sampling theorem s	autooth square wave	<u>cos sin +exponen</u> - exponential sig		widdle factor	linear and circular convolution	<u>unit ramp imulse step</u>
clc:						clear all
clear all;	f=input("Enter the Frequency");	clc;	s=a*exp(-a*t);	ilc;	clc;	n=input('ENTER THE LENGTH
t=0:0.001:0.1;	t=-2:0.01:5;	clear all;		lear all;	clear all;	t=-2:1:n
fm=input("enter the frequency :");	s=square(t)	close all;		lose all:	u=[3 2 1 2]	for i=1:1:length(t)
x=sin(2*pi*fm*t);	figure;	t=0:0.001:1;		1=16:	v=[1212]	
subplot(4,1,1)	subplot(2,1,1)	f=input("Enter the Frequency");		:=randi(10,n,1)+1i.*randi(10,n,1);		if t(i)>=0
	plot(t,s)	a=input("Enter the Amplitude");			k=conv(u,v);	y(i)=1
olot(t,x);	title('square Wave');	s=a*sin(2*pi*f*t);		ic;	subplot(3,1,1);	else
title('sine wave');	xlabel('Time');	subplot(2,2,1)		vn=exp(-(j*2*pi)/n);	stem(u);	y(i)=0
klabel('time');	ylabel('Amplitude');	plot(t,s)		or a=1:1:n	subplot(3,1,2);	end
ylabel('amplitude');	,,	title('Sine Wave');	figure;	for j=1:1:n	stem(v);	end
	y=sawtooth(2*pi*f*t);	xlabel('Time');	subplot(2,1,1)	v_n(a,j)=wn.^((a-1)*(j-1));	subplot(3,1,3);	
fs=10	subplot(2,1,2)	ylabel('Amplitude');	plot(t,s)	end	stem(k);	subplot(3,1,1)
=0:(1/fs):0.1;	plot(t,y);	ytabet(Amptitude),		end	xlabel('amplitude');	stem(t,y)
n=sin(2*pi*fm*n);	title('Sawtooth Wave');			lisp('Twiddle Factor Matrix');	ylabel('time');	title("step")
subplot(4,1,2)	xlabel('Time');	s=a*cos(2*pi*f*t);	ylabel('Amplitude');	_n;	title('linear convolut	tion');
stem(n,xn);				r=v_n*x;		t=-2:1:n
itle('fs<<2fm');	ylabel('Amplitude');	subplot(2,2,2)	t	ime(1)=toc;	figure;	for i=1:1:length(t)
label('time');		plot(t,s)	sine wave (kit)	_real=real(v_n)	clc;	if t(i)==0
rlabel('amplitude');		title('Cos Wave');		_imaginary=imag(v_n)	clear all;	y(i)=1
• •		xlabel('Time');	#include <stdio.h> s</stdio.h>	ubplot(2,1,1)	n=[3 2 1 2]	else
=30		ylabel('Amplitude');		mshow(v_real)	m=[1212]	y(i)=0
=0:(1/fs):0.1;				itle("Real Part of Twiddle factor	i=length(n);	end
n=sin(2*pi*fm*n);		s=a*exp(a*t);		natrix")	t=length(m);	end
subplot(4,1,3)		subplot(2,2,3)		ubplot(2,1,2)	c=max(i,t);	cina .
tem(n,xn);		plot(t,s)		mshow(v_imaginary)	h=cconv(n,m,c);	subplot(3,1,2)
itle('fs=2fm');		title('+ve Exp Wave');		itle("Imaginary Part of Twiddle	subplot(3,1,1);	stem(t,y)
dabel('time');		xlabel('Time');		actor matrix")	stem(n);	title("impulse")
ylabel('amplitude');		ylabel('Amplitude');	1	actor matrix)	subplot(3,1,2);	title(impulse)
ytabett ampirtude),			1		stem(m);	
		s=a*exp(-a*t);	,		subplot(3.1.3):	t=-2:1:n
		subplot(2,2,4)				for i=1:1:length(t)
fs=60		plot(t,s)			stem(h);	if t(i)>=0
n=0:(1/fs):0.1;		title('-ve Exp Wave');			xlabel('amplitude');	y(i)=t(i)
xn=sin(2*pi*fm*n);		xlabel('Time');			ylabel('time');	else
subplot(4,1,4)		ylabel('Amplitude');			title('circular convol	
stem(n,xn);		ytabet(Amptitude),				end
title('fs>>2fm');						end
xlabel('time');						
ylabel('amplitude');						subplot(3,1,3)
ylabel('amplitude');				ovelan add method		stem(t.v)
•	real time fft without i	buffer (kit)	real time fft with huffer (kit)	ovelap add method	ov	stem(t.v)
ylabel('amplitude'); ur convolution (kit)	real time fft without b	buffer (kit)	real time fft with buffer (kit)	ovelap add method	<u>ov</u>	
r convolution (kit)				ovelap add method	<u>ov</u>	stem(t.v)
r convolution (kit) finclude <stdio.h></stdio.h>	#include <stdio.h< td=""><td>></td><td>#include "L138_LCDK_aic3106_init.h"</td><td></td><td><u>ov</u></td><td>stem(t,y) <u>relap save method</u>^{title("ramp")}</td></stdio.h<>	>	#include "L138_LCDK_aic3106_init.h"		<u>ov</u>	stem(t,y) <u>relap save method</u> ^{title("ramp")}
r convolution (kit) #include <stdio.h> #define LENGHT1 6</stdio.h>	#include <stdio.h #include<math.h< td=""><td>> ></td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512</td><td>clear all;</td><td><u>ov</u></td><td>stem(t,y) <u>relap save method</u>title("ramp") close all</td></math.h<></stdio.h 	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512	clear all;	<u>ov</u>	stem(t,y) <u>relap save method</u> title("ramp") close all
<u>r convolution (kit)</u> #include <stdio.h> #define LENGHT1 6 #define LENGHT2 4</stdio.h>	#include <stdio.h #include<math.h #include<comple< td=""><td>> ></td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE];</td><td>clear all; clc;</td><td><u>ov</u></td><td>stem(t,y) <u>relap save method</u> title("ramp") close all x=[3-101320121];</td></comple<></math.h </stdio.h 	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE];	clear all; clc;	<u>ov</u>	stem(t,y) <u>relap save method</u> title("ramp") close all x=[3-101320121];
r convolution (kit) #include <stdio.h> #define LENGHT1 6 #define LENGHT2 4 int x!2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0</stdio.h>	#include <stdio.h #include<math.h #include<comple ,0,0); float mag[8];</comple </math.h </stdio.h 	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE]; int16_t buf_ptr = 0;	clear all; clc; close all;	<u>99</u>	stem(t,y) <u>relap save method</u> 'title("ramp") close all x=[3-101320121]; h=[11];
r convolution (kit) Hinclude <stdio.h> Hdefine LENGHT1 6 Hdefine LENGHT2 4 Int x[2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0</stdio.h>	#include <stdio.h #include<math.h #include<comple ,0,0); float mag[8];</comple </math.h </stdio.h 	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE];	clear all; clc; close all; x=[3-101320121]	99	stem(t,y) /elap save method title("ramp") close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3;
r convolution (kit) Iinclude <stdio.h> Idefine LENGHT1 6 Wdefine LENGHT2 4 nt x[2*LENGHT-1]=(1,2,3,4,5,6,0,0,0,0)</stdio.h>	#include <stdio.h #include<comple="" #include<math.h="" ,0,0);="" ,0,0];="" float="" int="" mag[8];="" main()<="" td=""><td>> ></td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE]; int16_t buf_ptr = 0; interrupt void interrupt4(void)</td><td>clear all; cle; close all; x=[3-101320121] h=[111]</td><td>ov</td><td>stem(t,y) <u>relap save method</u> 'title("ramp") close all x=[3-101320121]; h=[11];</td></stdio.h>	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 int32_t inbuffer [BUFSIZE]; int16_t buf_ptr = 0; interrupt void interrupt4(void)	clear all; cle; close all; x=[3-101320121] h=[111]	ov	stem(t,y) <u>relap save method</u> 'title("ramp") close all x=[3-101320121]; h=[11];
r convolution (kit) Winclude <stdio.h> Widefine LENGHT16 #define LENGHT24 int x[2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0 int h]z*LENGHT1-1]=(1,2,3,4,0,0,0,0) int y[LENGHT1-LENGHT21]</stdio.h>	#include <stdio.h #include<comple="" #include<math.h="" (int="" ,0,0);="" float="" i,n,l,k;<="" int="" mag[8];="" main()="" td=""><td>> ></td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 ini32_t inbuffer [BUFSIZE]; ini16_t_buf_ptr = 0; interrupt void interrupt4(void) (int16_t_sample_data;</td><td>clear all; clc; close all; x=[3-101320121] h=[111] L=3;</td><td>OV</td><td>stem(t,y) relap save method close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3;</td></stdio.h>	> >	#include "L138_LCDK_aic3106_init.h" #define BUFSIZE 512 ini32_t inbuffer [BUFSIZE]; ini16_t_buf_ptr = 0; interrupt void interrupt4(void) (int16_t_sample_data;	clear all; clc; close all; x=[3-101320121] h=[111] L=3;	OV	stem(t,y) relap save method close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3;
r convolution (kit) Finclude <stdio.h> rdefine LENGHT1 6 Rdefine LENGHT2 4 int x[2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0 int h[2*LENGHT1-1]=(1,2,3,4,0,0,0,0) int y[LENGHT1-LENGHT2-1];</stdio.h>	#include <stdio.h #include<comple="" #include<math.h="" (int="" ,0,0);="" char="" float="" i,n,l,k;="" mag[8];="" main!);="" sign;<="" td=""><td>> > ex.h></td><td>#include "L136_LCDK_aic3106_init.h" #ddefine BUFSIZE 512 int32_t inbuffer [BUFSIZE]; int16_t buf_ptr = 0; interrupt void interrupt4(void) { int16_t sample_data; sample_data; sample_data; input_left_sample();</td><td>clear all; clc; close all; x=[3-101320121] h=[111] L=3; M=3</td><td><u>ov</u></td><td>stem(t,y) relap save method title("ramp") close all x=[3:101320121]; h=[1:1]; L=3; M=3; N=L+M-1;</td></stdio.h>	> > ex.h>	#include "L136_LCDK_aic3106_init.h" #ddefine BUFSIZE 512 int32_t inbuffer [BUFSIZE]; int16_t buf_ptr = 0; interrupt void interrupt4(void) { int16_t sample_data; sample_data; sample_data; input_left_sample();	clear all; clc; close all; x=[3-101320121] h=[111] L=3; M=3	<u>ov</u>	stem(t,y) relap save method title("ramp") close all x=[3:101320121]; h=[1:1]; L=3; M=3; N=L+M-1;
r convolution (kit) finclude <stdio.h> define LENGHT1 6 tdefine LENGHT2 4 nt x[2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0 nt h[2*LENGHT1-1]=(1,2,3,4,0,0,0,0,0 nt y[LENGHT1-1]=(1,2,3,4,0,0,0,0,0 nt y[LENGHT1-1]=(1,2,3,4,0,0,0,0,0,0 nt y[LENGHT1-1]=(1,2,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</stdio.h>	#include <stdio.h #include<comple="" #include<math.h="" (0,0);="" ,n,l,k;="" char="" double="" float="" int="" mag[8];="" main()="" pl="acos(-</td" sign;="" {=""><td>1);</td><td>#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_inbuffer [BUFSIZE]; int16_t_buf_ptr = 0; inter_upt void interrupt4/void) (int16_t_sumple_data; sample_data = input_left_sample(); inbuffer[buf_ptr] = sample_data;</td><td>clear all; clc; close all; x=[3-101320121] h=[111] L=3; M=3 N=L+M-1</td><td>ον</td><td>stem(t,y) relap save method close all x=[3-101320121]; h=[1.11]; [-3; M-3; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)];</td></stdio.h>	1);	#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_inbuffer [BUFSIZE]; int16_t_buf_ptr = 0; inter_upt void interrupt4/void) (int16_t_sumple_data; sample_data = input_left_sample(); inbuffer[buf_ptr] = sample_data;	clear all; clc; close all; x=[3-101320121] h=[111] L=3; M=3 N=L+M-1	ον	stem(t,y) relap save method close all x=[3-101320121]; h=[1.11]; [-3; M-3; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)];
convolution (kit) include sctdio.h> define LENGHT1 6 idefine LENGHT2 4 nt x(2*LENGHT2+1]=(1,2,3,4,5,6,0,0,0 nt y(LENGHT1+1)=(1,2,3,4,0,0,0,0,0 nt y(LENGHT1+LENGHT2-1); nain()	#include <stdio.h #include<comple="" #include<math.h="" 0,0);="" [,n,l,k;="" char="" complex<="" double="" fint="" float="" int="" mag[8];="" main()="" pl="accof-" sign;="" td=""><td>>> >> BX.h> 1); X[8];</td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCS2_tibut/re_fBUFSZE; init.fs_tbut_pt = 0; interrupt void interrupt/divoid) (int.fs_tsample_data; sample_data; input_fer_fbut_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/lsck_pt/l</td><td>clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1) h=(1 1 1 1) L=3; M=3 N=L+M-1 x=(x zeros(1,L))</td><td><u>ov</u></td><td>stem(t,y) relap save method title("ramp") close all ==(3 · 1 0 1 3 2 0 1 2 1); h=(1 1 1); l=3; h=3; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(0);</td></stdio.h>	>> >> BX.h> 1); X[8];	#include "L138_LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCS2_tibut/re_fBUFSZE; init.fs_tbut_pt = 0; interrupt void interrupt/divoid) (int.fs_tsample_data; sample_data; input_fer_fbut_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/lsck_pt/l	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1) h=(1 1 1 1) L=3; M=3 N=L+M-1 x=(x zeros(1,L))	<u>ov</u>	stem(t,y) relap save method title("ramp") close all ==(3 · 1 0 1 3 2 0 1 2 1); h=(1 1 1); l=3; h=3; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(0);
convolution (kit) include stdio.h> define LENGHT1 6 idefine LENGHT2 4 it x[2*LENGHT1-11-(2,2,3,4,5,6,0,0,0 it h[2*LENGHT1-1]-(1,2,3,4,0,0,0,0,0 it h[2*LENGHT1-1]-(1,2,3,4,0,0,0,0,0 it i=0,1) it i=0,1;	#include <stdio.h #include<="" #include<comple="" td=""> #include #</stdio.h>	>> >> BX.h> 1); X[8];	#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_tinbuffe (BUFSIZE); int16_t_buf_pur = 0; inter_upt_void interrupt{void} (int16_t_sumple_data; sample_data = input_left_sample(); inbuffer(buf_pur) = ample_data; buf_pur (buf_pur-1)%BUFSIZE; output_left_sample(sample_data);	clear all; clc; close all; x=[3·10 13 20 12 1] h=[11.1] L=3; M=3 N=1-M-1 x=[x zeros(1,1)] h=[h zeros(1,2)]	ον	stem(t,y) close all x=[3-101320121]; h=[111]; 1=3; M=3; M=3; N=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(x); y=[];
r convolution (kit) include stdio,h> inc	#include <stdio.h #include<comple="" #include<math.h="" 0,0);="" [,n,l,k;="" char="" complex<="" double="" fint="" float="" int="" mag[8];="" main()="" pl="accof-" sign;="" td=""><td>>> >> BX.h> 1); X[8];</td><td>#include "L138_LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCS2_tibut/re_fBUFSZE; init.fs_tbut_pt = 0; interrupt void interrupt/divoid) (int.fs_tsample_data; sample_data; input_fer_fbut_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/lsck_pt/l</td><td>clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1] h=(11 1 1] L=3; M=3 N=L+M-1 x=(x zeros(1,L)] h=(h zeros(1,L)) s=size(x)</td><td><u>ov</u></td><td>stem(t,y) relap save method title("ramp") close all x=(3 -1 0 1 3 2 0 1 2 1); h=(1 1 1); l=3; M=2; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(t); y=(); for i=1:L:s(2)-L</td></stdio.h>	>> >> BX.h> 1); X[8];	#include "L138_LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCDK_aic3106_init.h" #define BUFSZE 512 collision = LCS2_tibut/re_fBUFSZE; init.fs_tbut_pt = 0; interrupt void interrupt/divoid) (int.fs_tsample_data; sample_data; input_fer_fbut_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/l = sample_data; but_pt/lcdk_ptr_l\square_fbuf_pt/lsck_pt/l	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1] h=(11 1 1] L=3; M=3 N=L+M-1 x=(x zeros(1,L)] h=(h zeros(1,L)) s=size(x)	<u>ov</u>	stem(t,y) relap save method title("ramp") close all x=(3 -1 0 1 3 2 0 1 2 1); h=(1 1 1); l=3; M=2; N=L+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(t); y=(); for i=1:L:s(2)-L
convolution (kit) include stdio.h> define LENGHT2 4 nt (2*LENGHT1-19(2,2,3,4,5,6,0,0,0 nt hiz*LENGHT1-19(2,2,3,4,0,0,0,0,0 nt hiz*LENGHT1-LENGHT2-1); main() nt i=0,j; or(i=0,j:<(LENGHT1+LENGHT2-1);++)	#include-stdio.h #include-compling #include-comp	>> >> BX.h> 1); X[8];	#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_tinbuffer (BUFSIZE); int16_t_buf_pur = 0; inter_upt_void interrupt{void} (int16_t_sumple_data; sample_data = input_lent_sample(); inbuffer(buf_pur = 1, sample_data; buf_prt_void_pur = 1, sample_data; buf_prt_void_pur = 1, sample_data; voiput_lent_sample(sample_data); return; }	clear all; clc; close all; x=[3-10 1 3 2 0 1 2 1] h=[11.11] L=3; N=1.4M-1 x=[x zeros(1,1)] h=[h zeros(1,2)] s=size(x) y=[]	ον	stem(t,y) close all x=[3-101320121]; h=[1.11]; [-3; M=3; M=3; N=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(x); y=[]; for =1:Ls(2)-1, L=x(ii+N-1);
convolution (kit) include <stdio.h> define LENGHT2 4 nt x[2*LENGHT1-161,2,3,4,5,6,0,0,0 nt h]z*LENGHT1-161,2,3,4,5,6,0,0,0 nt h]z*LENGHT1-161,2,3,4,0,0,0,0 nt h]z*LENGHT1-LENGHT2-1]; nain() nt i=0,i; iri=0;i-(LENGHT1+LENGHT2-1);++) iri=0;</stdio.h>	#include-stdio.h #include-compl #include-compl #include-compl #include-compl #include-compl #include-compl #include-compl #include-comple #include-stdio.h #include-comple #include-c	>> >> BX.h> 1); X[8];	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 inti32_inbuffer (BUFSIZE); inti32_inbuffer (BUFSIZE); inti32_inbuffer (BUFSIZE); inti32_inbuffer (BUFSIZE); inti42_inbuffer (BUFSIZE); inbuffer(buf_ptf) = sample_data; sample_data = input_left_sample_data; buf_ptf (Buf_ptf) = Sample_data; buf_ptf (Buf_ptf) NSBUFSIZE; output_left_sample(sample_data); return; } int main(void)	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1) h=(11 1 1) L=3; M=3 N=L+M-1 x=(x zeros(1,L)) h=(h zeros(1,L)) y=() for i=1:L:s(2)-L	ου	stem(t,y) relap save method title("ramp") close all x=[3-101320121]; h=[11]; l=3; M=3; N=L+M-1; x=[zeros(I,M-1) x zeros(I,M-1)]; s=size(x); y=[]; for i=1t.l=s(2)-t x1=x(i:M-1);
convolution (kit) include stdio.h> define LENGHT1 6 idefine LENGHT2 4 int z[**LENGHT1-19(1,2,3,4,5,6,0,0,0 in hiz**LENGHT1-19(1,2,3,4,0,0,0,0,0 in hiz**LENGHT1-19(1,2,3,4,0,0,0,0,0 int i=0-i; j(LENGHT1+LENGHT2-1); inain() int i=0,j; (ii=0; ii=(ii=0;i=(ii=ij++))	#include-stdio.h #include-compling #include-compling #include-compling #include-compling #include-compling #include-compling #int iN_Lk; char sign; double Pl=acosf- double complex #int X[8]=1,2,3,4,5 N=8; #or(k=0;k=N;k+1) {X k]=0;	>> >> BX.h> 1); X[8];	#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_tinbuffe (BUFSIZE); int16_t_buf_pur = 0; inter_upt_void interrupt{void} (int16_t_sumple_data; sample_data = input_left_sample(); inbuffer(buf_pur)=sample_data; buf_prt_louf_ptr-1]%BUFSIZE; output_left_sample(sample_data); return; } int main(void) (L138_initialise_intrifs_8000_HZ_ADC_GAIN_OC	clear all; clc; close all; x=[3-1.0 1 3 2 0 1 2 1] h=[1.1 0 1 3 2 0 1 2 1] h=[3: h=3: N=1.4M-1 x=[x zeros(1,0)] h=[h zeros(1,0)] s=size(x) y=[] y=[] y=[] y=[] y=[] y=[] y=[] y=[]	<u>ov</u>	stem(t,y) close all x=[3-101320121]; h=[111]; [1-3; M-3; N=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(x); y=[]; for =1:L:s(2)-1, L=x(i 4-N-1); x2=ccon(x(1,h,N)) y=[y x2(M-n(n)];
convolution (kit) include stdio.h> define LENGHT1 6 idefine LENGHT2 4 int z[**LENGHT1-19(1,2,3,4,5,6,0,0,0 in hiz**LENGHT1-19(1,2,3,4,0,0,0,0,0 in hiz**LENGHT1-19(1,2,3,4,0,0,0,0,0 int i=0-i; j(LENGHT1+LENGHT2-1); inain() int i=0,j; (ii=0; ii=(ii=0;i=(ii=ij++))	#includestdio.h #includescompi fincludescompi fincl	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=[3 -10 1 3 2 0 1 2 1] h=[11 1 1] L=3; M=3 N=L+M-1 x=[x zeros(1,L)] h=[h zeros(1,L)] y=[] for i=1:L:s(2)-L J=2=conv(4,h,N)] z=cconv(4,h,N)]	ου	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1-M-1; x=[seros(1,M-1) x zeros(1,M-1)]; x=size(0); y=[]; for i=1t.s(2)-1, x1=x(i+N-1); x2=cconv(x1,h,N) y=[y x2(M=nd)]; end
rconvolution (kit) linclude stdio.h> idedfine LENGHT1 6 idedfine LENGHT2 4 nt (2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0 nt h)z*LENGHT1-1]=(1,2,3,4,0,0,0,0,0 nt h)z*LENGHT1-LENGHT2-1]; main() nt i=0,j; ior(i=0);(i ENGHT1+LENGHT2-1];++) ii =0; ii =0;*(i ENGHT1+LENGHT2-1];++) ii =0;*(i =0;+ ENGHT1+LENGHT2-1];++) iii =0;*(i =0;+ ENGHT1+LENGHT2-1];++) iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	#include-stdio.h #include-compling #include-compling #include-compling #include-compling #include-compling #include-compling #int iN_Lk; char sign; double Pl=acosf- double complex #int X[8]=1,2,3,4,5 N=8; #or(k=0;k=N;k+1) {X k]=0;	>	#include"L138_LCDK_aic3106_init.h" #dofine BUFSIZE 512 int32_tinbuffe (BUFSIZE); int16_t_buf_pur = 0; inter_upt_void interrupt{void} (int16_t_sumple_data; sample_data = input_left_sample(); inbuffer(buf_pur)=sample_data; buf_prt_louf_ptr-1]%BUFSIZE; output_left_sample(sample_data); return; } int main(void) (L138_initialise_intrifs_8000_HZ_ADC_GAIN_OC	clear all; clc; close all; x=(3-1.01320121) h=(1.11) Le3 N=1.4M x=(3 N=1.4M x=(3 xeros(1,1)) h=(1 xeros(1,2)) s=sizeros(2,2) s=sizeros(1,2) y=(1) y=(1 x=(1,1); y=(1 x=(1,1); y=(1,1);	<u>ov</u>	stem(t,y) close all x=[3-101320121]; h=[111]; [-3; M=3; M=3; M=1; M=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(x); y=[]; for =1:Ls(2)-1, L=x(i 4-N-1); x2=ccon(x(1,h,N)) y=[y x2(Mend)]; end subplot(211)
r convolution (kit) Hinclude <stdio.h> Idefine LENGHT1 6 #define LENGHT2 4 Int (2*LENGHT1-16(1,2,3,4,5,6,0,0,0 Int (2*LENGHT1-16(1,2,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</stdio.h>	#includestdio.h #includescompi (0,0); float mag(8); (1,0,0); int main(); (1,0,0); int main(); (1,0,0); int main(); (1,0,0); (1,0,		#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=[3 -10 1 3 2 0 1 2 1] h=[11 1 1] L=3; M=3 N=L+M-1 x=[x zeros(1,L)] h=[h zeros(1,L)] y=[] for i=1:L:s(2)-L y=[] b= x[x[x[i+1-1] zeros(1,M-1)] x2=cconv(x1,h,N) if i=-1 y=x2	ον	stem(t,y) close all x=[3 · 1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1-M-1; x=[2erox[1,M-1) x zerox[1,M-1)]; x=[2erox[1,M-1] x zerox[1,M-1]]; y=[3 · 1 0 1 3 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
rconvolution (kit) finclude <stdio.h> define LENGHT2 4 finclude <stdio.h> define LENGHT2 4 finclude <stdio.h> define LENGHT2 4 fix (2**LENGHT1-16**L2,3,4,5,6,0,0,0 fix 2**LENGHT1-16**L2,3,4,0,0,0,0,0 fix 2**LENGHT1-16**L2,3,4,0,0,0,0,0 fix 2**LENGHT1-16**L2,3,4,0,0,0,0,0 fix 2**LENGHT1-16**L2,3,4,0,0,0,0,0 fix 2**LENGHT1-16**LENGHT2-1);i++) fix 2**LENGHT1-16**LENGHT2-1);i++) fix 2**LENGHT1-16**LENGHT2-1);i++) ort =0;i+(LENGHT1-16**LENGHT2-1);i++)</stdio.h></stdio.h></stdio.h>	#include-stdio.h #include-compling #include-compling #include-compling #include-compling #include-compling #include-compling #int Na_N; #int Na		#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3-10 13 2 0 1 2 1) h=(11 1] L-3-3 M=L-M-1 x=[x zeros(1,1)] h=(h zeros(1,2)) s=size(s(1,2)) y=(1) y=(1) x=(x zeros(1,M-1)) x=(x zeros(1,		stem(t,y) close all x=[3-101320121]; h=[111]; [-3; M=3; M=3; M=1; M=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(a); y=[]; for [=1:Ls(2)-1, L=x(ii+N-1); x2=ccon(x(1,h,N)) y=[y x2(Mendi)]; end subplot(211) stem(y) xlabe(("")
convolution (kit) include <stdio.h> define LENGHT2 4 nt zi=LENGHT3 6 nt zi=LENGHT3 1=1(2,3,4,5,6,0,0,0 nt hiz=LENGHT3-1=(1,2,3,4,0,0,0,0,0 nt hiz=LENGHT1-1=(1,2,3,4,0,0,0,0,0 nt i=0,i=(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</stdio.h>	#includestdio.h #includescompl fincludescompl fincl	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1] h=(11 11] L=3; M=3 N=L+M-1 x=(x zeros(1,L)] h=(1) zeros(1,L)] for i=1:L:s(2)-L z=(zeros(1,L)) x=(zeros(1,L)) x=(zeros(stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zerox[1,M-1] x zerox[1,M-1]]; s=size(0); y=[1,1:52]-1, d=x[i+N-1]; s=d subplot(211) stem(y) ylabe("n") title("Linear Convolution using Overlap save nr
r convolution (kit) Hinclude <stdio.h> Idefine LENGHT1 6 #define LENGHT2 4 Int (2*LENGHT1-16(1,2,3,4,5,6,0,0,0 Int (2*LENGHT1-16(1,2,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0 Int (1,2,1,3,4,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0,0,0,0,0) Int (1,2,1,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</stdio.h>	#includestdio.h #includescompl fincludescompl fincl		#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1) h=11 11] L=3; M=2-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4		stem(t,y) relap save method title("ramp") close all x=[3-10.132.0121]; h=[1.11]; l=3; M=3; M=1; N=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(a); y=[]; for =1:L:s(2)-1, 1.=x(i:H-N-1); x2=cconv(x,1h,N) y=[y x2(M:end)]; end subplot(211) stem(y) xlabe(("n") title("Linear Convolution using Overlap save r y1=conv(x,h;);
rconvolution (kit) finclude <stdio.h> idefine LENGHT2 4 rt (2*LENGHT1-16*1,2,3,4,5,6,0,0,0 rt h!z*LENGHT1-11-(1,2,3,4,0,0,0,0,0 rt h!z*LENGHT1-11-(1,2,3,4,0,0,0,0,0 rt z*LENGHT1-11-(1,2,3,4,0,0,0,0,0 rt z*C,1,2,3,4,0,0,0,0,0 rt z*C,1,2,3,4,0,0,0,0 rt z*C,1,2,3,4,0,0,0,0 rt z*C,1,2,3,4,0,0,0 rt z*C,1,2,3,4,0,0,0 rt z*C,1,2,3,4,0,0 rt z*C,1,2,3,4,0,0 rt z*C,1,2,3,4,0 rt z*C,1,2,3,4 rt z*C,1,3,4 rt z*C,1,3,4 r</stdio.h>	#includestdio.h #includescompi fincludescompi fincl	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1] h=(11 11] L=3; M=3 N=L+M-1 x=(x zeros(1,L)] h=(1) zeros(1,L)] for i=1:L:s(2)-L z=(zeros(1,L)) x=(zeros(1,L)) x=(zeros(stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zerox[1,M-1] x zerox[1,M-1]]; s=size(0); y=[1,1:52]-1, d=x[i+N-1]; s=d subplot(211) stem(y) ylabe("n") title("Linear Convolution using Overlap save nr
r convolution (kit) finclude 'stdio.h> define LENGHT2 4 fidefine LENGHT2 4 fizer in zero in	#include-catdio.h #include-catdio.h #include-compliance.include-compli	1); X 8]; X 8]; X 8]; X 9]; X 9]; X 2**P *** (N); X 2**P ** (N); X 2**	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -1 0 1 3 2 0 1 2 1) h=11 11] L=3; M=2-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4		stem(t,y) relap save method title("ramp") close all x=[3-10.132.0121]; h=[1.11]; l=3; M=3; M=1; N=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(a); y=[]; for =1:L:s(2)-1, 1.=x(i:H-N-1); x2=cconv(x,1h,N) y=[y x2(M:end)]; end subplot(211) stem(y) xlabe(("n") title("Linear Convolution using Overlap save r y1=conv(x,h;);
rconvolution (kit) finclude <stdio.h> rdefine LENGHT1 6 #define LENGHT2 4 nt <2**!ZENGHT1-10-12,3,4,5,6,0,0,0 nt hiz*!LENGHT1-11-(1,2,3,4,0,0,0,0,0 nt hiz*!LENGHT1-11-(1,2,3,4,0,0,0,0,0 nt =0,1; main() nt =0,1; fior =0; =(LENGHT1+LENGHT2-1); ++) fior =0; =(LENGHT1+LENGHT2-1); ++) rior =0; =(LENGHT1+LENGHT2-1); ++) rior =0; =(LENGHT1+LENGHT2-1); ++) rintf("%din",y(ii); parceval NO=5000</stdio.h>	#includestdio.h #includescomple fincludescomple for(#=0;k=k k++)	1); X 8]; X 8]; X 8]; X 9]; X 9]; X 2**P *** (N); X 2**P ** (N); X 2**	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=(3-101320121) h=1111] L=3; M=2-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4		stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zerox[1,M-1] x zerox[1,M-1]]; s=zize(x); y=[1]; for -1, (x) x=(x) x=(x) y=[x y x (M-1) y=[x y x (M-1) x (M-1) y=[x y x (M-1) x (M-1
r convolution (kit) Hinclude <stdio.h> Idefine LENGHT1 6 #define LENGHT2 4 Int : </stdio.h>	#include-stdio.h #include-math.h #include-math	1); X 8]; X 8]; X 8]; X 9]; X 9]; X 2**P *** (N); X 2**P ** (N); X 2**	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=[3 -10 1 3 2 0 1 2 1] h=[11 1 1] L=3; M=3 N=L+M-1 x=[x zeros(1,L)] h=[h zeros(1,L)] for i=1:1::s[2)-L y=[1] y=(2) z=(zoros(x,h,N) if i=-1 y=x2 else y=[y(1:end-(M-1)) y(end-M+2) end end		stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
rconvolution (kit) iinclude <stdio.h> idefine LENGHT1 6 idefine LENGHT2 4 nt (2*LENGHT1-1]=(1,2,3,4,5,6,0,0,0 nt hiz*LENGHT1-1]=(1,2,3,4,0,0,0,0,0 nt hiz*LENGHT1-1]=(1,2,3,4,0,0,0,0,0 nt =0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=0,i ii=1,i ii=1,i </stdio.h>	#includestdio.h #includescompl fincludescompl fincl	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_I_CDK_LINE_INPUT; while(1);	clear all; clc; close all; x=[3-101320121] h=[1111] L=3; M=3 N=L+M-1 x=[2 zeros(1,L)] h=[h zeros(1,2)] s==1bc(0) for i=11.L:s(2)-L y=2 zeros(x,L,h,N) if i=-1 y=x2 else y=[y(1:end-(M-1)) y(end-M+2 end end subplot(2,1,1) stem(y) xlabel("n")	bend)+x2(1:M-1) x2(M:end)]	stem(t,y) relap save method close all x=[3-101320121]; h=[111]; [1-3; M=3; M=3; M=1-M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=size(a); y=[]; for =1:Ls(2)-1, L=x(i:H-1); x2=cconv(x,lh,N) y=[y x2(Mend)]; end subplot(211) stem(y) title("Linear Convolution using Overlap save ry y1=conv(x,lh; subplot(212); stem(y(1))
r convolution (kit) Hinclude <stdio.h> Idefine LENGHTI 6 #define LENGHTI 74 Int (2**LENGHTI-16*1,2,3,4,5,6,0,0,0 Int (2**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (2**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1=0,1); Int (</stdio.h>	#include-stdio.h #include-math.h #include-math	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=[3-10 13 20 12 1] h=[11 1] L=3; M=3 N=L+M-1 x=[x zeros(1,L)] h=[h zeros(1,L)] s=size(x) y=[] for i=1:L:s(2)-L x=[x(i+1-1) zeros(1,M-1)] x2=cconv(x1,h,N) if i=1 y=2 end subplot(x1,h) gend subplot(x1,h) stem(y)	bend)+x2(1:M-1) x2(M:end)]	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
rconvolution (kit) finclude <stdio.h> rdefine LENGHT1 6 #define LENGHT2 4 int <pre>// Property</pre></stdio.h>	#include-settil.b. #include-settil.b. #include-senath.if #include-sena	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=(3-10.13.20.12.1] h=11.11.1] L=3; M=3 N=L+M-1 x=(2.2005(1,L)] h=[hz zeros(1,L)] y=[zeros(1,L)] st=[x] for i=11.1:x(2)-1 y=12 zeros(x(1,h,N)) if i=-1 y=x2 else y=(y(1:end-(M-1)) y(end-M+2 end end subplot(2,1,1) stem(y) xlabel("n") title("Linear Convolution using (bend)+x2(1:M-1) x2(M:end)]	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
r convolution (kit) #include <stdio.h> #define LENGHT1 6 #define LENGHT2 6 #define LENGHT2 16 #define LENGHT3 16 #define LENGHT3 11 #define LENGH</stdio.h>	#includestdio.h #includescompl fincludescompl fincl	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=[3.10 13 2 0 1 2 1] h=[11.11] L=3; M=3 N=L+M-1 x=[x zeros(1,L)] h=[h zeros(1,C)] s=size(x) y=[] JB, i=[i:1::s[2)-L xl=[x[i:i+-1] zeros(1,M-1)] x2=cconv(x1,h,N) if i=-1 y=x2 else y=iy(1:end-(M-1)) y(end-M+2 end subplot(2,1,1) stem(y) xlabel("n") xlabel("	bend)+x2(1:M-1) x2(M:end)]	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
rconvolution (kit) finclude <stdio.h> rdefine LENGHT1 6 rdefine LENGHT2 4 rint {2*L**PIGHT1-II-II-II-II-II-II-II-II-II-II-II-II-II</stdio.h>	#includestdio.h #fincludescompi fincludescompi finc	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -10 13 2 0 1 2 1) h=(11 1 1) L=3; M=3 N=L+M-1 x=(x zeros(1,L)) h=(h zeros(1,P)) y=(1 zeros(1,P)) y=(1 zeros(1,M-1)) x=(x zeros(1,M-1)) y=(x zeros(1,M-1)	bend)+x2(1:M-1) x2(M:end)]	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
r convolution (kit) Hinclude <stdio.h> Idefine LENGHTI 6 #define LENGHTI 74 Int (2**LENGHTI-16*1,2,3,4,5,6,0,0,0 Int (2**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0 Int (1**LENGHTI-16*1,2,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</stdio.h>	#includestdio.h #fincludescompi fincludescompi finc	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=[3.101320121] h=[11.1] L=3; M=3 N=1-M-1 x=[x zeros[1,1] h=[h zeros[1,2]] s=size(x) y=[] 18, i=1:1::s[2]-1 x=[xi:s[2]-1 y=x2 else y=[y-[y[i:end-(M-1)] y(end-M-2 end	bend)+x2(1:M-1) x2(M:end)]	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')
rconvolution (kit) finclude <stdio.h> idefine LENGHT2 4 idefine LENGHT2 4 int (2*LENGHT1-11-1(2,2,3,4,5,6,0,0,0 int hiz*LENGHT1-11-1(2,2,3,4,0,0,0,0,0 int hiz*LENGHT1-11-1(2,2,3,4,0,0,0,0,0 int hiz*LENGHT1-11-1(2,3,4,0,0,0,0,0 int hiz*LENGHT1-1-1(3,3,4,0,0,0,0,0 int hiz*Linght11-1(3,3,4,0,0,0,0,0,0 int hiz*Linght11-1(3,3,4,0,0,0,0,0,0 int hiz*Linght11-1(3,3,4,0,0,0,0,0,0) int hiz*Linght11-1(3,3,4,0,0,0,0,0,0) int hiz*Linght11-1(3,3,4,0,0,0,0,0,0,0,0,0,0,0) int hiz*Linght11-1(3,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,</stdio.h>	#includestdio.h #fincludescompi fincludescompi finc	>	#include "L138_LCDK aic3106_init.h" #define BUFSIZE 512 int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); int32_inbuffer (BUFSIZE); i(int16_isample_data; sample_data input_left_sample(); inbuffer(buf_ptf = sample_data; buf_ptf (buf_ptf = sample_data); inbuffer(buf_ptf = sample_data); return; } int main(void) (L138_initialise_intrifS_8000_HZ_ADC_GAIN_OT ACA_TTEN_DB_LCDB_LINE_INPUT; while(1);	clear all; clc; close all; x=(3 -10 13 2 0 1 2 1) h=(11 1 1) L=3; M=3 N=L+M-1 x=(x zeros(1,L)) h=(h zeros(1,P)) y=(1 zeros(1,P)) y=(1 zeros(1,M-1)) x=(x zeros(1,M-1)) y=(x zeros(1,M-1)	::end)+x2(1:M-1) x2(M:end)] Overlap add method")	stem(t,y) close all x=[3 -1 0 1 3 2 0 1 2 1]; h=[1 1 1]; L=3; M=3; N=1+M-1; x=[zeros(1,M-1) x zeros(1,M-1)]; s=zize(x); y=[]; for i=LLs(2)-1, 1.2×(x(i)+M-1), 2.2×(x(i)+M-1), y=[y x 2(Mx,nd)]; subplot(Z(1); stem(y) xlabe(l'n'n) title("Linear Convolution using Overlap save n y1=conv(x,h); subplot(Z(2); stem(y1) xlabe("n')