LIST COMPREHENSION EXERCISE

TASK 1: CONVERT FOR LOOP TO LIST COMPREHENSION

WORK WITH

```
simple = []
for i in range(10):
    s = i + 30
    simple.append(s)
simple
```

DESIRED OUT

[30, 31, 32, 33, 34, 35, 36, 37, 38, 39]

HINTS?

1. List comprehension doesn't have any append function or empty list.

TASK 2: CONVERT WHILE LOOP INTO LIST COMPREHENSION

WORK WITH

```
x= 0
case = []
while x < 10:
    s = "The number is " + str(x)
    case.append(s)
    x += 2
case
```

DESIRED OUTPUT

['The number is 0',
'The number is 2',
'The number is 4',
'The number is 6',
'The number is 8']

HINTS?

1. Like task 1, remove the empty list and append function.

TASK 3:CONTROL FLOW WITH MODULUS IN LIST COMPREHENSION

DESIRED OUTPUT

[1728, 2744, 4096, 5832]

HINTS?

1. Mulitply d by **3 or **pow**(d, 3).
2. Use **range**(20).
3. Use an **if** condition
4. Use modulus where d % 2
5. Use d > 10

TASK 4: CREATE TWO NESTED LIST COMPREHENSIONS INSIDE A DICTIONARY

DESIRED OUTPUT

```
d2 = {'k1': [0, 21, 42, 63, 84],
 'k2': [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]}

d3 = {'k3': [495]}
```

HINTS?

1. Use the built-in **sum()** function twice for **k3** dictionary.
2. For **k1**, use (g + 20 *g) with **range**(10) and g < 5
3. For **k2**, e*e with **range**(20) and e < 10
4. Use an **if** statement for both list comprehensions in **k1** and **k2**.

Remember, solutions are in the PDF resource for section 6!

END OF SECTION