Problem P1 [5,5,15 points]: MIPS Concepts

a) It is possible to use the logical AND operator and a bit mask to implement a “bit test” operation. For example, given an integer x: (x & 2 == 2) if and only if the second least significant bit (LSB) is set to one. Given an integer x, how would you create a “bit set” operation, to set the 2nd LSB to one?

b) How about a “bit clear” operation, to clear the 2nd LSB to zero?

c) In addition to the standard $s registers for variables and $t registers for temporary values, MIPS also has some special-purpose registers. One of these is $sp, the stack pointer. At the beginning of each iteration of the following function, what is the value of $sp? Assume the initial value is 0x43A8. Also assume that every time a function executes, it allocates eight bytes of stack space for the function to run.

```c
int factorial(int x) {
    /* What is the value here? */
    if (x == 1)
        return 1;
    else
        return x * fact(x - 1);
}
```

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Value of x</th>
<th>Value of fact(x)</th>
<th>Value of $sp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>24</td>
<td>0x43A8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Problem P2 [10,15 points]: Dot Products

Given two vectors v and w, the dot product x is equal to the sum of the products of the elements:

\[ x = v_0 w_0 + v_1 w_1 + \ldots + v_{n-1} w_{n-1} \]

a) Write a C function called `dot_product` to calculate the dot product of two vectors. Let the function take three parameters: integer arrays v and w, and a size n. Have v and w be of type pointer to integer, and n be of type integer.

b) Write the same function in MIPS assembly. Use the mult pseudo-instruction to perform the multiplication (assume 16-bit numbers). Follow all MIPS function call standards. Use real instructions (from the book). Assume that v is in register $a0, w is in $a1, and n is in $a2. Return the result in $v0.

Problem P3 [15 points]: Control Instructions

The following code counts the number of occurrences of the bit sequence “10” (one zero) in x:

```c
int b = 0, x;
while (x != 0) {
    b += ((x & 03) == 2);
    x >>= 1;
}
```

Write commented MIPS code that computes this number. Assume b is stored in $s0 and x is stored in $s1.