###############################################

# This program generates Fibonacci series.

# It stores the generated Fibonacci numbers first into Reg[2] ($2), and then into memory

# Assume that your data section in memory starts from address 2000. (Of course, since you will use separate memories for code and data for this part of the project, you could put data at address 0, but in the next phase of the project, you may use a single memory for both code and data, which is why we give you this program assuming a unified memory.)

addi $10, $0, 4 # number of generating Fibonacci-numbers

addi $1, $0, 1 # initializing Fib(-1) = 0

addi $2, $0, 1 # initializing Fib(0) = 1

addi $11, $0, 2000 # initializing the beginning of Data Section address in memory

addi $15, $0, 4 # word size in byte

loop: addi $3, $2, 0 # temp = Fib(n-1)

add $2, $2, $1 # Fib(n)=Fib(n-1)+Fib(n-2)

addi $1, $3, 0 # Fib(n-2)=temp=Fib(n-1)

mult $10, $15 # $lo=4\*$10, for word alignment

mflo $12 # assume small numbers

add $13, $11, $12 # Make data pointer [2000+($10)\*4]

sw $2, 0($13) # Mem[$10+2000] <-- Fib(n)

addi $10, $10, -1 # loop index

bne $10, $0, loop

EoP: beq $11, $11, EoP #end of program (infinite loop)

###############################################