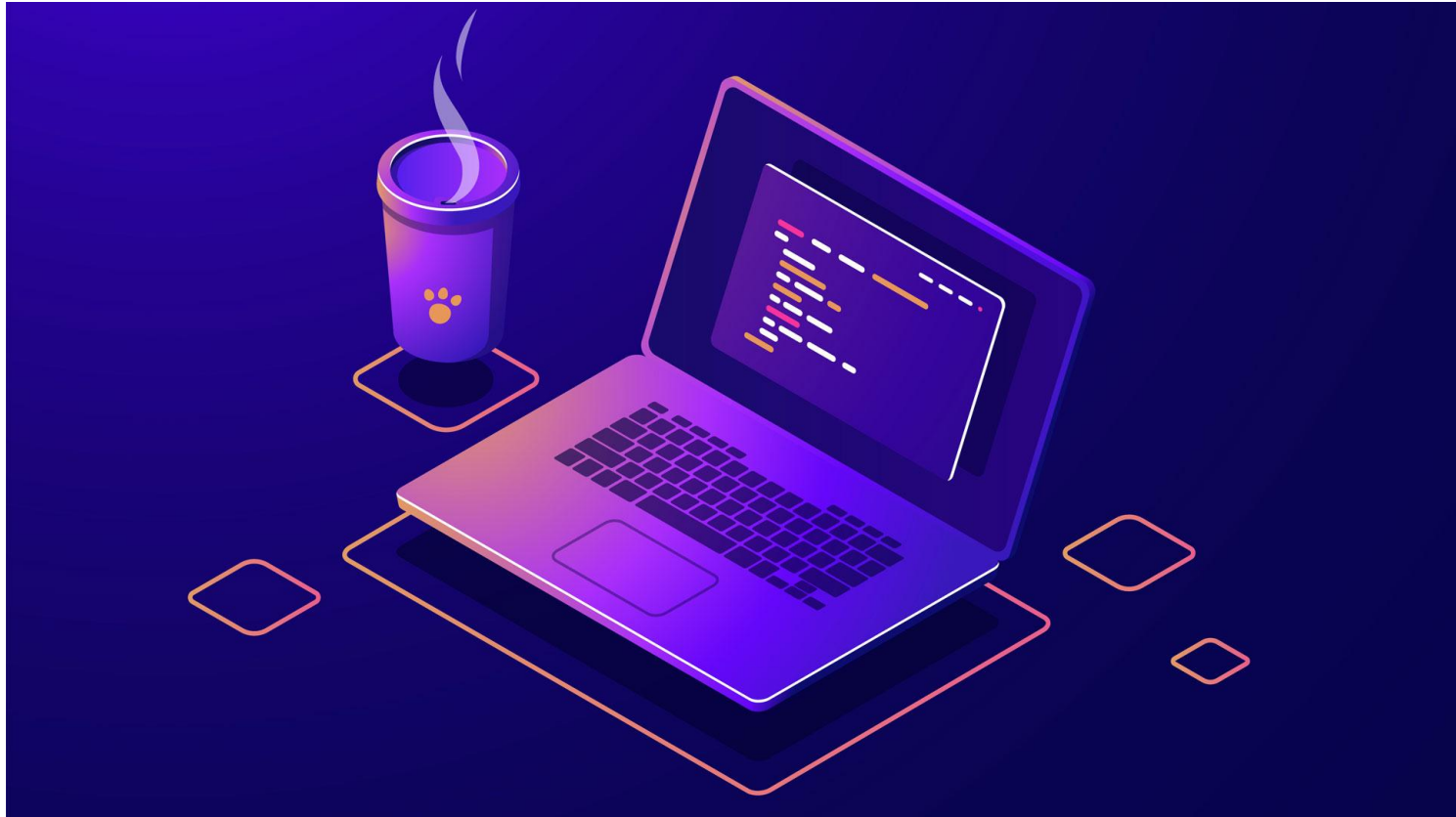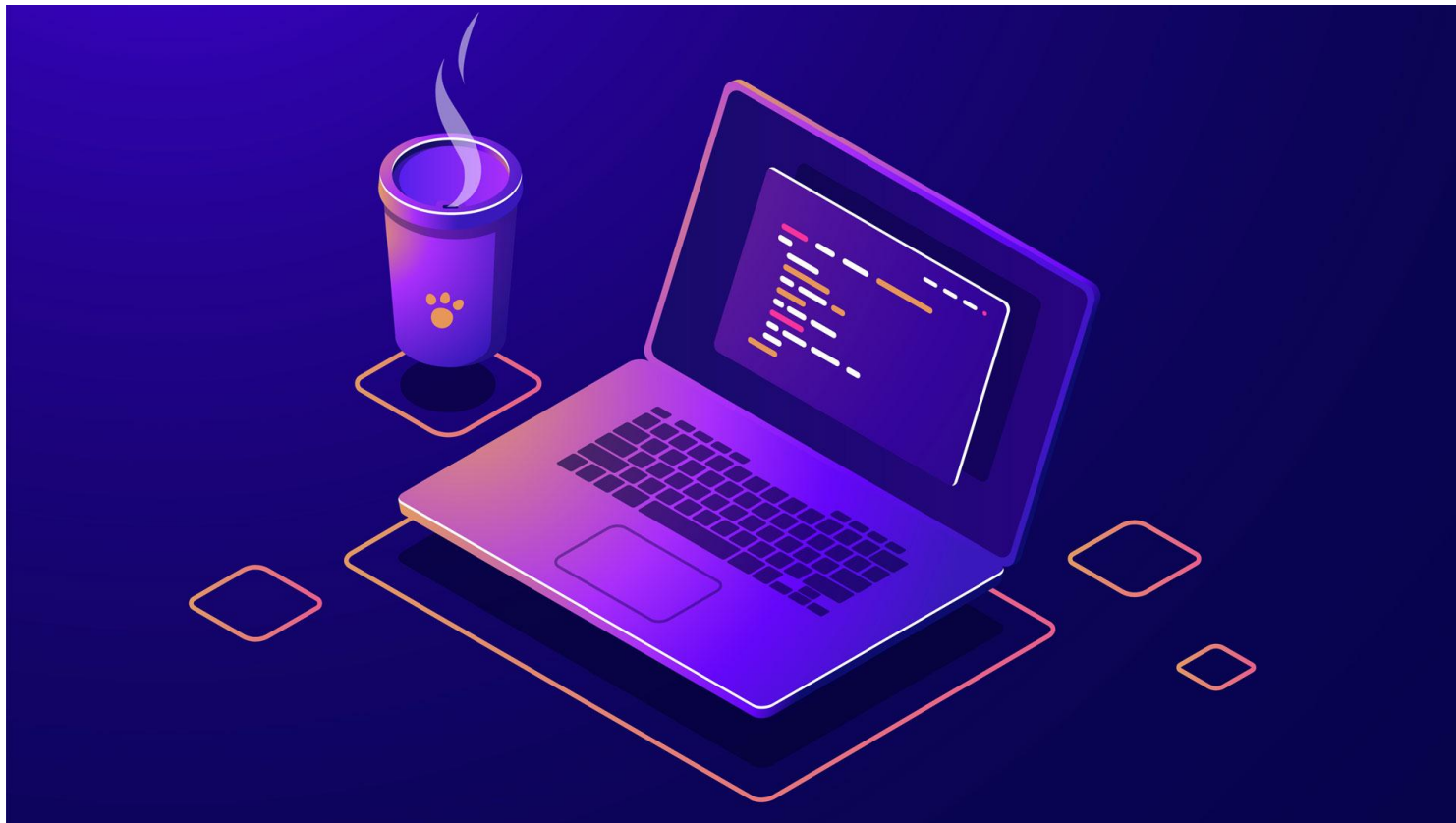# OpenMP Problems

Problem 3

Problem 3

Parallelise the following sequential code using task dependencies in OpenMP, following a producer-consumer execution model. Producer and consumer code should be in two different tasks.

```c
float sample[INPUT_SIZE+TAP1];
float coeff1[TAP1], coeff2[TAP2];
float data_out[INPUT_SIZE], final[INPUT_SIZE];

void main() {
    float sum;
    for (int i=0; i<INPUT_SIZE; i++) {
        // Producer: Finite Impulse Response (FIR) filter
        sum=0.0;
        for (int j=0; j<TAP1; j++)
            sum += sample[i+j] * coeff1[j];
        data_out[i] = sum;

        // Consumer: apply correction function
        for (int j=0; j<TAP2; j++)
            final[i] += correction(data_out[i], coeff2[j]);
    }
}
```