1. Define Amdahl’s Law. Using Amdahl’s Law, determine the overall speedup for an application if we can increase the performance of 60% of the application by a factor of 2 and another 20% of the application by a factor of 10.

2. Draw a diagram highlighting the steps needed to perform hardware/software partitioning. **Very briefly** define and highlight the importance of each step.

3. Convert the following C-like code to a high-level state machine using the template based conversion approach.

   ```
   while(1)
   {
       while(!go);
       done = 0;
       diff_avg=0;
       sum=0;
       prev = 0;
       i=0;
       while ( i < 256 )
       {
           sum = sum + (prev-A[i])/2;
           if( prev >= A[i] ) prev = A[i];
           i = i + 1;
       }
       diff_avg = sum / 256;
       done = 1;
   }
   ```

4. Partitioning the following C code to a loosely-coupled coprocessor design. Using the profile information annotated within C code, determine which of the two innermost loops will result in the best increase in performance when partitioned to a hardware coprocessor. Partition the selected innermost loop to hardware and estimate the speedup of the partitioned design over software only execution.

   ```
   int main()                       // Total Cycles: 8193437
   {
       int n;
       int i,j,k;

       for (n = 0 ; n < LOOPS ; n++) // Total Cycles: 8186006, Execs: 1,     Iters: 1000
       {
           for(i=1;i<=SIZE;i++)       // Total Cycles: 579000, Execs: 1000,  Iters: 5
           for(j=1;j<=SIZE;j++)    // Total Cycles: 520000, Execs: 5000,  Iters: 5
               c[i][j] = 0;

           for(i=1;i<=SIZE;i++)       // Total Cycles: 7579000, Execs: 1000,  Iters: 5
           for(j=1;j<=SIZE;j++)    // Total Cycles: 7520000, Execs: 5000,  Iters: 5
           for(k=1;k<=SIZE;k++) // Total Cycles: 7225000, Execs: 25000, Iters: 5
               c[i][j] += a[i][k] * b[k][j];
       }
       return 0;
   }
   ```