subroutine jacobi()  
dclarations of common, shared, and private variables  
initialize grid and new in parallel (see text)  
maxdiff = 0.0 ! initialize in main thread  

c start worker threads; each executes the main loop  
!$omp parallel  
!$omp& shared(n,maxiters,grid,new,maxdiff)  
!$omp& private(i,j,Iters,tempdiff)  
      do iters = 1,maxiters,2  
        !$omp do ! divide up the iterations of the outer loop  
        do j = 2,n-1  
          do i = 2,n-1  
            new(i,j) = (grid(i-1,j) + grid(i+1,j) +  
                         grid(i,j-1) + grid(i,j+1)) * 0.25  
          enddo  
        enddo  
        !$omp end do ! implicit barrier  
      enddo  
      !$omp do ! divide up the iterations of the outer loop  
      do j = 2,n-1  
        do i = 2,n-1  
          grid(i,j) = (new(i-1,j) + new(i+1,j) +  
                        new(i,j-1) + new(i,j+1)) * 0.25  
        enddo  
      enddo  
      !$omp end do ! implicit barrier  
    enddo ! end of main computational loop  

c compute maximum difference into a reduction variable  
!$omp do ! divide up the iterations of the outer loop  
!$omp& reduction(max: maxdiff) ! use a reduction variable  
      do j = 2,n-1  
        do i = 2,n-1  
          tempdiff = abs(grid(i,j)-new(i,j))  
          maxdiff = max(maxdiff,tempdiff) ! atomic update  
        enddo  
      enddo  
      !$omp end do ! implicit barrier  
!$omp end parallel ! end of parallel section  
return  
end  

Figure 12.4 Parallel Jacobi iteration using OpenMP.