Python assignment1

Name - Essamaraju Sasi kiran
Roll No - 20075031
Batch id - BA 1

---

1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given string.
If the string length is less than 2, return instead of the empty string.
Sample String : 'a1resource'
Expected Result : 'a1ce'
Sample String : 'a1'
Expected Result : 'a1a1'
Sample String : ' a'
Expected Result : Empty String

demo.py

```
1  def getString(s):
2      if len(s) < 2:
3          return " "
4      return s[:2] + s[-2:]
5
6  string = input("Enter the string - ")
7  print(getString(string))
8
```

```
E:\pythonfiles>python demo.py
Enter the string - a1resource
a1ce

E:\pythonfiles>
```

2. Write a Python program to get a string from a given string where all occurrences of its first char have
been changed to '$', except the first char itself.
Sample String : 'restart'
Expected Result : 'resta$t'

demo.py

```
1  def getString(s):
2      return s[0] + s[1:].replace(s[0], '$')
3
4  string = input("Enter the string - ")
5  print(getString(string))
6
```

```
E:\pythonfiles>python demo.py
Enter the string - restart
resta$t

E:\pythonfiles>
```

3. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.
Sample String : 'The lyrics is not that poor!'
'The lyrics is poor!'
Expected Result : 'The lyrics is good!'
'The lyrics is poor!'

demo.py

```
def modify(s):
    not_index = s.find("not")
    poor_index = s.find("poor")
    if not_index != -1 and poor_index != -1 and not_index < poor_index:
        return s[:not_index] + "good" + s[poor_index + 4 :]
    else:
        return s


string = input("Enter the string - ")
print(modify(string))
```

```
E:\pythonfiles>python demo.py
Enter the string - The lyrics is not that poor!
The lyrics is good!

E:\pythonfiles>
```

4. Write a Python function to insert a string in the middle of a string.
Sample function and result :
insert_sting_middle('[[]]', 'Python') -> [[Python]]
insert_sting_middle('{{}}', 'PHP') -> {{PHP}}

demo.py

```
1  def insertStringMiddle(a, b):
2      mid = len(a) // 2
3      return a[:mid] + b + a[mid:]
4
5  main = input("enter the main string - ")
6  sub = input("enter the string to be inserted - ")
7  print(insertStringMiddle(main, sub))
8
```

```
E:\pythonfiles>python demo.py
enter the main string - {{}}
enter the string to be inserted - Python
{{Python}}

E:\pythonfiles>
```

5. Write a Python program to sort a string lexicographically.

demo.py

```
1  string = input("Enter the string - ")
2  lst = sorted(string)
3  print(''.join(lst))
4
```

```
E:\pythonfiles>python demo.py
Enter the string - sasikiran
aaiiknrss

E:\pythonfiles>
```

6. Write a Python program to count repeated characters in a string.
Sample string: 'thequickbrownfoxjumpsoverthelazydog'
Expected output :
o 4
e 3
u 2
h 2
r 2
t 2

demo.py

```
1  from collections import Counter
2  string = input("Enter the string - ")
3  for key, value in Counter(string).items():
4      if value > 1:
5          print(key, value)
6
```

```
E:\pythonfiles>python demo.py
Enter the string - thequickbrownfoxjumpsoverthelazydog
t 2
h 2
e 3
u 2
r 2
o 4

E:\pythonfiles>
```

7. Write a Python function that prints out the first ‘n’ rows of Pascal's triangle. ‘n’ is user input.

demo.py

```python
from math import factorial

def C(n, r):
    """ returns the value of nCk """
    return factorial(n) // (factorial(n - r) * factorial(r))

def printPascal(n):
    """ prints first n rows in pascal's triangle """
    for i in range(n):
        print(' ' * (n - 1 - i), end="")
        for j in range(i + 1):
            print(C(i, j), end= " ")
        print()

printPascal(5)
```

```
E:\pythonfiles>python demo.py
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1

E:\pythonfiles>
```

8. Write a Python function that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.
Sample Items : green-red-yellow-black-white
Expected Result : black-green-red-white-yellow

demo.py

```
1  def sortHyphenInput(s):
2      words = s.split('-')
3      words.sort()
4      print('-'.join(words))
5
6
7  string = input("Enter the hyphen separated string : ")
8  sortHyphenInput(string)
9
```

```
E:\pythonfiles>python demo.py
Enter the hyphen separated string : green-red-yellow-black-white
black-green-red-white-yellow

E:\pythonfiles>
```

9. Write a Python function to merge two sorted arrays.
Sample Items :
m=[1, 4, 8]
n=[2, 3, 7]
Expected Output: mn=[1, 2, 3, 4, 7, 8]

demo.py

```python
def merge(list1, list2):
    newlist = []
    i = j = 0
    while i < len(list1) and j < len(list2):
        if list1[i] <= list2[j]:
            newlist.append(list1[i])
            i += 1
        else:
            newlist.append(list2[j])
            j += 1
    while i < len(list1):
        newlist.append(list1[i])
        i += 1
    while j < len(list2):
        newlist.append(list2[j])
        j += 1

    return newlist


m = [1, 4, 8]
n = [2, 3, 7]
print(merge(m, n))

```

```
[1, 2, 3, 4, 7, 8]
[Finished in 110ms]
```

10. Write a Python function that accepts a string and calculate the number of upper case letters and
lower case letters.
Sample String : 'The quick Brow Fox'
Expected Output :
No. of Upper case characters : 3
No. of Lower case Characters : 12

demo.py

```
def countUpperLower(s):
    upperCount = lowerCount = 0
    for i in s:
        if i >= 'A' and i <= 'Z':
            upperCount += 1
        elif i >= 'a' and i <= 'z':
            lowerCount += 1

    return lowerCount, upperCount

string = input("Enter the string : ")
lower, upper = countUpperLower(string)
print(f"No of uppercase characters : {upper}")
print(f"No of lowercase characters : {lower}")

```

```
E:\pythonfiles>python demo.py
Enter the string : The quick Brow Fox
No of uppercase characters : 3
No of lowercase characters : 12

E:\pythonfiles>
```

11. Write a Python program which accepts a sequence of comma separated 4 digit binary numbers as
its input and print the numbers that are divisible by 5 in a comma separated sequence.
Sample Data : 0100,0011,1010,1001,1100,1001
Expected Output : 1010

demo.py

```
vals = input("enter - ").split(',')
for val in vals:
    if int(val, 2) % 5 == 0:
        print(val, end=" ")

```

```
E:\pythonfiles>python demo.py
enter - 0100,0011,1010,1001,1100,1001
1010
E:\pythonfiles>
```

12. Write a Python program to insert a new item before the second element in an existing array.

demo.py

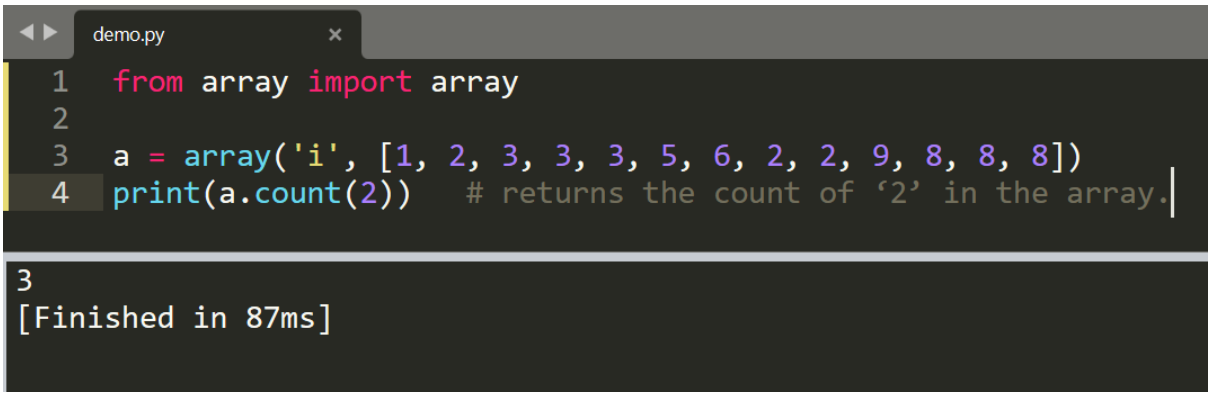
```
from array import array
arr = array('i', [1, 2, 3, 4, 5, 6])
arr.insert(2, 0)    # insert value 0 before position 2
print(arr)
```

```
array('i', [1, 2, 0, 3, 4, 5, 6])
[Finished in 78ms]
```

13. Write a Python program to print the alphabet pattern 'R'.
Expected Output:

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

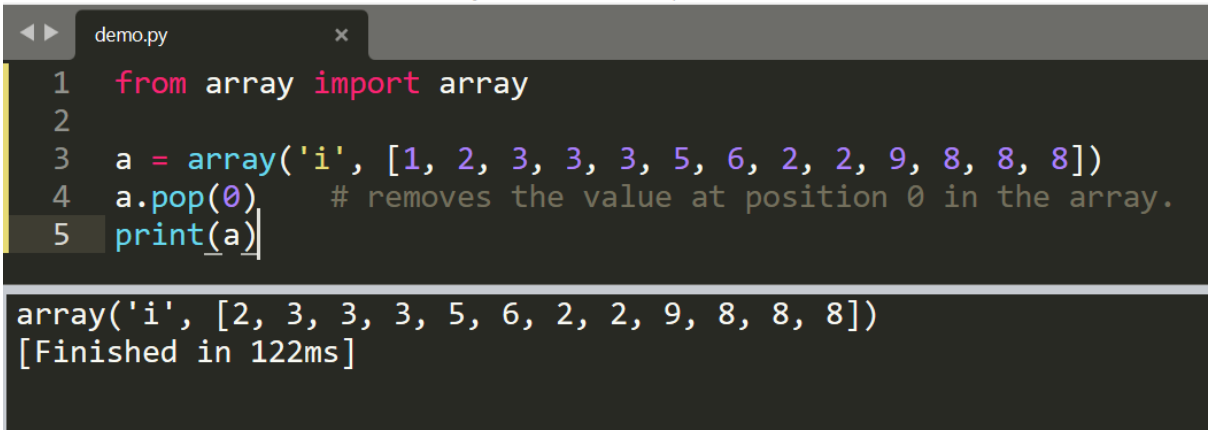
14. Write a Python program to get the number of occurrences of a specified element in an array.



```
demo.py
1  from array import array
2
3  a = array('i', [1, 2, 3, 3, 3, 5, 6, 2, 2, 9, 8, 8, 8])
4  print(a.count(2))   # returns the count of '2' in the array.
```

```
3
[Finished in 87ms]
```

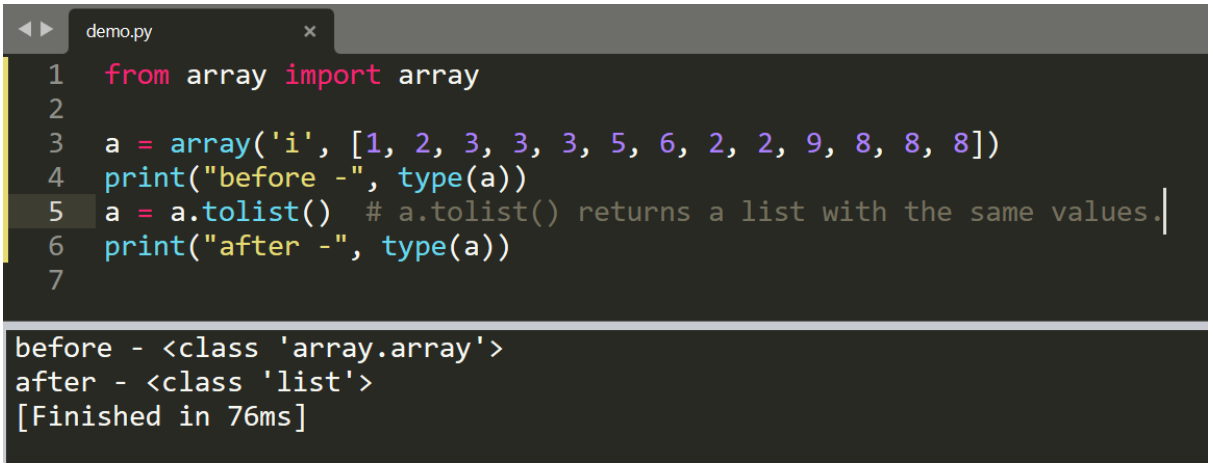
15. Write a Python program to remove a specified item using the index from an array.



```
demo.py
1  from array import array
2
3  a = array('i', [1, 2, 3, 3, 3, 5, 6, 2, 2, 9, 8, 8, 8])
4  a.pop(0)    # removes the value at position 0 in the array.
5  print(a)
```

```
array('i', [2, 3, 3, 3, 5, 6, 2, 2, 9, 8, 8, 8])
[Finished in 122ms]
```

16. Write a Python program to convert an array to an ordinary list with the same items.



```
demo.py
1  from array import array
2
3  a = array('i', [1, 2, 3, 3, 3, 5, 6, 2, 2, 9, 8, 8, 8])
4  print("before -", type(a))
5  a = a.tolist()  # a.tolist() returns a list with the same values.
6  print("after -", type(a))
7
```

```
before - <class 'array.array'>
after - <class 'list'>
[Finished in 76ms]
```

17. Write a Python program to remove duplicates from a list of lists.
Sample list : [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
New List : [[10, 20], [30, 56, 25], [33], [40]]

demo.py

```
sample = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
# creating a set of tuple.
new = set(tuple(item) for item in sample)
# converting tuple into list.
sample = [list(item) for item in new]
print(sample)
```

```
[[40], [33], [30, 56, 25], [10, 20]]
[Finished in 85ms]
```

demo.py

```
sample = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
new = []
for i in sample:
    if i not in new:
        new.append(i)

print(new)
```

```
[[10, 20], [40], [30, 56, 25], [33]]
[Finished in 88ms]
```

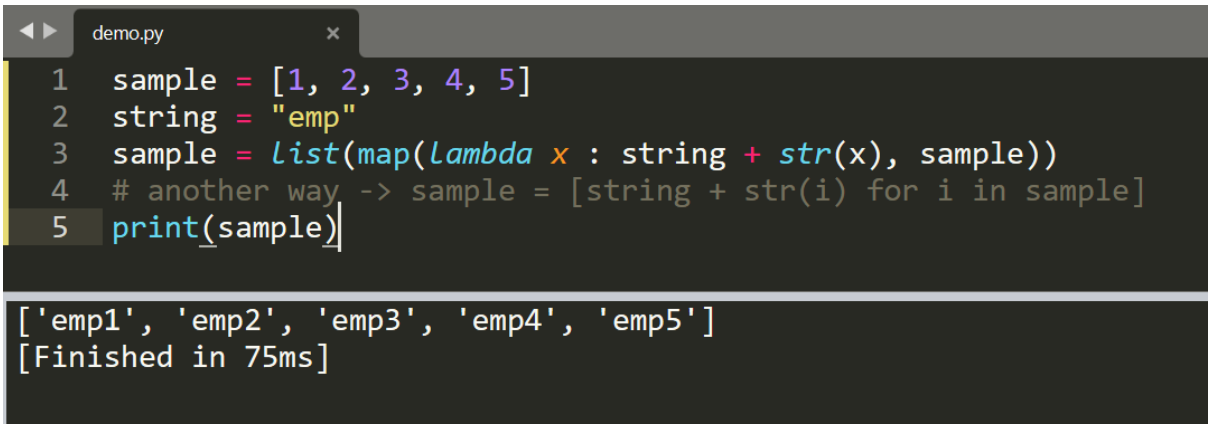
18. Write a Python program to find the list in a list of lists whose sum of elements is the highest.
Sample lists: [1,2,3], [4,5,6], [10,11,12], [7,8,9]
Expected Output: [10, 11, 12]

demo.py

```
sample = [[1, 2, 3], [4, 5, 6], [10, 11, 12], [7, 8, 9]]
maxList = sample[0]
for i in range(1, len(sample)):
    if sum(sample[i]) > sum(maxList):
        maxList = sample[i]
print(maxList)
```

```
[10, 11, 12]
[Finished in 76ms]
```

19. Write a Python program to insert a given string at the beginning of all items in a list.
Sample list : [1,2,3,4], string : emp
Expected output : ['emp1', 'emp2', 'emp3', 'emp4']



20. Write a Python program to compute the similarity between two lists.
Sample data: ["red", "orange", "green", "blue", "white"], ["black", "yellow", "green", "blue"]
Expected Output:
Color1-Color2: ['white', 'orange', 'red']
Color2-Color1: ['black', 'yellow']

21. Write a Python program to split a list every Nth element.
Sample list: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']
Expected Output: [['a', 'd', 'g', 'j', 'm'], ['b', 'e', 'h', 'k', 'n'], ['c', 'f', 'i', 'l']]

demo.py

```python
def SplitList(lst, n):
    new = []
    for i in range(n):
        new.append(lst[i::n])
    return new

sample = list(input("Enter the list : ").split())
n  = int(input("Enter n : "))
print(SplitList(sample, n))
```

```
E:\pythonfiles>python demo.py
Enter the list : a b c d e f g h i j k l m n
Enter n : 3
[['a', 'd', 'g', 'j', 'm'], ['b', 'e', 'h', 'k', 'n'], ['c', 'f', 'i', 'l']]
```

22. Write a Python program to convert a list of multiple integers into a single integer.
Sample list: [11, 33, 50]
Expected Output: 113350

demo.py

```python
from functools import reduce

sample = [11, 33, 50]
sample = int(reduce(lambda x, y : str(x) + str(y), sample))
print(sample)
```

```
113350
[Finished in 81ms]
```

23. Write a Python program to iterate over two lists simultaneously

demo.py

```
list1 = [2, 5, 9, 7, 10, 8]
list2 = [10, 7, 3, 5, 2, 4]
# zip function iterates till any of the list get's exhausted.
for i, j in zip(list1, list2):
    print(f"{i} + {j} = {i + j}")

```

```
2 + 10 = 12
5 + 7 = 12
9 + 3 = 12
7 + 5 = 12
10 + 2 = 12
8 + 4 = 12
[Finished in 75ms]
```

24. Write a Python program to create a list by concatenating a given list which range goes from 1 to n.
Sample list : ['p', 'q']
n =5
Sample Output : ['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']

demo.py

```
sample = list(input("Enter values in list - ").split())
n = int(input("Enter count - "))
new = []
for i in range(1, n + 1):
    new.extend([str(item) + str(i) for item in sample])
print(new)

```

```
E:\pythonfiles>python demo.py
Enter values in list - a b
Enter count - 5
['a1', 'b1', 'a2', 'b2', 'a3', 'b3', 'a4', 'b4', 'a5', 'b5']

E:\pythonfiles>
```

Name - Essamaraju Sasi kiran
Roll No - 20075031
E-mail - essamaraju.sasikiran.cse20@itbhu.ac.in