Ashwini Shamrao

1) List: [1, 2, 2, 3, 4, 1, 4, 5, 5, 6, 7, 7]

```
def occurence():
    l = [1, 2, 2, 3, 4, 1, 4, 5, 5, 6, 7, 7]
    li = []
    for i in l:
        if (i not in li):
            Print([l, 'i', l.count(i), "times"])
            li.append(i)
            return(li)
Occurance()
Occurance()
```

18) Recursive function:-

A function that calls itself is called a recursive function.

```
eg:- def sum_sum(n):
    if (n==1):
        return 1
    else
        return n+sum_sum(n-1)
    Print (sum_sum(4))
```

16) Built-in types

1) Numeric type - int, float, complex
2) Text type - String
3) Sequence type - list and Tuple
4) Mapping type - Dictionary

* Boolean type - bool
* None type - None

19) * lambda function is an inline function

Syntax :-

lambda arguments : expression

* we can use any no of arguments but only one expression.

* use of lambda functions :-

lambda functions mainly used in higher order functions.

Python higher order functions are :-

1) map() :-

map() is used to apply the given function to each elements in the given iterable and return a new map object

eg: n = [1, 2, 3, 4] —> square of elements

map(lambda x : x ** 2, n)

2) filter() :-

filter function is used to filter the elements from the sequence after applying given function to each element of the given sequence.

eg :- n = [1, 2, 3, 4, 5, 6]

filter(lambda x : x % 2 == 0, n)

3) Reduce ():.

Syntax:

import functools

functools.reduce(function, iterable)

4 Reduce function used to reduce the given sequence into a single value after applying function to it.

15) Fibonacci Series

```
a, b = 0, 1
count = 0
while (count <= 10)
 Print (a)
 a, b = b, a+b
 count = count + 1
```

16) program to check a string is palindrome

```
s = input ("Enter the string: ")
rev = ""
for i in range (len(s) -1, -1, -1):
    rev = rev + s[i]
Print (rev)
If (rev == s):
    Print ("String is Palindrome")
else:
    Print ("String is not Palindrome")
```

14) Armstrong

```
n = input ("Enter the number")
l = len(n)
sum = 0
for i in n:
    sum = sum + int(i)** l
if (sum == int(n)):
    Print ("Number is Armstrong")
else:
    Print ("Number is not armstrong")
```

3)

```
def remove-duplicate (list):
    new_list=[ ]
    for i in list:
        if (i not in new_list):
            new_list.append(i)
    return new-list
list = [1,1,2,3,4,5,6,6]
result = remove-duplicate (list)
Print (result)
```

o/p: [1, 2, 3, 4, 5, 6]

8)
```
words = ['banana', 'cherry', 'apricot', 'kiwi',
         'coconut', 'avacado', 'apple']
for i in words:
    If i.startswith('a') or i.startswith('A'):
        Print(i)
        break.
```

11)
```
s = input("Enter the string"
s1 = s.split()
longest_word = ""
for i in s1:
    If (len(i) > len(longest_word):
        longest_word = i
Print
Print("Longest word:", longest_word).
```