Lab Instructions

Objective: At the end of this lab, you will be able to apply the divide-and-conquer strategy by creating your own functions to effectively solve a problem.

In class exercises

1. Write a function called **is_even(n)** that takes an integer as a parameter and returns True if the parameter is an **even number** and **False** if it is **odd**. This function is stored in the file **even.py**.
2. Write the function **is_odd(n)** that returns **True** when n is odd and **False** otherwise. This function is stored in the file **odd.py**.
3. Modify **is_odd** so that it calls the function **is_even** to determine if its parameter is an odd integer.
4. Write a function that takes a year as a parameter and determines if it is a leap year. The function returns True if the year is a leap year and False otherwise. A year is a **leap year** when:
    - It is divisible by 4 **except** when it is also divisible by 100
    - **Except** if the year is divisible also by 400 (it is a leap year then)
    - Some examples:
        - 1992 (divisible by 4 but not by 100) is a leap year
        - 2000 (divisible by 400) is a leap year
        - 1900 (divisible by 4 and 100 but not 400) is not a leap year
        - 2017 (not divisible by 4,100 or 400) is not a leap year

5. Write a function, **is_prime**, that takes a single integer parameter and returns **True** when the argument is a prime number and **False** otherwise. (Hint: use a web search to find out what a prime number is and how to determine if an integer is a prime number). After that, apply divide-and-conquer strategy by using the above function to write a program that prints all the prime numbers, which are smaller than an integer value provided by the user.

6. Apply divide-and-conquer strategy to write a program that asks the user for a positive integer n and then print out a number pattern base on n. For example, when n is 5, the pattern looks like this.

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Hint: print(i, " ", end="") leaves one whitespace after printing i and makes the cursor stay in the same line instead of going to a new line.