Consider the following code:

```c
for (i = 0; i < 500; i++)
    e[i] = (a[i] + b[i]) * (c[i] - d[i]);
```

Assume that the processor has a maximum vector length of 64 and the startup overheads of the load/store unit is 15 cycles, the multiply unit is 8 cycles, and the add/subtract unit is 5 cycles.

a. First, strip mine the C source code above so that each inner loop iterates for at most 64 times.

b. Convert the strip-mined C loop into VMIPS assembly code. Assume that Ra, Rb, Rc, Rd, and Re contain the starting addresses of the arrays a, b, c, d, and e, respectively. Further assume all vector and integer VMIPS registers are available for use in the loop.

c. Assuming chaining and a single vector memory unit, how many chimes are required for each iteration of the loop containing the vector operations?

d. Again assuming chaining and a single vector memory unit, how many clock cycles are required for each e[i] result on average, including startup overhead? You can assume the scalar MIPS instructions can be overlapped with the vector operations.

The assignment is due at the beginning of class on April 19.