ASSINGMENT 2

Name: varna nemulla

ID: 700744920

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

```
        rows = 5     #taking standard row number as 5
for i in range(0, rows):    #using for loop to dispaly the star pattern from 0 to rows
   for j in range(0, i + 1):
      print("*", end=' ') #Displaying '*' in the output
   print("\r")


for i in range(rows, 0, -1):  #using for loop to dispaly the star pattern from 0 to -1
   for j in range(0, i - 1):
      print("*", end=' ')   #Displaying '*' in the output
   print("\r")
```

Description: In the above code I have used a for loop and standard row value as 5 to get the star pattern up to 5 and decrease the star pattern I have given the range from rows to -1 . then printed the printed the star display as mentioned in the problem.

Screenshot of source code and output:

jupyter ML Assignment 2 Last Checkpoint: 28 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python

In [1]:

```
rows = 5     #taking standard row number as 5
for i in range(0, rows):    #using for loop to dispaly the star pattern from 0 to rows
    for j in range(0, i + 1):
        print("*", end=' ') #Displaying '*' in the output
    print("\r")

for i in range(rows, 0, -1):  #using for loop to dispaly the star pattern from 0 to -1
    for j in range(0, i - 1):
        print("*", end=' ')   #Displaying '*' in the output
    print("\r")
```

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Question 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]

# stat from index 1 with step 2( means 1, 3, 5, an so on)

for i in my_list[1::2]:

   print(i, end=" ")
```

Screenshot of source code and output:

jupyter ML Assignment 2 Last Checkpoint: 29 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trus

In [2]:

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# stat from index 1 with step 2( means 1, 3, 5, an so on)
for i in my_list[1::2]:
    print(i, end=" ")
```

```
20 40 60 80 100
```

Description: In here I have used for loop from index 1 with step 2 . and printed the even value indexes.

Question 3 **Write a code that appends the type of elements from a given list.**

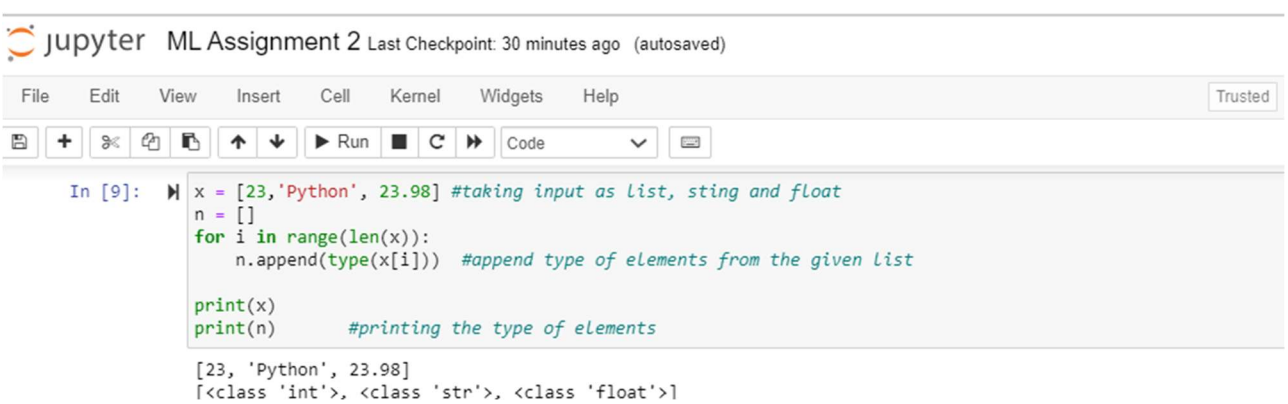
Input x = [23, 'Python', 23.98]

Expected output [23, 'Python', 23.98] [, ,]:

```
x = [23,'Python', 23.98] #taking input as list, string and float

n = []

for i in range(len(x)):

   n.append(type(x[i]))  #append type of elements from the given list


print(x)

print(n)      #printing the type of elements
```

Description: In the above program we are taking input as list, string, and float and given n as empty index then appended the type of the element and printed the output

Screenshot of source code and output:



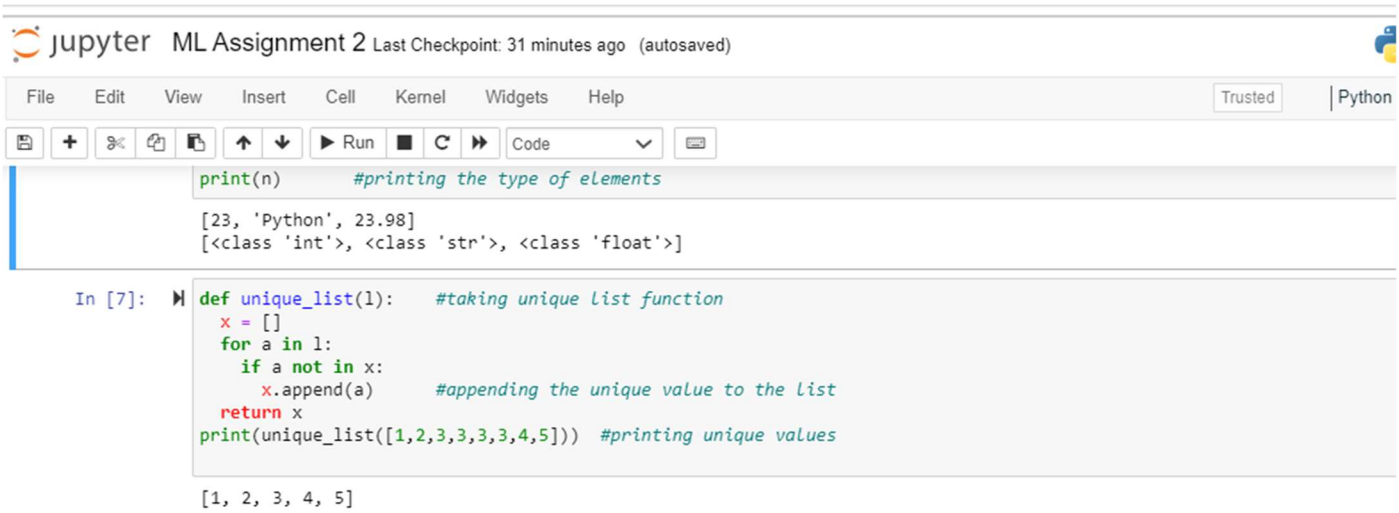
Question 4: **Write a function that takes a list and returns a new list with unique items of the first list.**

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

```
def unique_list(l):    #taking unique list function
 x = []
 for a in l:
  if a not in x:
   x.append(a)     #appending the unique value to the list
 return x
print(unique_list([1,2,3,3,3,3,4,5]))  #printing unique values
```

Description: In this problem defining the unique list function as I and given X as an empty index then appending the value to the output then retuning x so that it will print all the unique values.

Screenshot of source code and output:



Question 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 string_test(s):        #using  string_test to store the string
```

```
d={"UPPER_CASE":0, "LOWER_CASE":0}
for c in s:
    if c.isupper():     #generating upper_case letter
      d["UPPER_CASE"]+=1
    elif c.islower():     #generating Lower_case letter
      d["LOWER_CASE"]+=1
    else:
      pass
print ("Original String : ", s)
print ("No. of Upper case characters : ", d["UPPER_CASE"])
print ("No. of Lower case Characters : ", d["LOWER_CASE"])
```

string_test('The quick Brown Fox') #original string

Description: In this defining a function which accepts the string and c.isupper to get the upper _case values and c.islower to get the lower_case values and also giving original string then printing the output as number of upper_case and lower_case values.

Screenshot of source code and output:

```
In [8]:  def string_test(s):            #using  string_test to store the string
             d={"UPPER_CASE":0, "LOWER_CASE":0}
             for c in s:
                 if c.isupper():     #generating upper_case letter
                    d["UPPER_CASE"]+=1
                 elif c.islower():     #generating Lower_case letter
                    d["LOWER_CASE"]+=1
                 else:
                    pass
             print ("Original String : ", s)
             print ("No. of Upper case characters : ", d["UPPER_CASE"])
             print ("No. of Lower case Characters : ", d["LOWER_CASE"])

         string_test('The quick Brown Fox') #original string

         Original String :  The quick Brown Fox
         No. of Upper case characters :  3
         No. of Lower case Characters :  13

In [ ]:
```

Video Link: https://drive.google.com/file/d/1q-Ro2udsRR7XSVmVqYDQpxLu7bYnEUxV/view?usp=sharing

GitHub Link: