods = set (stopusords "woods ('english')) feltered tokens = Ctopen for token in topen of Hoken - tower () not in stop - words] point (filtered bokens) depending too = [(token for topen in laken inhead. prant (dependency true) # dependency fraging NLP was successfully implemented