```
.data
 message1: .asciiz "Insert a positive integer number: "
 message2: .asciiz "Fibbonaci = "

.text
 # Print user input message
 li $v0, 4
 la $a0, message1
 syscall

    # Get user input number
    li $v0, 5
    syscall
    move $s0, $v0

    # Print user result message
    li $v0, 4
    la $a0, message2
    syscall

    # Run function
    move $a0, $s0
    jal fibbonaci
    move $s1, $v0

    # Print result
    li $v0, 1
    move $a0, $s1
    syscall

    # End program
    li $v0, 10
    syscall


    # Function instructions
    fibbonaci:
        # Initialize registers
        addi $t0, $zero, 2  # Counter
        addi $t1, $zero, 1  # Previous
        addi $t2, $zero, 1  # Actual

        while:
            # Condition to stop loop
            blt $t0, $a0, continue
                j end_while
            continue:

            # Continue sequence
            add $t3, $t1, $t2
            move $t1, $t2
            move $t2, $t3

            # Update counter
            addi $t0, $t0, 1

            # Restart loop
            j while
        end_while:

        # Result return
        move $v0, $t2
        jr $ra
```