Python

1. Predict output of
   a. print([i+j for i in "abc" for j in "def"])
      a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’].
      b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]].
      c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]].
      d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’].
   b. print([i.lower() for i in "HELLO"])
      a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’].
      b) ‘hello’
      c) [‘hello’].
      d) Hello
   c. text = "Zero One Two Three Four Five Six Seven Eight Nine"
      result = [word[0]+word[-1] for word in text.split()]
      print(result)
   d. text = "Zero One Two Three Four Five Six Seven Eight Nine"
      result = [word[0]+word[-1] for word in text.split() if word[0] > word[-1]]
      print(result)
   e. text = "bangalore : city with lakes and punctures"
      result = [word for word in text.split() if word.startswith((‘a’,’e’,’l’,’o’,’u’))]
      print(result)

2. Convert to list comprehension:

```
L = [10,20,30,40]
D = []
for i in L:
    D.append(i/10)
```

```
L = []
for x in range(10):
    if x % 2 == 0:
        L.append(x)
```

```
word = 'aLphaBEts'
count = 0
for char in word:
    if char in 'aeiouAEIOU':
        count += 1
```

```
word = 'aLphaBEts'
new_word = []
for char in word:
    if char.isupper():
        new_word.append(char.lower())
    else:
        new_word.append(char.upper())
```

3. Consider a list of words:
   Words = [‘Python’, ‘Object’, ‘Oriented’, ‘Language’]
   Write a loop to store the first character of each word in a list from the above list.
   Update the program to use list comprehension instead.

4. Input a string from user, and print only those words whose length is more than 5 characters.
5. WAP to take a string as a command line argument and print whether it is palindrome or not.
6. Find Output of:

```
word = 'synonymous'
g = ['a','o','n']
s = [ch if ch in g else '_' for ch in word]
s = ' '.join(s)
print('_' in s, s)
```

7. Write a list comprehension to store the following in a list: [Use nested and simple list comprehension both]
   [‘w’, ‘wo’, ‘wor’, ‘word’, ‘words’]
8. WAP to input 2 string from command line and search whether the first string is present in second one or not.