**NIKITHA YELDI**

**700758003**

* The for loop iterates from 1 to 4 exclusive while the second for loop with iterates from 5 down to 1 inclusive. The step value of -1 ensures it decrements by 1 in each iteration.
* Inside the loop i\*’\*’ multiplies the current loop variables I by the asterisk character.
* The print function then prints the resulting string which creates a pattern of increasing and decreasing number of asterisks on each line.
* Here the program represents how to use loops and strings repetition to create the star pattern.

A screenshot of a computer program

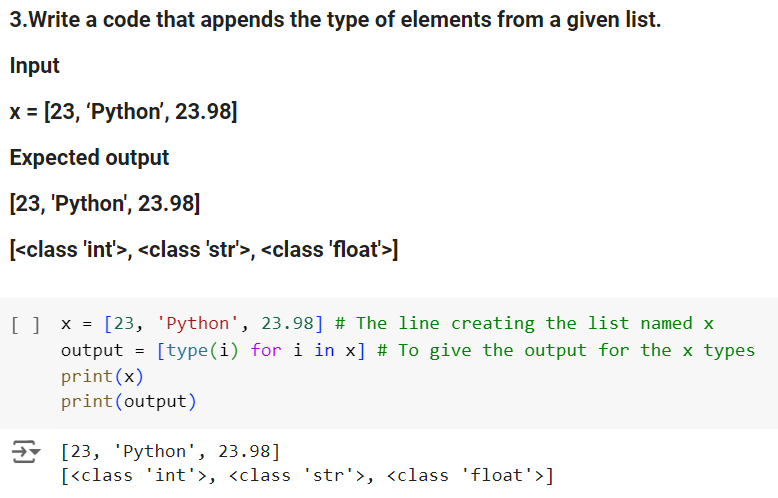
Description automatically generated

GITHUB LINK:

**https://github.com/nyeldi/machinelearning**

**Summer 2024: CS 5710 –Machine Learning**

**Programming Assignment-2**



* The for loop iterates a specific number of times based on the length of the list but it only considers odd indexes. Where range part controls how many times the loop runs and which element it accesses.
* The 1 sets the starting index for the loop 1 by skipping the first element at index 0.
* The length loop tells to iterate up but not including the length of the list.
* 2 is the step value which tells the loop to jump by 2 each iteration, skipping even indexes and focusing on odd ones.
* Therefore, the code creates the loop that skips even indexes and prints the value of elements at odd positions in the list.

A screenshot of a computer

Description automatically generated

* The above code takes a list and returns a new list with unique items. It acts like a sorting machine. It takes that the list and checks each element one by one. If it has not seen the unique element yet, it puts in a separate, empty new list.
* But if it sees the duplicate, it leaves out. At the end, the new list will only hold the unique elements from the original list.

A screenshot of a computer program

Description automatically generated

* The above question creates a list called x, uses a list comprehension to find the data types of its elements, and them it prints both the original list and the list of data types.
* Inside the loop, it uses type (i) function to determine the data type of the current element i. The type function returns a built in type depending on the element data.
* The output will store the results of the type(i) and prints the result.

* This code tackles counting uppercase and lowercase letters in a string. It defines a function named count that takes a string as input.
* The function employs list comprehensions for both uppercase and lowercase counting.
* These comprehensions go through each character in the string. If a character is uppercase (using c.isupper()), the comprehension yields 1, otherwise it yields 0.
* The sum function then adds up these 1s and 0s to determine the total count of uppercase and lowercase characters, which are then printed with informative messages.

A screenshot of a computer program

Description automatically generated