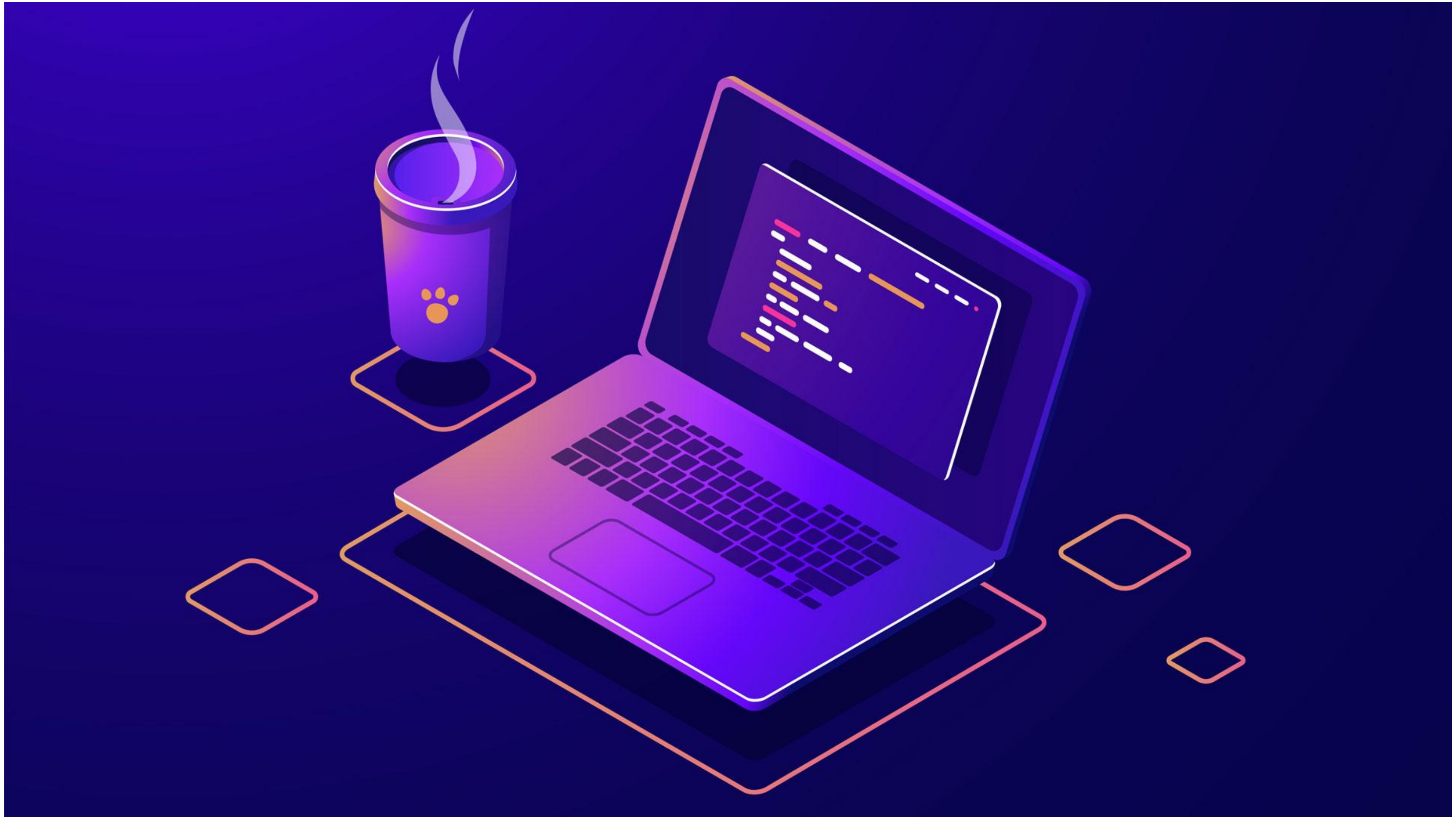
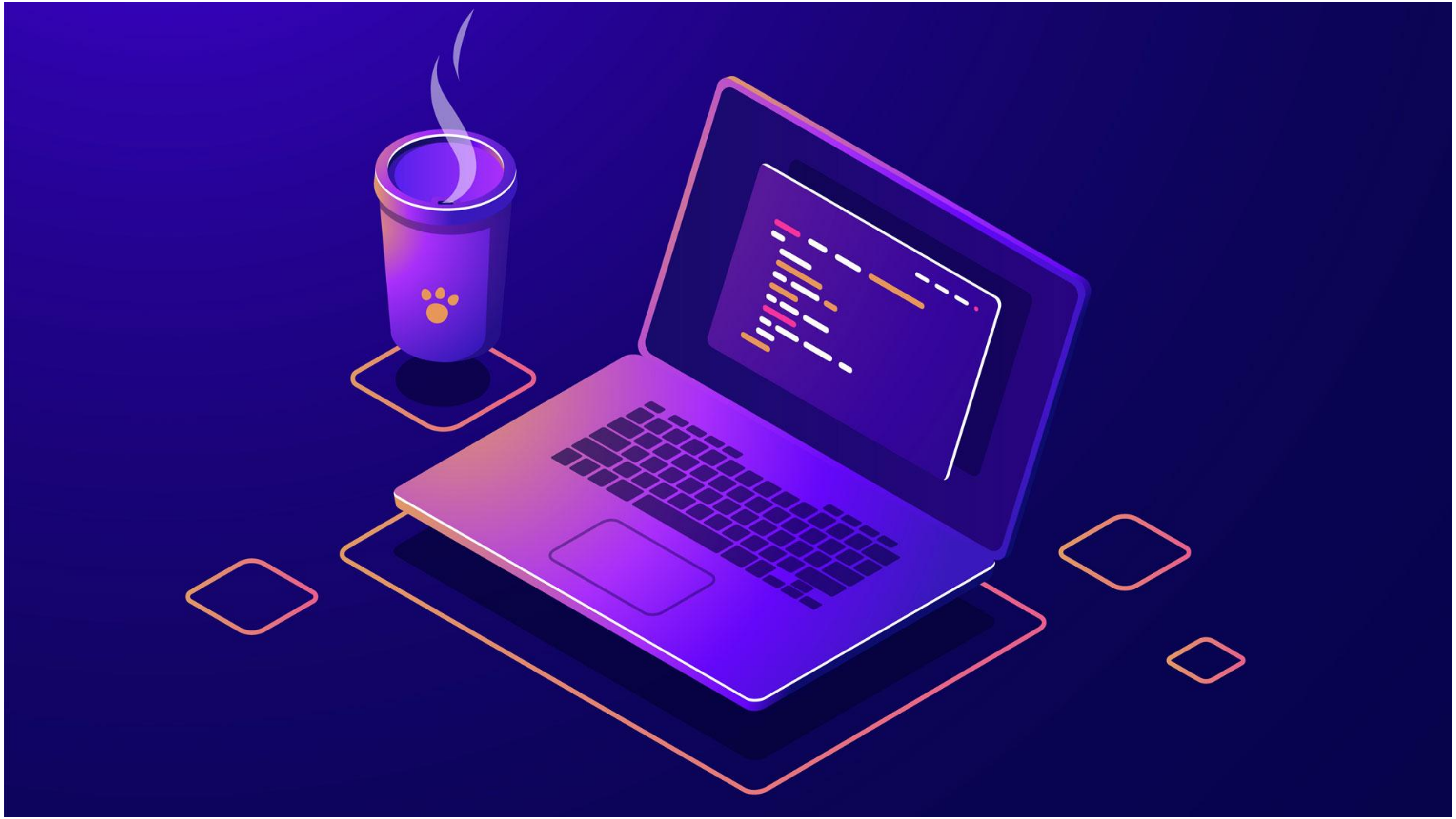
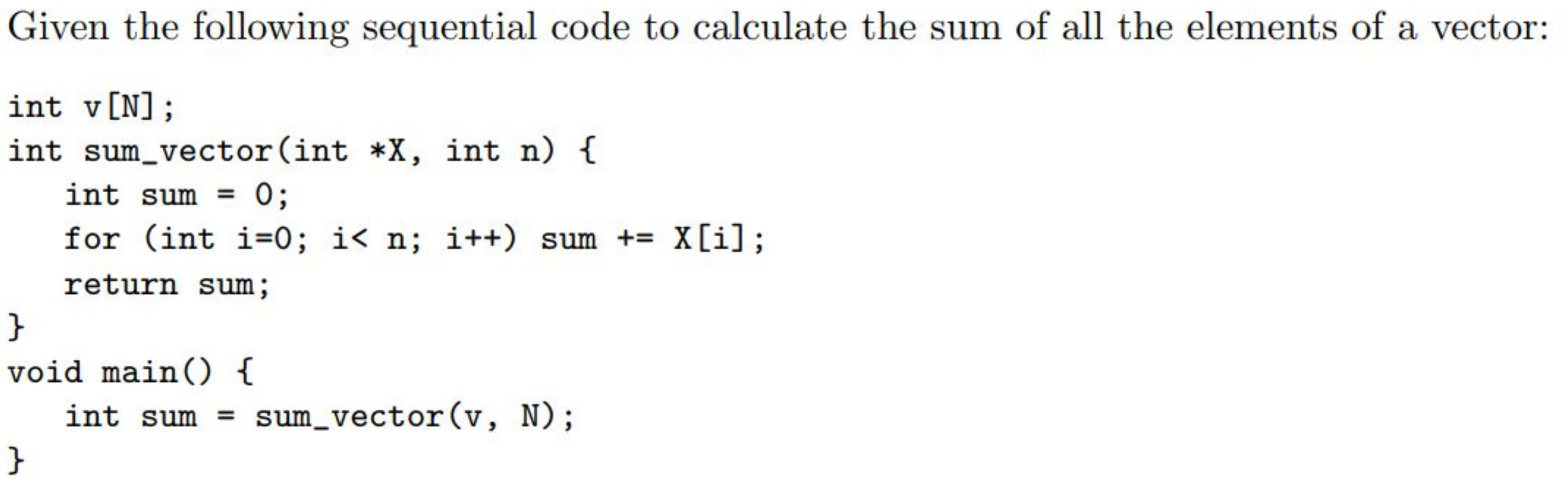
OpenMP Problems



Problem 6



Problem 6



Given the following sequential code to calculate the sum of all the elements of a vector:

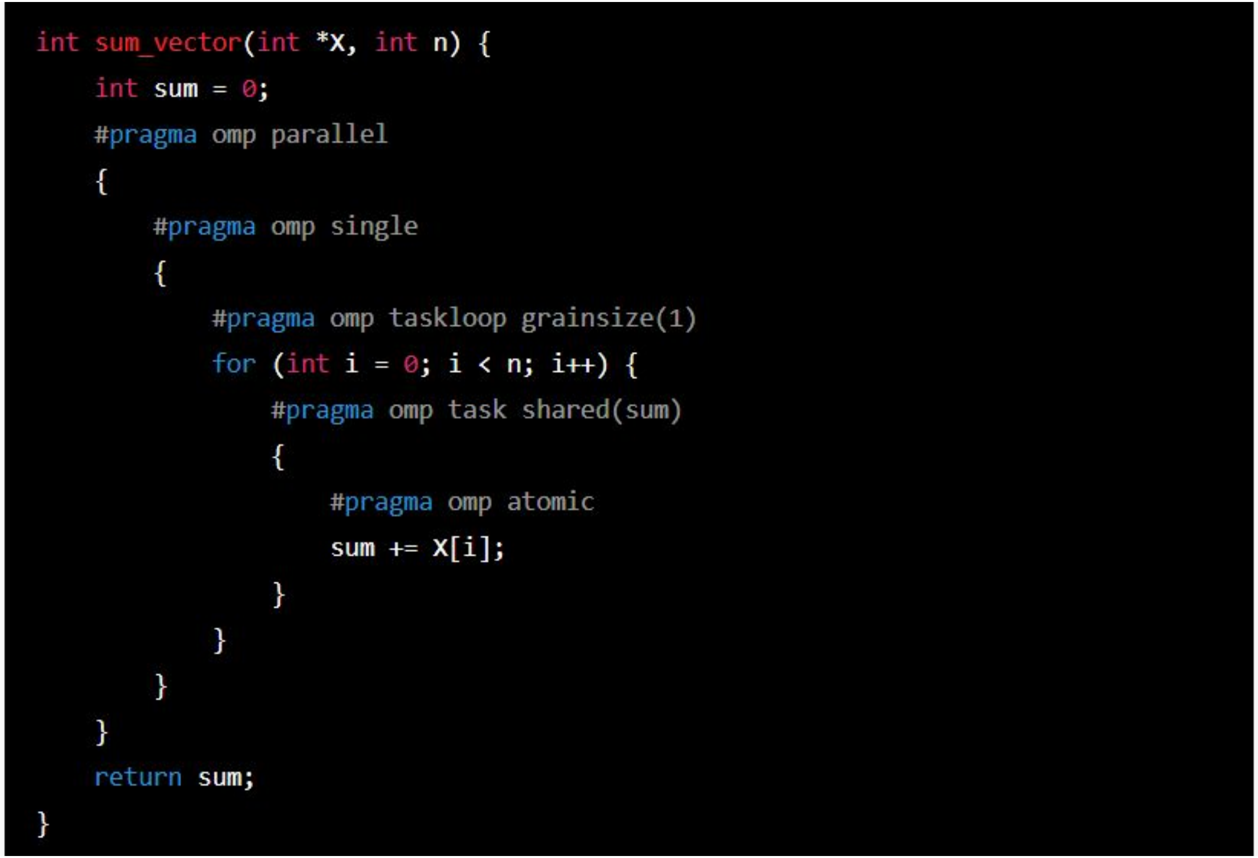
```
int v[N];
int sum_vector(int *X, int n) {
   int sum = 0;
   for (int i=0; i< n; i++) sum += X[i];
   return sum;
}
void main() {
   int sum = sum_vector(v, N);
}
```

Problem 6

a) Write a parallel version in OpenMP implementing an iterative task decomposition parallelisation strategy making use of the OpenMP tasking model, with taskloop.

b) Write a new parallel version in OpenMP of the previous sum_vector function (now called recursive_sum_vector) that implements a divide and conquer recursive strategy over vector v.

Problem 6 - Point A - Solution



```
int sum_vector(int *X, int n) {
    int sum = 0;
    #pragma omp parallel
    {
        #pragma omp single
        {
            #pragma omp taskloop grainsize(1)
            for (int i = 0; i < n; i++) {
                #pragma omp task shared(sum)
                {
                    #pragma omp atomic
                    sum += X[i];
                }
            }
        }
    }
    return sum;
}
```