Some exercises are marked with an asterisk. Solve these exercises using a while loop first. Then, try to rewrite it using a for loop where possible.

Problem 1. Which is the final value of variable `x`?

```
x = 0
n = 16
while n % 2 == 0:
    x = x + 1
    n = n / 2
```

Problem 2. Write a program that computes and displays the sum of all integers from 1 to an ending number. Prompt the user for the ending number that must be greater than 0.

Problem 3. Write a program that computes and displays the multiplication of all integers from 1 to an ending number. Prompt the user for the ending number that must be greater than 0. **(*)**

Problem 4. Write a program that reads a number `n` > 0, and writes `n` asterisks. Sample output for `n = 3`:
***

Problem 5. Write a program that prints all numbers between 20 and 10 in descending order. **(*)**

```
20
19
18
...
10
```

Problem 6. Write a program that prints all even numbers between 40 and 20 in descending order. **(*)**

```
40
38
36
...
20
```

Problem 7. Write a program that prints all multiples of 3 less than 100. **(*)**

Problem 8. Write a program that computes the following mathematical expression:

$$\sum_{i=1}^{100} i^2 + \frac{1}{i}$$

Problem 9. Change the previous program to prompt the user for the limits of the summation variable `i` (`lower_limit` and `upper_limit`) and calculates and prints the mathematical expression again. Take into account: **(*)**

- `lower_limit` and `upper_limit` must be integers greater than 0
- `upper_limit` must be greater than or equal to `lower_limit`

Problem 10. Write a program that reads input from keyboard one character at a time until the user enters ".". Then, the program must display the number of "a" in the input.

Problem 11. Write a program that reads any two numbers n and m. The program must print all the numbers between n and m, including themselves, indicating, for each number if it's even or odd. **(*)**

Problem 12. Write a program that reads numbers one by one until the user enters a negative number. Then, the program must display the average of all the numbers.

Problem 13. Write a program that reads numbers, a single number at a time, until the user enters a negative number. Then, the program must display the average of all the even numbers and the average of all odd numbers.

Problem 14. Write a program that reads two integers `a` $\geq$ `0` and `b` $\geq$ `0` and calculates the product using successive additions.

Problem 15. Write a program that reads two integers `a` $\geq$ `0` and `b` $\geq$ `0` and calculates the quotient and remainder of a/b using successive subtractions.

Problem 16. Write a program that reads an integer number (`n` > 0) and prints its digits right to left. Sample output for `n = 432`: `inverted_n = 234`. Tip: You can use division and modulus operations to obtain the digits of the number. To obtain the inverted number, consider that, for instance: `432 = 400 + 30 + 2 = 4 * 10 * 10 + 3 * 10 + 2`

Problem 17. Write a program that reads an integer n >= 0 and counts the number of figures in the number. Examples:
`n = 482 → The number of figures is 3`
`n = 0 → The number of figures is 1`

Problem 18. Write a program that reads a number `n` > 0 and prints all its divisors. **(*)**

Problem 19. Write a program that reads a number `n` > 0 and prints whether that number is, or not, a perfect number. An integer is said to be a perfect number if the sum of its divisors, including 1 (but not the number itself), is equal to the number. Examples:
`6 is a perfect number because 1 + 2 + 3 = 6`
`12 is not a perfect number because 1 + 2 + 3 + 4 + 6 ≠ 12`

Problem 20. An *Armstrong number* has an interesting property that the sum of cubes of its digits is equal to the number itself (e. g., $153 = 1^3 + 5^3 + 3^3$). Write a program that prints all three-digit Amstrong numbers.

Problem 21. Write a program that reads a number `n` > 0 and prints the first `n` terms of the Fibonacci sequence. In mathematical terms, the sequence Fn of Fibonacci numbers is defined by: $F_n = F_{n-1} + F_{n-2}$ where $F_0 = 0$ and $F_1 = 1$. **(*)**

Problem 22. Write a program that implements access control to a safe. The combination will be a constant number set in the program. The program will ask the user to type a combination to open it and the user is only allowed three attempts. If the user enters an incorrect combination, the program will print the message "Incorrect Combination" plus the number of attempts left. If the combination matches, the program displays the message "The safe is open". **(*)**

Problem 23. Write a program that reads a number `n > 0`, and then writes n rows and in each row, n asterisks. Sample output for `n = 3`:

```
***
***
***
```

Problem 24. Write a program that reads two numbers `n` and `m`, both greater than 0, and writes `n` rows and in each row, `m` asterisks. Sample output for `n = 3` and `m = 4`: **(*)**

```
****
****
****
```

Problem 25. Which is the expected output?

```
rows = 5
columns = 3
while rows != 0:
    i = 1
    while i <= columns:
        print("*", end = ""),
        i += 1
    print(" ")
rows = rows - 1
```

Problem 26. Write a program that reads a number `n` > 0 and prints the following output: **(*)**

```
11 12 ... 1n
21 22 ... 2n
...   ...
n1 n2 ... nn
```

Problem 27. Write a program that reads two numbers `n` and `m`, both greater than 0 and prints the following output: **(*)**

```
11 12 ... 1m
21 22 ... 2m
...   ...
n1 n2 ... nm
```

Problem 28. Write a program that reads a number `n` > 0 and prints the following output:

```
1
1 2
1 2 3
...
1 2 3 ... n
```

Problem 29. Repeat the last exercise in inverse order. **(*)**