process Worker[w = 0 to n-1] {
    double a[n], b[n], c[n];  # my row or column of each
    double temp[n];  # used to pass vectors on
    double total;  # used to compute inner product

    # receive rows of a; keep first and pass others on
    receive vector[w](a);
    for [i = w+1 to n-1] {
        receive vector[w](temp); send vector[w+1](temp);
    }

    # get columns and compute inner products
    for [j = 0 to n-1] {
        receive vector[w](b);  # get a column of b
        if (w < n-1)  # if not last worker, pass it on
            send vector[w+1](b);
        total = 0.0;
        for [k = 0 to n-1]  # compute one inner product
            total += a[k] * b[k];
        c[j] = total;  # put total into c
    }

    # send my row of c to next worker or coordinator
    if (w < n-1)
        send vector[w+1](c);
    else
        send result(c);

    # receive and pass on earlier rows of c
    for [i = 0 to w-1] {
        receive vector[w](temp);
        if (w < n-1)
            send vector[w+1](temp);
        else
            send result(temp);
    }
}

Figure 9.6 (b)  Matrix multiplication pipeline: Worker processes.