In [5]:

```
#QUESTION 1:
#Create an empty list. Accept 10 numbers from the user and append to it the list
def is_even_num(l):
    enum = []
    for n in l:
        if n % 2 == 0:
            enum.append(n)
    return enum
print(is_even_num([1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

```
[2, 4, 6, 8]
```

In [8]:

```
#QUESTION 2:
# Create a notebook on LIST COMPREHENSION. This exercise is to put you in a Self

# Definition:
print("List comprehensions are used for creating new lists from other iterables.
```

```
List comprehensions are used for creating new lists from other iterables. As li
st comprehensions return lists, they consist of brackets containing the express
ion, which is executed for each element along with the for loop to iterate over
each element.
```

In [13]:

```
# QUESTION 2 Continuation:
# EXAMPLES FOR LIST COMPREHENSION
#Example 1: Iterating through a string Using for Loop
m_letters = []

for letter in 'mirzapur':
    m_letters.append(letter)

print(m_letters)

# EXAMPLE 2: Iterating through a string Using List Comprehension
m_letters = [ letter for letter in 'sacredgames' ]
print( m_letters)


#Example 3: Using Lambda functions inside List
letters = list(map(lambda x: x, 'popcorn'))
print(letters)


#Example 4: Using if with List Comprehension
number_list = [ x for x in range(20) if x % 3 == 0]
print(number_list)


#Example 5: Nested IF with List Comprehension
num_list = [y for y in range(100) if y % 2 == 0 if y % 5 == 0]
print(num_list)


#Example 6: if...else With List Comprehension
obj = ["prime" if i%3==0 else "even" for i in range(10)]
print(obj)
```

```
['m', 'i', 'r', 'z', 'a', 'p', 'u', 'r']
['s', 'a', 'c', 'r', 'e', 'd', 'g', 'a', 'm', 'e', 's']
['p', 'o', 'p', 'c', 'o', 'r', 'n']
[0, 3, 6, 9, 12, 15, 18]
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
['prime', 'even', 'even', 'prime', 'even', 'even', 'prime', 'even', 'even', 'pr
ime']
```

In [1]:

```
# Question 3:
#Related to Dictionary
n=int(input("Input a number "))
d = dict()

for x in range(1,n+1):
    d[x]=x*x

print(d)
```

```
Input a number 8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

In [3]:

```
# Question 4:  Write a program to compute the distance between the current posit
#If the distance is a float, then just print the nearest integer (use round() fu
it into an integer).
import math
pos = [0,0]
while True:
    s = float(input())
    if not s:
        break
    movement = s.split(" ")
    direction = movement[0]
    steps = int(movement[1])
    if direction=="UP":
        pos[0]+=steps
    elif direction=="DOWN":
        pos[0]-=steps
    elif direction=="LEFT":
        pos[1]-=steps
    elif direction=="RIGHT":
        pos[1]+=steps
    else:
        pass

print (float(round(math.sqrt(pos[1]**2+pos[0]**2))))
```

```
4

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-3-7585e075446e> in <module>
      5     if not s:
      6         break
----> 7     movement = s.split(" ")
      8     direction = movement[0]
      9     steps = int(movement[1])

AttributeError: 'float' object has no attribute 'split'
```

In [ ]:

In [ ]: