Team Members:
Shahed Alhanbali (670025395)
Cindy Jurado (653416500)
Qudsia Sultana (657855210)

Link to GitHub Repository: https://github.com/shahedalhanbali/Project-2-366

Contribution Breakdown:

- Shahed Alhanbali:
  Worked on the iterative logic in MIPS and helped test the output for accuracy.
- Cindy Jurado:
  Assisted with debugging, wrote the README draft, and helped prepare files for submission.
- Qudsia Sultana:
  Helped write and test the Fibonacci function in MIPS and contributed to the README instructions.

Summary: In this progress report, our team implemented the function Fibonacci(n) in MIPS Assembly language using the iterative approach provided in the assignment. The logic is based on the provided Python pseudocode, where we initialize the first two Fibonacci numbers and iteratively compute the nth number.

(a) Design a MIPS program that will implement Fibonacci(n) of Figure 1. [Points : 50]

```
# fibonacci.asm

# Input: $a0 = n

# Output: $v0 = Fibonacci(n)

.text

.globl main



main:

   # If n <= 1, return n

   ble $a0, 1, base_case
```

```
    li $t0, 0          # a = 0
    li $t1, 1          # b = 1
    move $t2, $a0      # counter = n

loop:
    sub $t2, $t2, 1    # counter--
    move $t3, $t1      # temp = b
    add $t1, $t0, $t1  # b = a + b
    move $t0, $t3      # a = temp
    bgt $t2, 1, loop

    move $v0, $t1      # return b
    beq $zero, $zero, done

base_case:
    move $v0, $a0      # return n if n <= 1

done:
    nop
```

How to Run the Program:

- Open the file fibonacci.asm in MARS MIPS Simulator.
- Load the desired value of n into register $a0.
- Assemble (first build by hitting the gear icon) and run the program (green play button).
- The computed Fibonacci number will be stored in register $v0.

Sample Inputs/Outputs:

Input (n)	Fibonacci(n) (decimal)
0	0
1	1
5	5
7	13
10	55