PROBLEM 1. [10 points] Consider the following MIPS code:

```
addi $8, $0, 1
addi $9, $0, 2
addi $10,$0, 3
beq $16, $8, ONE
beq $16, $9, TWO
beq $16, $10, THREE
addi $2, $0, 0
j EXIT

ONE:    addi $2, $0, 10
        j EXIT
TWO:    addi $2, $0, 50
        j EXIT
THREE:  addi $2, $0, 100

EXIT:
    jr $31
```

1.1 In C-style pseudo-code, describe what this MIPS code computes.

PROBLEM 2. [20 points] Consider the following MIPS code:

```
Count:
    addi $sp, $sp, -8
    sw $s0, 4($sp)
    sw $ra, 0($sp)
    bne $a0, $0, else
    add $v0, $0, $0
    addi $sp, $sp, 8
    jr $ra
else:
    andi $s0, $a0, 1
    srl $a0, $a0, 1
    jal Count
    add $v0, $v0, $s0
    lw $ra, 0($sp)
    lw $s0, 4($sp)
    addi $sp, $sp, 8
    jr $ra
```
2.1 If register \$a0 stores the integer 11, what is the value in \$v0 after the execution of the above MIPS code?

2.2 In C-style pseudo-code, describe what this MIPS code computes.

PROBLEM 3.  [30 points] Assume that, in MIPS assembly, the base address of an array \( v \) in stored in \$a0 and that the length of that array is stored in \$a1.

3.1 Regarding the consecutive words contained in \( v \) as signed integers, implement a MIPS function returning the maximum and minimum elements of \( v \) in \$v0 and \$v1, respectively.

3.2 Write a complete MIPS program testing your function. Similarly to the program \texttt{W18\_bubble\_sort.asm} seen in class, the test-array would be defined in the program itself and you can use:

\begin{center}
5, 18, 9, 1, 3, 14, 10, 4, 6, 7
\end{center}

Your program should work correctly with \texttt{SPIM}. Here are additional references about MIPS: a \texttt{tutorial}, lecture notes a MIPS Instruction cheat sheet.

PROBLEM 4.  [40 points] Consider the following definition of the combination numbers, for integers \( 0 \leq k \leq n \)
\begin{equation}
\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots1}
\end{equation}
see \url{https://en.wikipedia.org/wiki/Combination}

Recall also the recurrence relation:
\begin{equation}
\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}
\end{equation}

4.1 Implement a function computing \( \binom{n}{k} \), for \( 0 \leq k \leq n \), based on Equation (1) in MIPS assembly. To perform the division use \texttt{div}, see multiplication and division in MIPS. Try to use a minimum number of MIPS instructions.

4.2 Implement a function computing \( \binom{n}{k} \), for \( 0 \leq k \leq n \), based on Equation (2) (together with the base-cases \( \binom{n}{0} = 1 \) and \( \binom{n}{n} = 1 \)) in MIPS assembly. Try to use a minimum number of MIPS instructions.

4.3 For each of the above functions, write a complete MIPS program testing those functions. Your program should read in the values of \( n \) and \( k \), and print \( \binom{n}{k} \).