*// Sonu Sharma, Roll No :EXTC- B-630*

*//A). Linearity Property*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:1:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

clc

disp("To Verify : DFT[a1\*x1(n) + a2\*x2(n)] == a1\*X(k) + a2\*X(k)")

x1 = input("Enter a sequence x1(n)=")

x2 = input("Enter another sequence x2(n)=")

a1 = input("Enter a scalar a1 =")

a2 = input("Enter another scalar a2 =")

x = a1\*x1 + a2\*x2

disp(x,"x(n) = a1\*x1(n) + a2\*x2(n) = ")

LHS = dft\_s(x)

disp(LHS,"LHS = ")

X1 = dft\_s(x1)

disp(X1,"X1(k) = ")

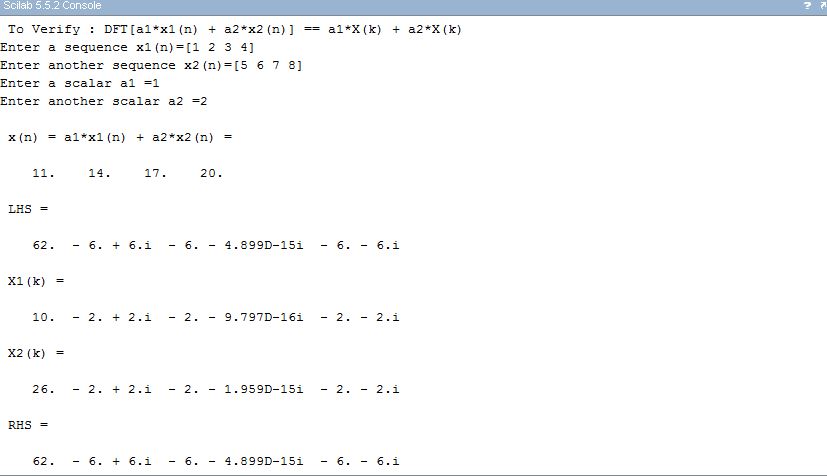
X2 = dft\_s(x2)

disp(X2,"X2(k) = ")

RHS = a1\*X1 + a2\*X2

disp(RHS, "RHS = ")

OUTPUT



*// Sonu Sharma, Roll No :EXTC- B-630*

*//B.) Time Reversal property*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:1:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

clc

disp("To Verify : DFT[x(-n)] == X(-k)")

x = input("Enter a sequence x(n)=")

X = dft\_s(x)

disp(X,"X(k) = ")

N = length(x)

n = 0:N-1

index = pmodulo(N-n,N)+1

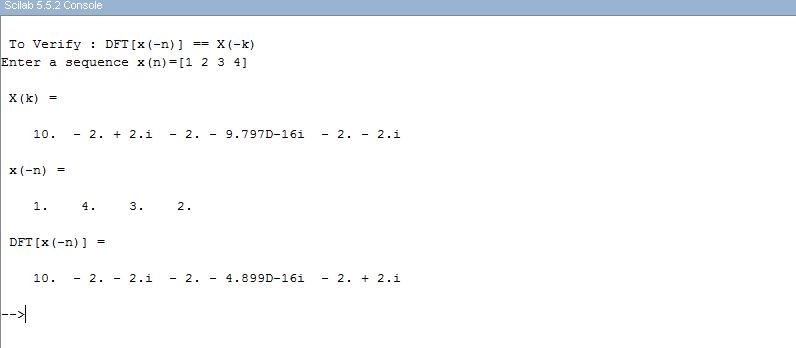
x1 = x(index)

disp(x1,"x(-n) = ")

X1 = dft\_s(x1)

disp(X1,"DFT[x(-n)] = ")

OUTPUT



*// Sonu Sharma, Roll No :EXTC- B-630*

*//C.) Time Shifting property*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:1:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

clc

disp("To Verify : DFT[x(n-l)] == X(k)\*exp(-i\*2\*pi\*l\*k/N)")

x = input("Enter a sequence x(n)=")

l = input("Enter time to be shifted, l = ")

N = length(x)

n = 0:N-1

k = 0:N-1

index = pmodulo(n-l,N)+1

x1 = x(index)

disp(x1,"x(n-l) = ")

LHS = dft\_s(x1)

disp(LHS,"LHS = DFT[x(n-l)] = ")

X = dft\_s(x)

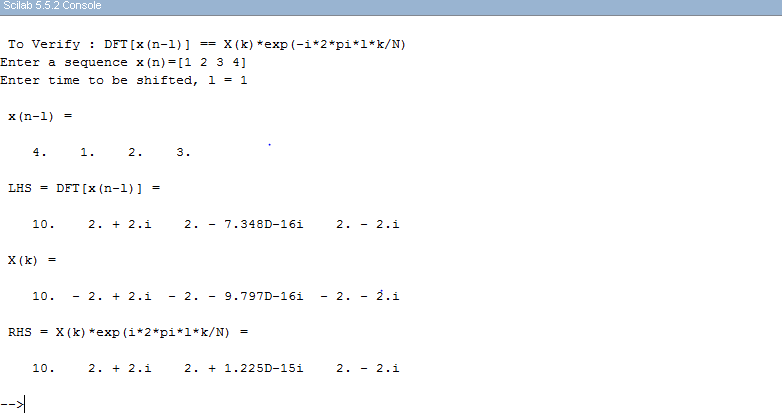
disp(X,"X(k) = ")

e = exp(-1\*%i\*2\*%pi\*k\*l/N)

RHS = X.\*e

disp(RHS,"RHS = X(k)\*exp(i\*2\*pi\*l\*k/N) = ")

OUTPUT



*// Sonu Sharma, Roll No :EXTC- B-630*

*//D.) Frequency Shifting property*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:1:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

clc

disp("To Verify : DFT[exp(-i\*2\*pi\*n\*l/N)\*x(n)] == X(k-l)")

x = input("Enter a sequence x(n)=")

l = input("Enter frequency to be shifted, l = ")

N = length(x)

n = 0:N-1

k = 0:N-1

e = exp(1\*%i\*2\*%pi\*k\*l/N)

x1 = x.\*e

LHS = dft\_s(x1)

disp(LHS,"LHS = DFT[exp(i\*2\*pi\*n\*l/N)\*x(n)] = ")

X = dft\_s(x)

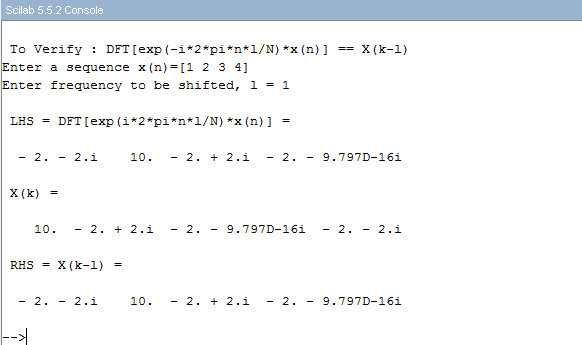
disp(X,"X(k) = ")

index = pmodulo(n-l,N)+1

X1 = X(index)

disp(X1,"RHS = X(k-l) = ")

OUTPUT



*// Sonu Sharma, Roll No :EXTC- B-630*

*//E.) Multiplication Theorem*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

function [**a**]=idft\_s(**A**)

Na = length(**A**)

na = 0:Na-1

wa = exp(%i\*2\*%pi/Na)

za = na'\*na

TFa = wa.^za

**a** = (**A**\*TFa)/Na

endfunction

clc

disp("To Verify :DFT[Multiplication in time domain] == Convolution in frequency domain/N")

x1 = input("Enter a sequence x1(n)=")

x2 = input("Enter another sequence x2(n)=")

N = length(x1)

n = 0:N-1

k = 0:N-1

x = x1.\*x2

disp(x,"Multiplication in time domain = ")

X = dft\_s(x)

disp(X,"LHS = DFT[Multiplication in time domain] = ")

X1 = dft\_s(x1)

disp(X1,"X1(k) = ")

X2 = dft\_s(x2)

disp(X2, "X2(k) = ")

*//conv of X1 and X2*

X11 = dft\_s(X1)

X22 = dft\_s(X2)

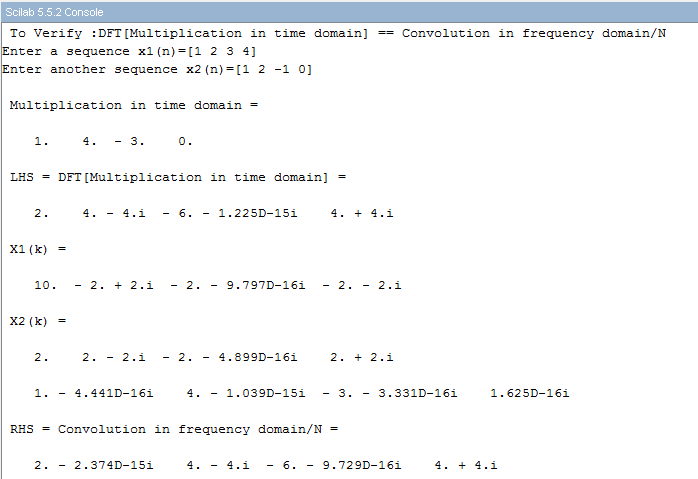
XX = X11.\*X22

RHS = idft\_s(XX)/N

disp(idft\_s(RHS))

disp(RHS,"RHS = Convolution in frequency domain/N =")

OUTPUT



*// Sonu Sharma, Roll No :EXTC- B-630*

*//F). Periodicity property*

clear

function [**X**]=dft\_s(**x**)

N = length(**x**)

n = 0:1:N-1

w = exp(-1\*%i\*2\*%pi/N)

z = n'\*n

TF = w.^z

**X** = **x**\*TF

endfunction

clc

x = input("Enter a sequence x(n)=")

X = dft\_s(x)

disp(X,"X =")

N = length(x)

n = 0:1:N-1

x2 = x(pmodulo(n-N,N)+1)

disp(x2, "after shifting x, N times x2 =")

X2 = dft\_s(x2)

disp(X2, "X2 = ")

if X == X2 then

disp("Periodic in nature")

else disp("Not periodic")

end

OUTPUT

