do j = 2, N-1

```
v(i, j) = v(i, j) - dt * ( (u(i, j) * (v(i+1, j) - v(i-1, j))) / (2*dx) 
(v(i, j) * (v(i, j+1) - v(i, j-1))) / (2*dy)
          end do
      end do
    end subroutine compute_velocity
    ! Function to solve for pressure (simplified Poisson equation solver)
    subroutine update_pressure(p, dx, dy)
8
      real, dimension(:,:), intent(inout) :: p
      real, intent(in) :: dx, dy
      integer :: i, j
      ! Simple pressure Poisson equation (Jacobi iteration)
      do i = 2, N-1
          do j = 2, N-1
               p(i, j) = 0.25 * (p(i+1, j) + p(i-1, j) + p(i, j+1) + p(i, j-1))
          end do
      end do
    end subroutine update_pressure
19
 end program lid_driven_cavity
23
26
30
```