

IN CLASS PROGRAMMING \_ 2  
MACHINE LEARNING

700743010

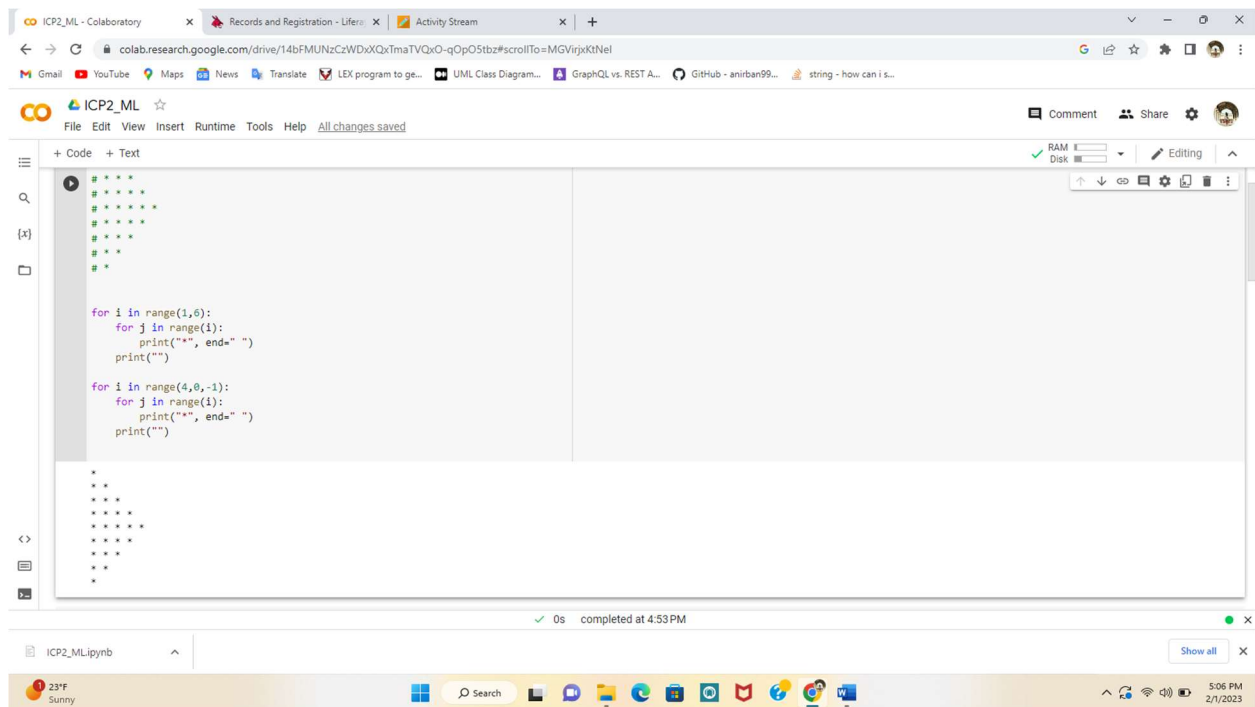
Praveena Goli

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

```
# *  
# * *  
# * * *  
# * * * *  
# * * * * *  
# * * * * *  
# * * *  
# * *  
# *
```

```
for i in range(1,6):  
    for j in range(i):  
        print("*", end=" ")  
    print("")
```

```
for i in range(4,0,-1):  
    for j in range(i):  
        print("*", end=" ")  
    print("")
```



Here in the first question I took i and j variables and for the range (1,6) I incremented and for range(4,0,-1) I decremented eventually and printed the pattern accordingly.

```
# 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]

my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
for index in range(1, len(my_list), 2):
    print(my_list[index])
```

```
[ ] # 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]

my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
for index in range(1, len(my_list), 2):
    print(my_list[index])

20
40
60
80
100

[ ] # 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'>]
```

Here in the second question, I have printed the elements in odd indexes using for loop and range until all the elements are done in the list.

```
# 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]
```

```
types = []
```

```
for i in x:
    types.append(type(i))
```

```
print(x)
print(types)
```

The screenshot shows a Google Colab notebook with two code cells. The first cell contains the following code:

```
[ ]  
88  
100  
  
[ ] # 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]  
  
types = []  
  
for i in x:  
    types.append(type(i))  
  
print(x)  
print(types)  
  
[23, 'Python', 23.98]  
[<class 'int'>, <class 'str'>, <class 'float'>]  
  
[ ] # 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]
```

The second cell contains the following code:

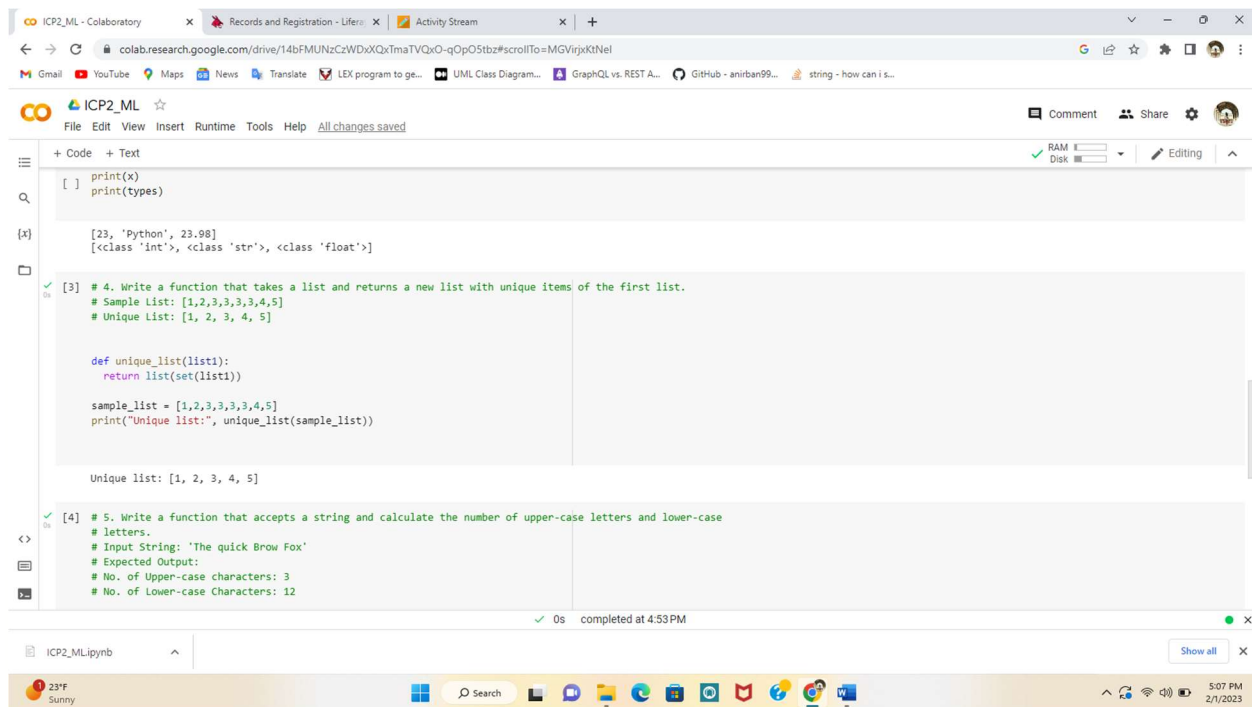
```
def unique_list(list1):  
    return list(set(list1))  
  
sample_list = [1,2,3,3,3,3,4,5]  
print("Unique list:", unique_list(sample_list))
```

Here in the third question, created the list of given elements as x, and have created an empty list types and using for loop and range , I have appended the types of the given elements accordingly.

```
# 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 unique_list(list1):  
    return list(set(list1))
```

```
sample_list = [1,2,3,3,3,3,4,5]  
print("Unique list:", unique_list(sample_list))
```

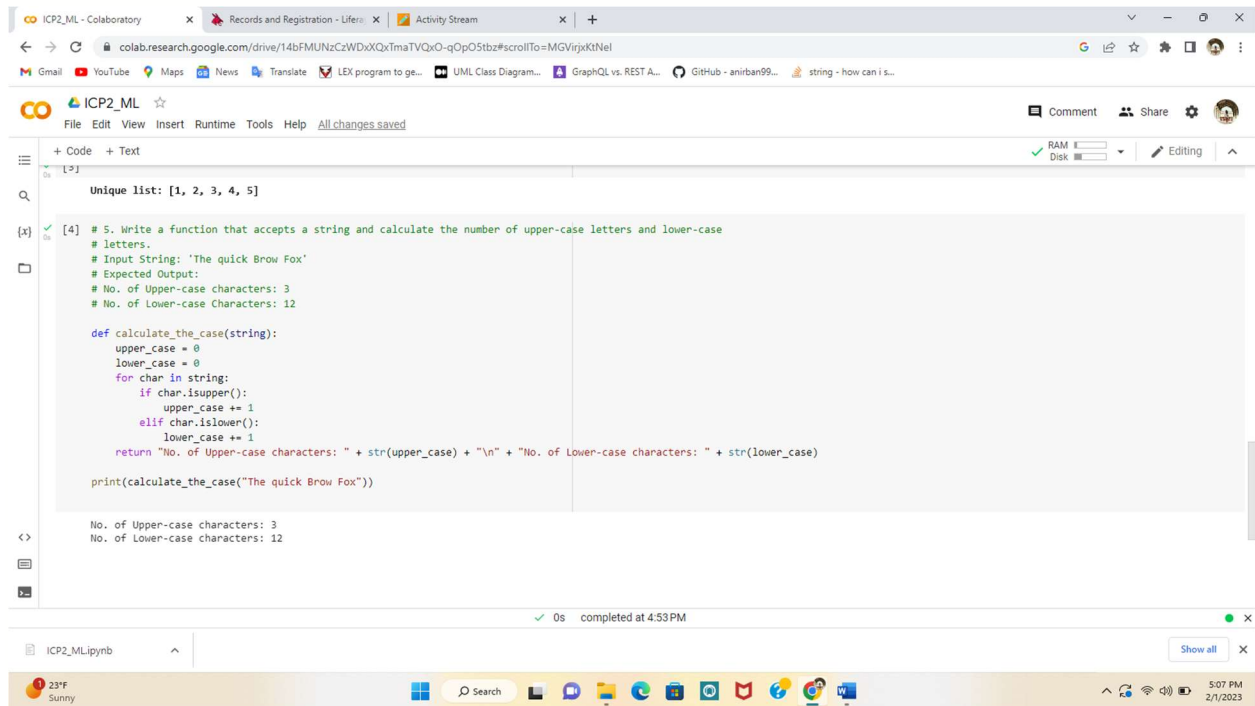


Here in the fourth question, I have printed the unique elements list using set and a function.

```
# 5. Write a function that accepts a string and calculate the number of up
per-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
```

```
def calculate_the_case(string):
    upper_case = 0
    lower_case = 0
    for char in string:
        if char.isupper():
            upper_case += 1
        elif char.islower():
            lower_case += 1
    return "No. of Upper-
case characters: " + str(upper_case) + "\n" + "No. of Lower-
case characters: " + str(lower_case)
```

```
print(calculate_the_case("The quick Brow Fox"))
```



The screenshot shows a Google Colab notebook titled "ICP2\_ML". The notebook contains a Python function named `calculate_the_case` that takes a string as input and returns the number of upper-case and lower-case characters. The function uses `isupper()` and `islower()` methods to count the characters. The output of the function is displayed as "No. of Upper-case characters: 3" and "No. of Lower-case characters: 12".

```
[4] # 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

def calculate_the_case(string):
    upper_case = 0
    lower_case = 0
    for char in string:
        if char.isupper():
            upper_case += 1
        elif char.islower():
            lower_case += 1
    return "No. of Upper-case characters: " + str(upper_case) + "\n" + "No. of Lower-case characters: " + str(lower_case)

print(calculate_the_case("The quick Brow Fox"))
```

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

0s completed at 4:53 PM

Here in the fifth question, I have counted the upper case and lower case characters in a given sentence , using `isUpper` and `isLower` and if-else conditions.

Repo: [https://github.com/Golil18/ICP2\\_ML.git](https://github.com/Golil18/ICP2_ML.git)

Video Link: <https://www.veed.io/view/f4050eb4-8e8a-4959-a2cd-b218e672ebe9?source=compressor-sharing>