Questions

In [1]:

```
# Creat an empty list
xs = [1,2,3,4,5,6,7,8,9,10]

even_list = []

for i in xs:
    if i % 2 == 0:
        even_list.append(i)

print(even_list)
```

```
[2, 4, 6, 8, 10]
```

In [3]:

```
# LIST COMPREHENSION
# [1,4,9,16,25,36]
el = []
for x in range(1,7):
    el.append(x ** 2)
print(el)
```

```
[1, 4, 9, 16, 25, 36]
```

In [4]:

```
# creating a empty dict
dict = {}

#  getting value for "n" from user
n = int(input())

for i in range(1,n+1):
  dict[i] = i*i

print(dict)
```

```
4
{1: 1, 2: 4, 3: 9, 4: 16}
```

In [5]:

```
#  creating a origin position
pos = {"x":0,"y":0}

# getting movement from user
n = int(input())

# for loop
for i in range (n):
    move =  input().split(" ")      # ACCEPT MOVEMENT COMMAND AND STORE AS A L
IST

    if move[0].lower() == "up":     # EXTRACT DIRECTION AND COMPARE
        pos["y"] += int(move[1])    # INCREMENT/DECREMENT APPROPRIATE CO-ORDIN
ATES

    elif move[0].lower() == "down":
        pos["y"] -= int(move[1])

    elif move[0].lower() == "left":
        pos["x"] -= int(move[1])

    elif move[0].lower() == "right":
        pos["x"] += int(move[1])

#  printing the result

print(int(round((pos["x"]**2 + pos["y"]**2)**0.5)))   # DISTANCE FROM ORIGIN
```

```
4
UP 5
DOWN 3
LEFT 3
RIGHT 2
2
```

In [ ]: