Lab02: The PingPong Sequence

1 Task

This lab introduces a sequence termed the *PingPong* sequence that you are tasked with computing. Here are the specific rules governing this sequence:

- For a given sequence $f(n) = \{(v_n, d_n) | v_n \in \mathbb{Z}, d_n \in \{+, -\}, n \geq 1\}$
  - $f(1) = (3, +)$
  - $v_{n+1} = 2v_n \; d_n \; 2$, for example, if $d_n = +$, then $v_{n+1} = 2v_n + 2$
  - After computing v_{n+1}, if v_{n+1} is divisible by 8 or if the last digit of its decimal representation is '8', then d_{n+1} changes to another, else $d_{n+1} = d_n$
- The following code may help you to understand the rules

```python
def calculate_next_term(v_n: int, d_n: bool):
    if d_n:
        v_next = 2 * v_n + 2
    else:
        v_next = 2 * v_n - 2

    if v_next % 8 == 0 or last_digit(v_next) == 8:
        d_next = not d_n
    else:
        d_next = d_n

    return v_next, d_next
```

You are required to devise a program that calculates $f(N)$. The value of N will be stored in x3102.

Constraints:

- When determining a term of $f(n)$, such as $f(N)$, all your arithmetic operations should be executed modulo $4096 = 2^{12}$. As a result, no term of $f(n)$ will surpass 4096.

Your Job: Compute $f(N)$ and save the result in x3103.

Examples:

N	1	2	3	4	5	6	7	8	9
$f(N)$	3	8	14	26	50	98	198	394	786
direction*	$\uparrow$ (init)	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$

$*$ The *direction*(N) in the table is after computing $f(N)$

1.1 Score

Your score will be split between correctness (50%) and the report (50%).

1.2 Submission

For this lab, you are required to use assembly code. Please adhere to the following guidelines:

1. Your program should start with `.ORIG x3000`
2. Ensure your program ends with `.END`
3. Your last instruction must be `TRAP x25` (HALT)
4. Use capitalized keywords and labels (e.g., "ADD" rather than "add").
5. Maintain spaces after commas for clarity.
6. Prefix decimal constants with `#` and hexadecimal constants with a lowercase `x`.
7. Include comments in your code where necessary for clarification.

1.3 Reports

Your report should be structured into the following sections:

1. **Purpose**: Clarify the objective of this experiment and your anticipated outcomes.
2. **Principles**: Discuss how specific operations like modulus are dealt with.
3. **Procedure**: Narrate any bugs or challenges encountered and how they were resolved.
4. **Results**: Present the outcomes of your tests.
5. **Improvements**: Respond to the question: How might you optimize the efficiency of loop structures in your program?

1.4 Something Interesting

While not required for the main report, consider pondering over these challenges:

1. By studying certain cases, can you discern any recurring patterns or periodicity in the PingPong sequence?
2. If a pattern is evident, can it be universally applied? If it's not universally applicable, provide an illustrative counterexample.

Engaging with these questions may offer a deeper insight into the sequence's characteristics.