

## Week - 4: Sets, Dictionaries and Strings

1. Write a function called `is_sorted` that takes a list as a parameter and returns `True` if the list is sorted in ascending order and `False` otherwise. **Program:**
- ```
def is_sorted(input_list):
    for i in range(len(input_list) - 1):
        if input_list[i] > input_list[i + 1]:
            return False
    return True

list1 = [1, 2, 3, 4, 5]
list2 = [5, 3, 8, 2, 10]
```

```
print(is_sorted(list1))
print(is_sorted(list2))
```

**output:**

```
True
False
```

2. Write a function called `has_duplicates` that takes a list and returns `True` if there is any element that appears more than once. It should not modify the original list.

**Program:**

```
def has_duplicates(list1):
    a = len(list1)
    for i in range(0, a - 1):
        for j in range(i + 1, a):
            if list1[i] == list1[j]:
                return True
    return False

list1 = [1, 2, 6, 4, 5, 1]
c = has_duplicates(list1)
print(c)
```

**ou**

tp  
ut:  
Tr  
ue

i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. **Hint: they don't have to be in the same order.**

**Program:** def  
remove\_duplicates(input\_list):  
unique\_list = list(set(input\_list))  
return unique\_list  
original\_list =  
[2, 3, 2, 5, 6, 5, 8, 2, 9]  
new\_list =  
remove\_duplicates(original\_list)  
print(new\_list) **output:**  
[2, 3, 5, 6, 8, 9]

ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.

**Program:**

**Output:**

iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.

**Program:**

```
original_dict = {'a': 1, 'b': 2, 'c': 3}  
interchanged_dict = {value: key for key, value in original_dict.items()}
```

```
print(interchanged_dict)
```

t) **output:**

```
{1: 'a', 2: 'b', 3: 'c'}
```

3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'. **program:** `x="apple"`

```
y=','.join(x) print(y) output: A,p,p,l,e
```

ii) Remove the given word in all the places in a string?

**Program:**

```
x="This is just a test."  
y=x.replace("is","") print(y)
```

**output:**

```
Th just a test.
```

**iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?**

**Program:** def capitalizing(sentence):  
words = sentence.split()  
cap\_words=[] for i in words:  
cap\_words.append(i.capitalize()) cap\_sentence = '.join(cap\_words)  
print(cap\_sentence)  
input\_sentence = input("Enter a sentence: ")  
capitalizing(input\_sentence)

**Output:**

Enter a sentence: hello world hiii!

Hello World Hii!

**4. Writes a recursive function that generates all binary strings of n-bit length Program:** def generate\_binary\_strings(n, prefix=""):

if n == 0:

print(prefix)

else:

generate\_binary\_strings(n - 1, prefix + "0")

generate\_binary\_strings(n - 1, prefix + "1")

n\_bits = 4

generate\_binary\_strings(n\_bits)

**output:**

0000

0001

0010

0011

0100

0101

0110

0111  
1000  
1001  
1010  
1011  
1100  
1101  
1110  
1111

**5. Write a Python program to implement all set operations**

**Program:**

```
set1={1,2,3,4,5}
set2={3,4,5,6,7}
union_result=set1.union(set2)
union_intersection=set1.intersection(set2)
union_difference=set1.difference(set2)
symmetric_difference_result =
set1.symmetric_difference(set2) print("Union:
",union_result) print("Intersection:
",union_intersection) print("Difference :
",union_difference) print("symmetric difference :
",symmetric_difference_result) output:
Union: {1, 2, 3, 4, 5, 6, 7}
Intersection: {3, 4, 5}
Difference : {1, 2}
symmetric difference : {1, 2, 6, 7}
```

**6. Write a program to check whether a string is palindrome or not.**

**Program:**

```
def
palindrome(x):
for i in
range(0,len(x)):
if(x[i]==x[len(x)-
i-1]):
return True
else:
return False
```

```
Input=input("enter a string ")  
c=palindrome(Input)  
print(c)
```

**output:**

```
enter a  
string  
madam  
True
```