National University of Computer and Emerging Sciences

Lab Exercise 02
CL2002-Artificial Intelligence Lab

Semester	Spring 2024

Department of Computer Science FAST-NU,
Lahore, Pakistan

Exercise (35 Marks)

- For this exercise, use the notebook *AI_Lab_2_Exercise.ipynb* provided on Google classroom.
- Open the *AI_Lab_2_Exercise.ipynb* on Google Collab by uploading it and remain it to Lab2_Rollno.
- The *main()* function is already defined and complete. DO NOT change it.
- Only enter your code within the space provided in the notebook and upload it on GC after Completion

You MUST COMPLETE 16 Exercises in the Class and submit them till 11:20pm. The remaining exercises 17-22 are due till Tuesday.

1 Number of chickens (5 Marks)

Given an int count of a number of chickens, return a string of the form 'Number of chickens: <count>', where <count> is the number passed in. However, if the count is 10 or more, then use the word 'many' instead of the actual count. So chickens(5) returns 'Number of chickens: 5' and chickens(23) returns 'Number of chickens: many'.

2 Strings from both ends (5 Marks)

Create a string made of the first three and the last three characters of the original string s and return the new string, so 'intelligence' creates 'intcne'. However, if the string length is less than 3, return instead the empty string.

3 Replace occurrences of first character (5 Marks)

For a given string s, create a new string such that it replaces some characters of the string with '@'. To replace, find the first character of the given string s. Now find all occurrences of this first character in the string and replace them with '@'. Do not change the first character itself.

e.g. 'Ooogle' create 'O@@gle'

Assume that the string is length 1 or more and take care of the word case.

Hint: s.replace(stra, strb) returns a version of string s where all instances of stra have been replaced by strb.

4 String jumble (5 Marks)

Create a new string with the help of two input strings a and b. The new string should be separated by a space b/w a and b. Also swap the first 2 chars of each string and return it. e.g.

'mix', pod' -> 'pox mid'

'dog', 'dinner' -> 'dig donner'

Assume a and b are length 2 or more.

5 Matching first and last characters (5 Marks)

Taking a list of strings as input, our matching function returns the count of the number of strings whose first and last chars of the string are the same.

Also, only consider strings with length of 2 or more.

6 Group strings in a list (5 Marks)

Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first.

e.g. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] creates ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']

Hint: use a custom key= function

7 Sort tuple by last element (5 Marks)

Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple.

e.g. [(1, 7), (1, 3), (3, 4, 5), (2, 2)] creates [(2, 2), (1, 3), (3, 4, 5), (1, 7)]

Hint: use a custom key= function to extract the last element form each tuple.

8 Palindrome Detection (5 Marks)

Given a string s, create a function to detect if it's a palindrome. A palindrome is a word, phrase, number, or other sequence of characters that reads the same backward or forward.

9 List Flattening (5 Marks)

Given a nested list of integers, create a function to flatten the list to a single-level list. For example, [[1, 2, 3], [4, 5]] will be flattened to [1, 2, 3, 4, 5].

10 List Intersection (5 Marks)

Given two lists of integers, create a function to return a list that contains only the elements that are common between the two lists (without duplicates). Make sure your program works with lists of different sizes.

11 String Anagrams (5 Marks)

Given two strings, create a function to detect if they are anagrams. An anagram is a word, phrase, or name formed by rearranging the letters of another, such as cinema, formed from iceman.

12 Binary Search (5 Marks)

Given a sorted list of integers, create a function to perform binary search on the list and return the index of the target element. If the target element is not found, return -1.

13 Merge Sort (5 Marks)

Given a list of integers, create a function to sort the list using the merge sort algorithm. The merge sort algorithm is a recursive sorting algorithm that works by breaking down the list into smaller sub-lists and then recombining them in a sorted manner.

14 Quick Sort (5 Marks)

Given a list of integers, create a function to sort the list using the quick sort algorithm. The quick sort algorithm is a recursive sorting algorithm that selects a pivot element from the list and then partitions the other elements into two sub-lists, according to whether they are less than or greater than the pivot.

15 Frequency Counter (5 Marks)

Given a list of integers, create a function to return a dictionary that contains the frequency count of each element in the list. For example, the list [1, 2, 2, 3, 4, 2] would return {1:1, 2:3, 3:1, 4:1}.

16 Count Distinct Elements (5 Marks)

Given a list of integers, create a function to return the count of distinct elements in the list. For example, the list [1, 2, 2, 3, 4, 2] would return 4.

You MUST COMPLETE 16 Exercises in the Class and Submit them till 11:20am. The remaining exercises 6 are due till Tuesday as HOME TASK.

Always read the submission instructions carefully.

- Rename your Jupyter notebook to your **Lab2_roll number** and download the notebook as **.ipynb** extension.
- To download the required file, go to **File->Download .ipynb** · Only submit the **.ipynb** file. DO NOT **zip** or **rar** your submission file
- Submit this file on Google Classroom under the relevant assignment.
- Late submissions will not be accepted
