chan bodies[1:PR](int worker; point pos[*], vel[*]);
chan forces[1:PR](point force[*]);

process Worker[w = 1 to PR] {
    int blockSize = size of my block of bodies;
    int tempSize = maximum number of other bodies in messages;
    point p[1:blockSize], v[1:blockSize], f[1:blockSize];
    point tp[1:tempSize], tv[1:tempSize], tf[1:tempSize];
    double m[1:n];
    declarations of other local variables;
    initialize all local variables;
    for [time = start to finish by DT] {
        # send my bodies to lower numbered workers
        for [i = 1 to w-1]
            send bodies[i](w, p[*], v[*]);
        calculate f for my block of bodies;
        # receive bodies from and send forces back to
        # higher numbered workers
        for [i = w+1 to PR] {
            # get bodies from others
            receive bodies[w](other, tp[*], tv[*]);
            calculate forces between my block and other block;
            send forces[other](tf[*]);
        }
        # get forces from lower numbered workers
        for [i = 1 to w-1] {
            receive forces[w](tf[*]);
            add forces in tf to those in f;
        }
        update p and v for my bodies;
        re-initialize f to zeros;
    }
}

Figure 11.13  Heartbeat program for the \( n \)-body problem.