def show_output():
    total = 0
    for number in range(9, 20):
        if number % 2 == 0 or number % 3 == 0:
            total += 1
    print(total)

def main():
    show_output()

main()
Complete the loops

- Complete the for...in loop so that the output is:

```
for number in [7, 10, 13, 16, 19, 22]:
    print(number, end = " ")
print()
```

- Complete the for...in loop so that the output is:

```
for value in [30, 25, 20, 15, 10, 5, 0, -5, -10]:
    print(value, end = " ")
print()
```
Complete the function

An amount doubles each year. Using a for...in loop complete the `double_each_year()` function which prints the growth of the parameter value, `(start_amt)` for the given number of years `(num_years)`. The first line printed by the function is the starting amount.

Each line of the output is numbered starting from the number 1. The function returns the final amount.

```python
def double_each_year(start_amt, num_years):

def main():
    print("After 4 years:", double_each_year(24, 4))
    print("After 3 years:", double_each_year(235, 3))
    print("After 5 years:", double_each_year(15, 5))
main()
```
Complete the function

- Using a for...in loop complete the `print_series()` function which prints a series of numbers starting from the parameter value, `start_num`. The second number printed is the first number plus 1, the third number is the second number plus 2, the fourth number is the third number plus 3, and so on, e.g., a series of 8 numbers starting from the number 2 is:

  2 3 5 8 12 17 23 30
  +1 +2 +3 +4 +5 +6 +7

```python
def print_series(start_num, how_many):
    # TODO: Complete the function


def main():
    print_series(2, 8)
    print_series(5, 12)
    print_series(16, 9)

main()
```
Convert - while loop ↔ for...in loop

- Convert the following while loop into a for...in range(...) loop:

```python
counter = 12
while counter < 260:
    print(counter)
    counter = counter + 10
```

- Convert the following for...in range(...) loop into a while loop:

```python
for num in range(45, 3, -5):
    print(num * 2)
```
Complete the function

- A perfect number is an integer that is equal to the sum of its divisors (excluding the number itself), e.g., $28 = 1 + 2 + 4 + 7 + 14$. Complete the `get_sum_of_divisors()` function using a `for...in range(...)` loop for the iteration.

```python
def get_sum_of_divisors(number):
    # Implementation goes here

def check_perfection(number):
    if number == get_sum_of_divisors(number):
        print(number, "is a perfect number")
    else:
        print(number, "is NOT a perfect number")

def main():
    check_perfection(28)
    check_perfection(54)
    check_perfection(496)

main()
```

28 is a perfect number
54 is NOT a perfect number
496 is a perfect number
The `get_series_sum()` function returns the sum of the given number of terms of the series:

\[
\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots
\]

e.g., `get_series_sum(4)` returns the sum of one half plus one quarter plus one eighth plus one sixteenth. Complete the function

```python
def get_series_sum(num_terms):
    # Implementation

def main():
    for num in range(1, 10):
        comment = "Terms: " + str(num) + " Sum:
        print(comment, get_series_sum(num))
main()
```

Terms: 1 Sum: 0.5
Terms: 2 Sum: 0.75
Terms: 3 Sum: 0.875
Terms: 4 Sum: 0.9375
Terms: 5 Sum: 0.9688
Terms: 6 Sum: 0.9844
Terms: 7 Sum: 0.9922
Terms: 8 Sum: 0.9961
Terms: 9 Sum: 0.998