```
program per
use ifport
implicit none

integer, dimension(:,:),allocatable :: a
integer     :: x,y,ca,i,L,Lx,Ly
integer :: b1,b2,b3,b4,w
real :: po

open(10,file="data.dat")
allocate(a(400,400))

do L=10,400,10
    Lx=L
    Ly=L
    po=0.0
    do i=1,200
        b1=1
        b2=2
        b3=3
        b4=4
        ca=0
        a=0
        do while(b1.ne.b2 .and. b2.ne.b3 .and. b3.ne.b4)
            do
                x=nint(rand()*Lx)
                y=nint(rand()*Ly)
                if (x<1)  x=1
                if (y<1)  y=1
                if (x>Lx)  x=Lx
                if (y>Ly)  y=Ly
                if (a(x,y)==0) then
                    a(x,y) = ca
                    ca=ca+1
                    exit
                end if
            end do
            w=change(x,y)
        end do
        po = ca*1.0/(Lx*Ly*1.0)+po
    end do ! Loop for i
    print *,L, po/200.0
    write(10,*),L,po/200.0
end do ! main loop for L
close(10)

contains
recursive function change(x,y)
    integer :: change
    integer, intent(in) :: x,y
    integer :: w
        if (x<Lx) then
            if (a(x+1,y).ne.0 .and. a(x+1,y).ne.a(x,y)) then
                if (x+1==Lx) b1=a(x,y)
                a(x+1,y)=a(x,y)
                w=change(x+1,y)
            end if
        end if
        if (x>1) then
            if (a(x-1,y).ne.0 .and. a(x-1,y).ne.a(x,y)) then
                if (x-1==1) b2=a(x,y)
                a(x-1,y)=a(x,y)
                w=change(x-1,y)
            end if
        end if
        if (y<Ly) then
            if (a(x,y+1).ne.0 .and. a(x,y+1).ne.a(x,y)) then
                if (y+1==Ly) b3=a(x,y)
                a(x,y+1)=a(x,y)
                w=change(x,y+1)
            end if
        end if
        if (y>1) then
            if (a(x,y-1).ne.0 .and. a(x,y-1).ne.a(x,y)) then
                if (y-1==1) b4=a(x,y)
                a(x,y-1)=a(x,y)
                w=change(x,y-1)
            end if
        end if
        change=0
end function change

end program per
```