MIPS Function Call Example

Write the MIPS assembly code to perform the following recursive function call to compute a factorial. For this problem, assume that n is always a non-negative number.

```
int fact (int n) {
        if ((n == 1) || (n == 0))
                return (1)
        else
                return (fact(n-1) *n)
}
```

Let: register 1 = constant 1
register 4 = n
register 2 = result

Given: register 29 = stack pointer ($sp)
register 30 = frame pointer ($fp)
register 31 = return address ($ra) for procedure/function calls

```
fact:       addi  $1, $0, 1          # initialize reg. 1 to 1
            beq   $4, $0, return1    # if (n == 0) then goto return1:
            bne   $4, $1, continue   # if (n != 1) then goto continue:
return1:    addi  $2, $1, 0          #       assign result = 1
            jr    $31                #       return

continue:   addi  $29, $29, -12      # allocate stack space for fp, ra, n
            sw    $30, 8 ($29)       # save frame pointer
            sw    $31, 4 ($29)       # save return address
            addi  $30, $29, 8        # update frame pointer
            sw    $4, 0 ($29)        # save n
            addi  $4, $4, -1         # make n-1
            jal   fact               # recursive call to fact(n-1)
            lw    $4, 0 ($29)        # restore n
            mult  $2, $4             # fact (n-1) * n
            mflo  $2                 # put product in result reg.
            lw    $31, 4 ($29)       # restore return address
            lw    $30, 8 ($29)       # restore frame pointer
            addi  $29, $29, 12       # restore stack pointer
            jr    $31                # return
```