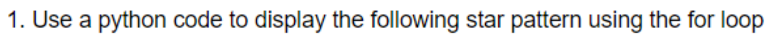
Machine Learning Assignment-2

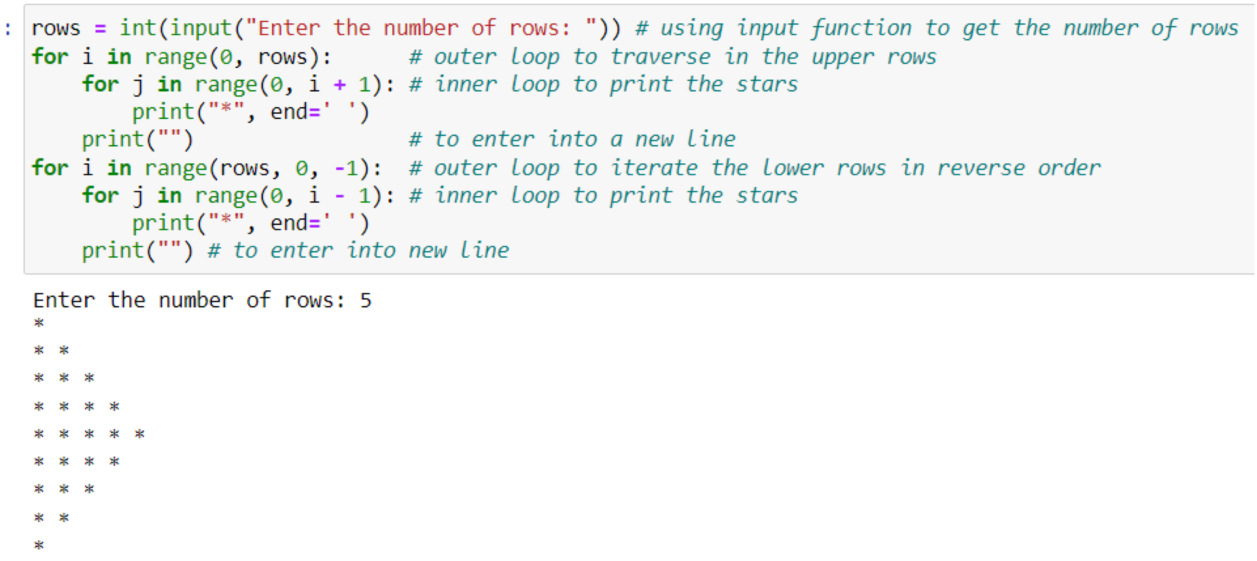
Sushmitha Virri: 700742289

Github code link: https://github.com/Sushmitha-Virri/MLAssignments21627/blob/main/ML_Assignment_2.ipynb

Github video link: https://github.com/Sushmitha-Virri/MLAssignments21627/blob/main/Sushmitha_Virri_ML_Assignment_2_video.mp4



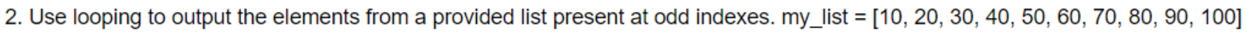
1. Use a python code to display the following star pattern using the for loop



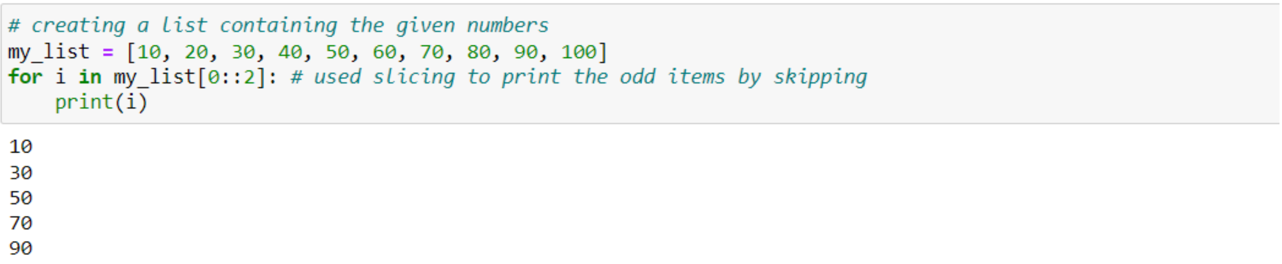
```
:  rows = int(input("Enter the number of rows: ")) # using input function to get the number of rows
   for i in range(0, rows):      # outer loop to traverse in the upper rows
       for j in range(0, i + 1): # inner loop to print the stars
           print("*", end=' ')
       print("")                 # to enter into a new line
   for i in range(rows, 0, -1):  # outer loop to iterate the lower rows in reverse order
       for j in range(0, i - 1): # inner loop to print the stars
           print("*", end=' ')
       print("") # to enter into new line
```

```
Enter the number of rows: 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Firstly I have used input() function to enter the number of rows of stars then with the help of outer for loop I have traversed through each row and using the inner for loop printed the stars. Similarly for the second part I used the outer for loop to iterate through the rows in reverse order and printed the stars.



2. Use looping to output the elements from a provided list present at odd indexes. my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

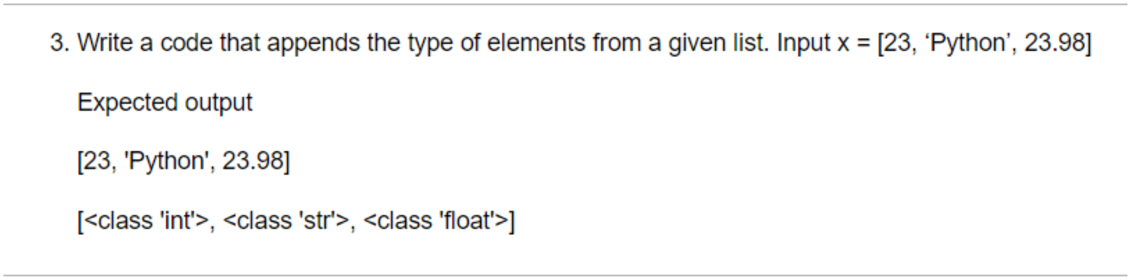


```
# creating a list containing the given numbers
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
for i in my_list[0::2]: # used slicing to print the odd items by skipping
    print(i)
```

```
10
30
50
70
90
```

For the second question I have created a list ‘my_list’ with the numbers 10,20,30,40,50,60,70,80,90,100.

Next I have used for loop to traverse through odd indexed elements using slicing started from the first index which is ‘0’ and skipping one element and go to the next element and print the elements at odd indices.

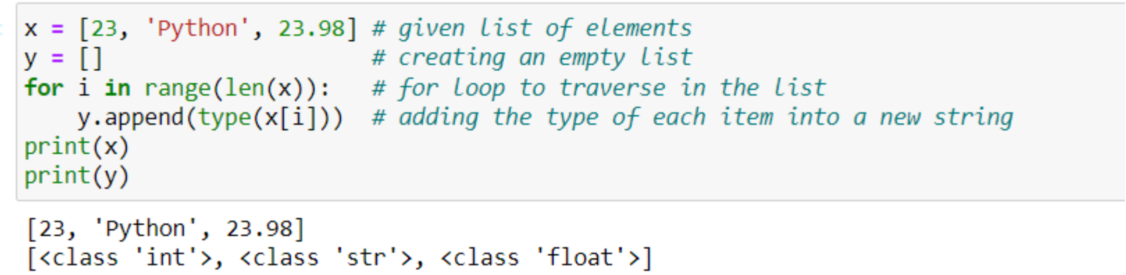


3. Write a code that appends the type of elements from a given list. Input x = [23, ‘Python’, 23.98]

   Expected output

   [23, 'Python', 23.98]

   [<class 'int'>, <class 'str'>, <class 'float'>]

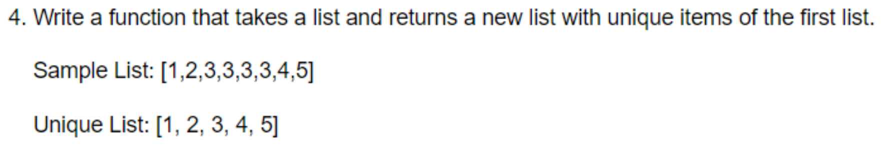


```
x = [23, 'Python', 23.98] # given list of elements
y = []                    # creating an empty list
for i in range(len(x)):   # for loop to traverse in the list
    y.append(type(x[i]))  # adding the type of each item into a new string
print(x)
print(y)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

Here I have created a list with a name ‘x’ and another empty list with the name ‘y’.

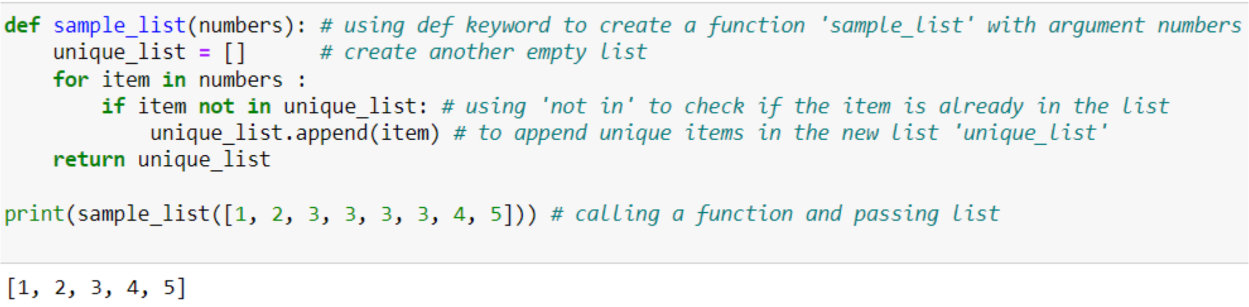
Next I have used a for loop to iterate through each element in x through the range of length in x and appended the type of each element in x using the type() function to the list y. Then printed both the lists.



4. Write a function that takes a list and returns a new list with unique items of the first list.

   Sample List: [1,2,3,3,3,3,4,5]

   Unique List: [1, 2, 3, 4, 5]



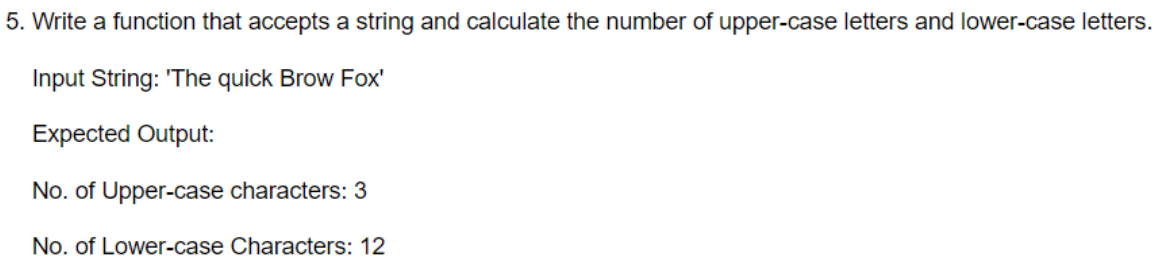
```
def sample_list(numbers): # using def keyword to create a function 'sample_list' with argument numbers
    unique_list = []      # create another empty list
    for item in numbers :
        if item not in unique_list: # using 'not in' to check if the item is already in the list
            unique_list.append(item) # to append unique items in the new list 'unique_list'
    return unique_list

print(sample_list([1, 2, 3, 3, 3, 3, 4, 5])) # calling a function and passing list
```

```
[1, 2, 3, 4, 5]
```

For this question for I have created a function with argument using ‘def’ keyword. I have called the function and passed the given list to the argument numbers list.

In the sample_list function I have created an empty list with the name unique_list. For each item in the numbers list I have used not in to check for the item is already in the list. If not then append the item to the new list unique_list then returned the unique_list to print.



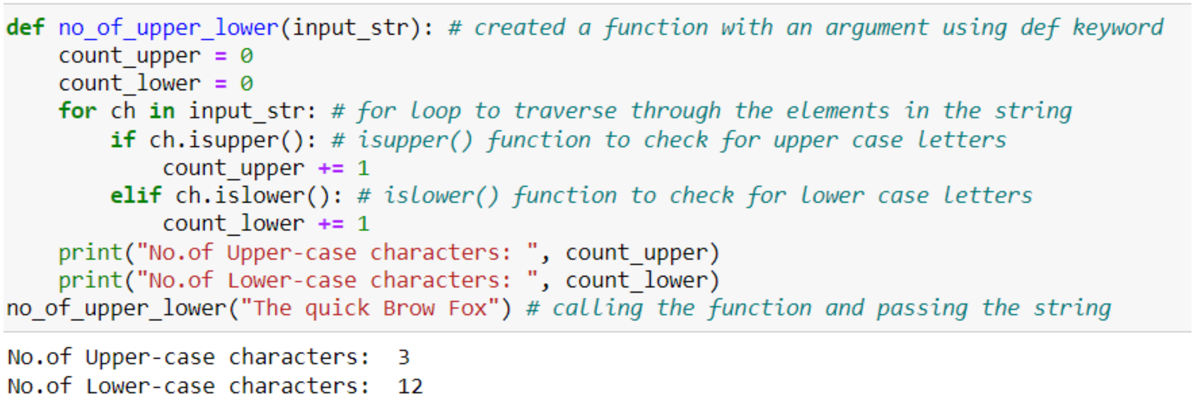
5. Write a function that accepts a string and calculate the number of upper-case letters and lower-case letters.

Input String: 'The quick Brow Fox'

Expected Output:

No. of Upper-case characters: 3

No. of Lower-case Characters: 12



```python
def no_of_upper_lower(input_str): # created a function with an argument using def keyword
    count_upper = 0
    count_lower = 0
    for ch in input_str: # for loop to traverse through the elements in the string
        if ch.isupper(): # isupper() function to check for upper case letters
            count_upper += 1
        elif ch.islower(): # islower() function to check for lower case letters
            count_lower += 1
    print("No.of Upper-case characters: ", count_upper)
    print("No.of Lower-case characters: ", count_lower)
no_of_upper_lower("The quick Brow Fox") # calling the function and passing the string
```

```
No.of Upper-case characters:  3
No.of Lower-case characters:  12
```

For the fifth question I have created a new function no_of_upper_lower with an argument. Called the function and passed the given string to the function.

In the function I have initialized two variables with ‘0’ to update the count of the upper- and lower-case letters.

Next with the help of for loop I have traversed each character in the string and checked for upper case using isupper() function and incremented the count if one found else used islower() function and incremented the lower case count if one found.