1) . What is the difference between enclosing a list comprehension in square brackets and parentheses?

In [ ]:

```
enclosing a list comprehension in square brackets will results a list object
enclosing a list comprehension in square parentheses will results a generator object
```

2) What is the relationship between generators and iterators?

In [ ]:

```
All the generators are iterators nut not vice versa
All the sequences are iterators
The Iterators mean we can travel from one element to the other itratively or sequentially
The Generators are like funtions whhich will result which are traversed Itertatively using next() fun
```

3) What are the signs that a function is a generator function?

In [ ]:

```
In generatoes no return type is available
In Genrerators we will produce output using Yield key word
In Gnerators will not provide enitre anser in single step
we have to go to the next generated anser using next() function
```

4) What is the purpose of a yield statement?

In [ ]:

```
This will turn a function into generator
This will produce the anser on demand

```

5) What is the relationship between map calls and list comprehensions? Make a comparison and contrast between the two.

In [ ]:

```
compare:
Both will provide element wise operations on the literables
Both will support Lambda functions
in map we will specify the function outisde and we will just give the name
but in list comprehension we can either define fuction insdie or out side like map()

contrast:
The map call returs a map object which we nedd to furher convert into list
In list comprhension default is list
map is a function
list comprehension is a logical impletmentation
```