do j=1,n  
    p=p+data(j)**2  
end do  

pm=p/n  
wk1(1)=data(1)  
wk2(n-1)=data(n)  
do j=2,n-1  
    wk1(j)=data(j)  
    wk2(j-1)=data(j)  
end do  
do k=1,m  
    pneum=0  
    denom=0  
do j=1,n-k  
        pneum=pneum+wk1(j)*wk2(j)  
        denom=denom+wk1(j)**2+wk2(j)**2  
end do  
cof(k)=2*pneum/denom  
pm=pm*(1-cof(k)**2)  
if (k.ne.1) then  
    do i=1,k-1  
        cof(i)=wkm(i)-cof(k)*wkm(k-i)  
    end do  
end if  
if (k.eq.m) return  
do i=1,k  
    wkm(i)=cof(i)  
end do  
do j=1,n-k-1  
    wk1(j)=wk1(j)-wkm(k)*wk2(j)  
    wk2(j)=wk2(j+1)-wkm(k)*wk1(j+1)  
end do  
end do
1. For the routine on the following page, generate the control flow graph and then show the interval derived sequence.

2. Write an irreducible program that makes sense.

3. Is the following flow graph reducible or irreducible? Compute the interval derived sequence to prove your statement.

4. Design an efficient algorithm to construct the dominance tree of a reducible flow graph.

5. Design an algorithm to identify IF statements in a control flow graph. Your algorithm should deal properly with all sort of branching situations and with nested IF statements. Here IF statements means the compound construct containing both the IF condition and all the statements whose execution is controlled by the condition.