**ME (Embedded Systems)**

**DIGITAL SIGNAL PROCESSING**

**Lab Assignment 1**

**Name : SRILATHA P**

**Reg. No. : 221039001**

**1. Write a Matlab function “linearconv.m” to perform linear convolution of two sequences. Use**

**this function to verify following properties of Linear Convolution.**

**a. Commutative Property**

**b. Associative Property**

**c. Distributive Property**

**Code :**

**linearconv.m**

function [output]= linearconv(a,b)

l1=length(a);

l2=length(b);

lo=l1+l2-1;

output=zeros(1,lo); %initialise output vector to m+n-1 zeros

for n=1:lo

sum=0;

for k=1:l1

if(n-k+1>0 && n-k+1<=l2)

sum=sum+a(k)\*b(n-k+1);

end

end

output(n)=sum;

end

**verify.m**

clc;

%test commutative property

% c=a\*b d=b\*a

a=[1,2,3,4,5];

b=[1,2,3,4];

c=linearconv(a,b);

d=linearconv(b,a);

disp('LHS of commutative property');

disp(c);

disp('RHS of commutative property');

disp(d);

disp('COMMUTATIVE PROPERTY VERIFIED.');

disp(' ')

%test associative property

a=[1,2,3,4,5];

b=[1,2,3];

c=[2,3];

%find (a\*b)\*c where d=a\*b

d=linearconv(a,b);

e=linearconv(d,c);

%find a\*(b\*c) where f=b\*c

f=linearconv(b,c);

g=linearconv(a,f);

disp('LHS of associative property');

disp(e);

disp('RHS of associative property');

disp(g);

disp('ASSOCIATIVE PROPERTY VERIFIED.');

disp(' ')

%test distributive property

a=[1,2,3];

x1=[2,3,4,5];

x2=[8,9];

com\_length=max([length(a),length(x1),length(x2)]);

a=[a zeros(1,com\_length-length(a))];

x1=[x1 zeros(1,com\_length-length(x1))];

x2=[x2 zeros(1,com\_length-length(x2))];

lhs=linearconv(a,x1)+linearconv(a,x2);

rhs=linearconv(a,x1+x2);

disp('LHS of distributive property');

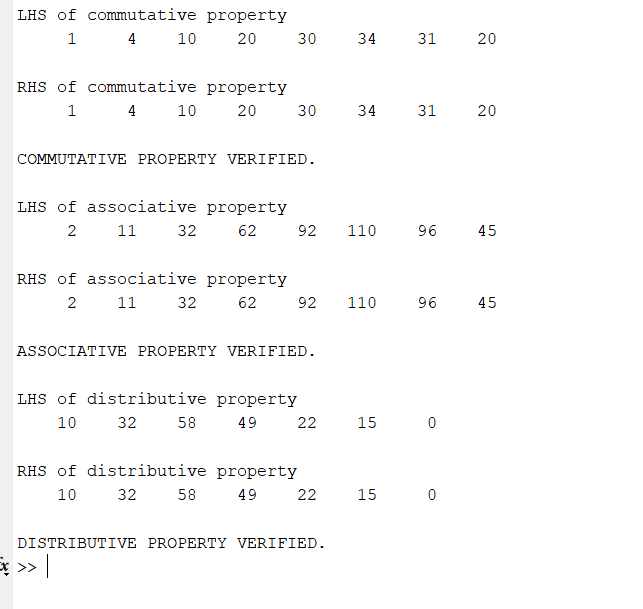
disp(lhs);

disp('RHS of distributive property');

disp(rhs);

disp('DISTRIBUTIVE PROPERTY VERIFIED.');

**Output**:



**2. Write a Matlab function “dft.m” to find DFT of a sequence. Use this function to find**

**the convolution of two sequences**.

**Code :**

**dft.m**

function [output]=dft(x,N)

%initialise ouput vector to N zeros

output= zeros(1,N);

%constant factor in dft equation pre-computed

b= -1i\*2\*pi/N;

lx=length(x);

for k=1:N

output(k)=0;

for n=0:lx-1

output(k)=output(k)+x(n+1)\*exp(b\*(k-1)\*n);

end

end

**verifydft.m**

%find convolution using dft function

clc;

N=8;

% a and b are two vectors whose convolution is to be computed

a=[1,2,3,4,5];

b=[1,2,3,4];

%convolution using user defined function

c=linearconv(a,b);

Ak=dft(a,N);

Bk=dft(b,N);

Ck=Ak.\*Bk;

%convolution by idft of pointwise multiplied DFT sequences

conSeq=(1/N).\*(conj(dft(conj(Ck),N)));

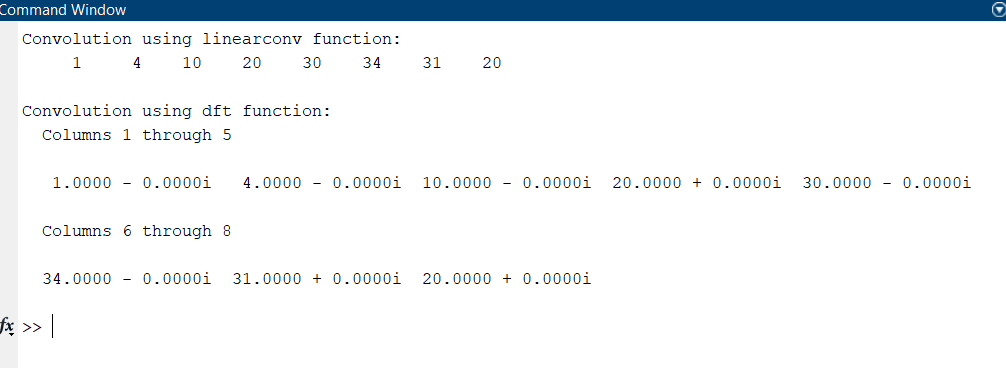
disp('Convolution using linearconv function:');

disp(c);

disp('Convolution using dft function:');

disp(conSeq);

**Output**:



**3. Write a Matlab program to find DFT of a sequence using DIF-FFT.**

**Code :**

**dif\_fft.m**

%dft using DIF-FFT

clc;

x =[1,-1,-1,-1,1,1,1,-1];

N =8;

Xk=fft(x);

p=log2(N);

Half=N/2;

for stage=1:p %process input vector stage by stage and write back to it

for index=0:(N/(2^(stage-1))):(N-1)

for n=0:(Half-1)

pos=n+index+1;

pow=(2^(stage-1))\*n;

w=exp((-1i)\*(2\*pi)\*pow/N); %twiddle factors

a=x(pos)+x(pos+Half); %butterfly addition

b=(x(pos)-x(pos+Half)).\*w; %butterfly subtraction

x(pos)=a;

x(pos+Half)=b;

end

end

Half=Half/2;

end

y=bitrevorder(x); %inbuilt function to bit reverse the indices

disp('FFT computed using inbuilt function');

disp(Xk);

disp('FFT computed using program');

disp(y);

**Output**:

