**Aim :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 1 | **STUDY OF MATLAB** | Date : |

**Result :**

**Aim :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 2 | **GENERATION OF SIGNALS** | Date : |

**Program :**

%program to generate unit impulse signal

clc;

clear all;

close all;

t=-2:1:2;

y=[zeros(1,2),ones(1),zeros(1,2)];

figure;

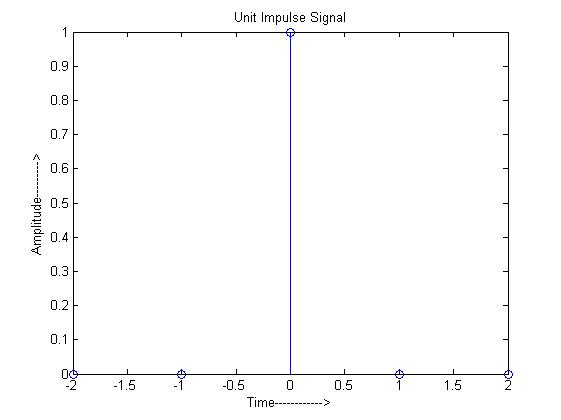
stem(t,y);

ylabel('Amplitude--------->');

xlabel('Time------------>');

title('Unit Impulse Signal');

**Output :**



**Program :**

%Program to generate unit step signal(discrete signal);

clc;

close all;

clear all;

n=input('Enter the length of the sequence');

t=0:1:n-1;

y=[ones(1,n)];

stem(t,y);

ylabel('Amplitude------>');

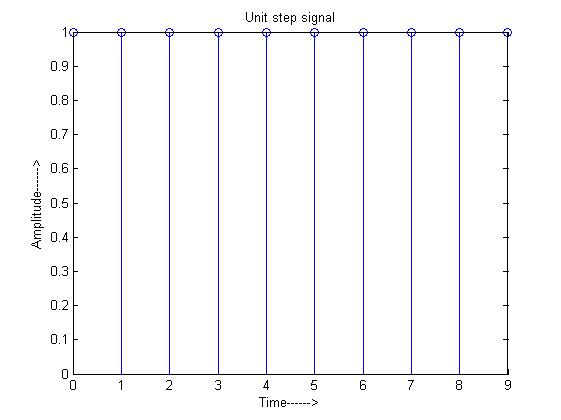
xlabel('Time------>');

title('Unit step signal');

**Output :**

Enter the length of the sequence10

>>



**Program :**

%Program to generate the unit step signal (continuous signal) ;

clc;

clear all;

close all;

n=input('Enter the length of the sequence');

t=0:1:n-1;

y=[ones(1,n)];

plot(t,y);

ylabel('Amplitude---------->');

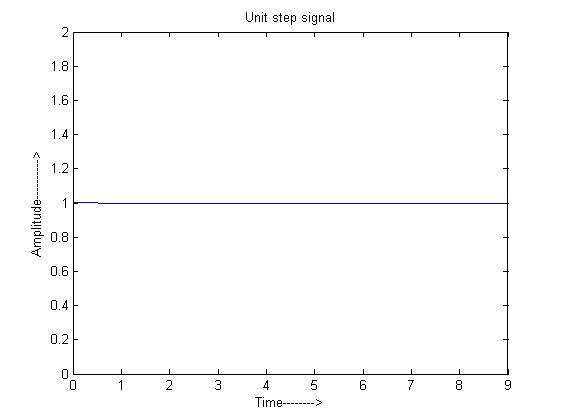
xlabel('Time-------->');

title('Unit step signal');

**Output :**

Enter the length of the sequence10

>>



**Program :**

%Program to generate unit ramp signal(discrete signal);

clc;

clear all;

close all;

n=input('Enter the length of the sequence');

t=0:1:n-1;

stem(t,t);

ylabel('Amplitude------>');

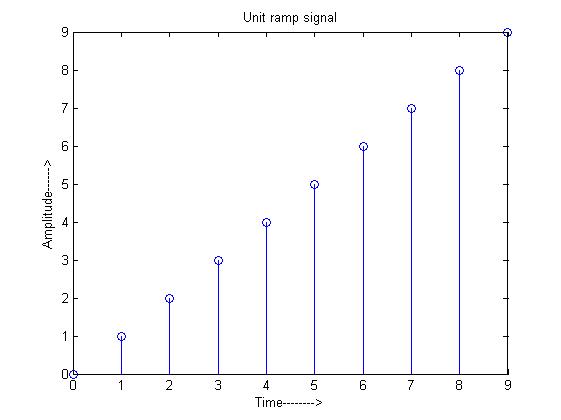
xlabel('Time-------->');

title('Unit ramp signal');

**Output :**

Enter the length of the sequence10

>>



**Program :**

%Program to generate unit ramp signal(continuous signal);

clc;

clear all;

close all;

n=input('Enter the length of the sequence');

t=0:1:n-1;

plot(t,t);

ylabel('Amplitude----->');

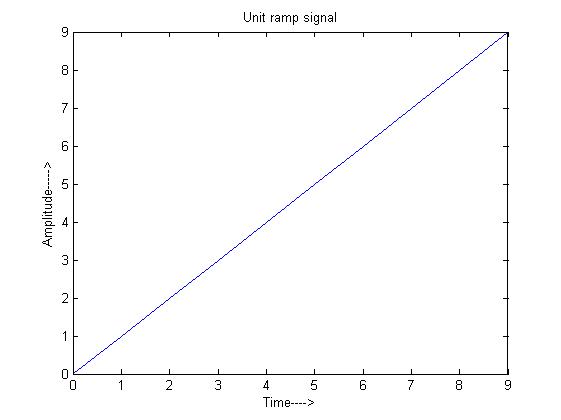
xlabel('Time---->');

title('Unit ramp signal');

**Output :**

Enter the length of the sequence10

>>



**Program :**

%Program to generate sinusoidal signal;

clc;

clear all;

close all;

f=input('Enter the frequency of the signal');

t=0:0.01:pi;

y=sin(2\*pi\*f\*t);

plot(t,y);

ylabel('Amplitude-------->');

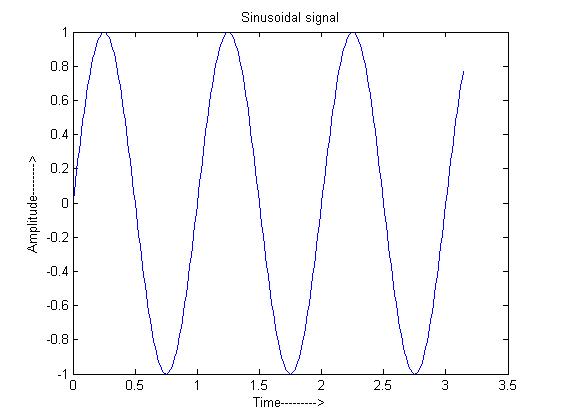
xlabel('Time--------->');

title('Sinusoidal signal');

**Output :**

Enter the frequency of the signal1

>>



**Program :**

%Program to generate co sinusoidal signal;

clc;

clear all;

close all;

f=input('Enter the frequency of the signal');

t=0:0.01:pi;

y=cos (2\*pi\*f\*t);

plot(t,y);

ylabel('Amplitude-------->');

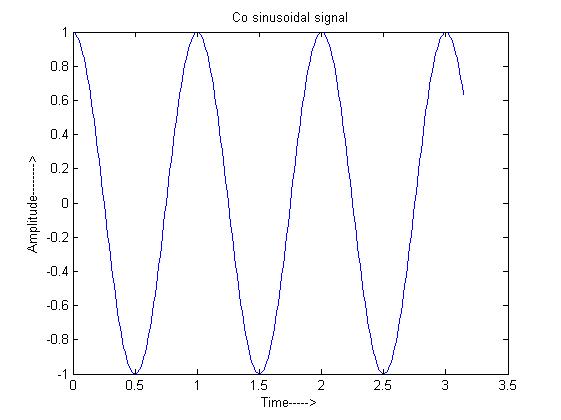
xlabel('Time----->');

title('Co sinusoidal signal');

**Output :**

Enter the frequency of the signal1

>>



**Program :**

%Program to generate discrete time signal;

clc;

clear all;

close all;

x=input('Enter the sequence');

figure;

stem(x);

ylabel('Amplitude-------->');

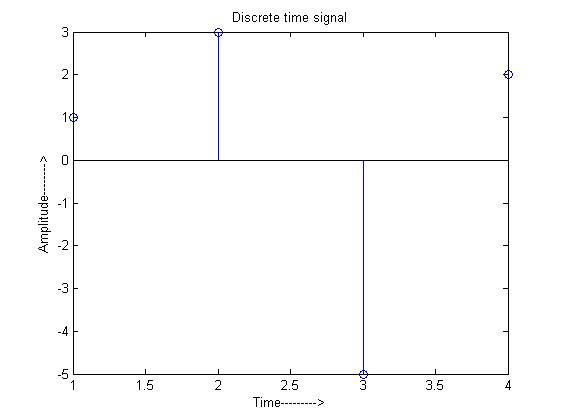
xlabel('Time--------->');

title('Discrete time signal');

**Output :**

Enter the sequence[1 3 -5 2]

>>



**Result :**

**Aim :**

**Algorithm :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 3 | **IMPLEMENTATION OF FFT** | Date : |

**Program:**

%Program to calculate FFT;

clc;

clear all;

close all;

x=input('Enter the input sequence');

y=fft(x);

subplot(2,1,1);

stem(x);

xlabel('Time------->');

ylabel('Amplitude----------->');

subplot(2,1,2);

stem(y);

xlabel('Time-------->');

ylabel('Amplitude---------->');

disp('Input sequence');x

disp('FFT values');y

title('Calculation of FFT');

**Output:**

Enter the input sequence[1 2 3 4 4 3 2 1]

Input sequence

x =

1 2 3 4 4 3 2 1

FFT values

y =

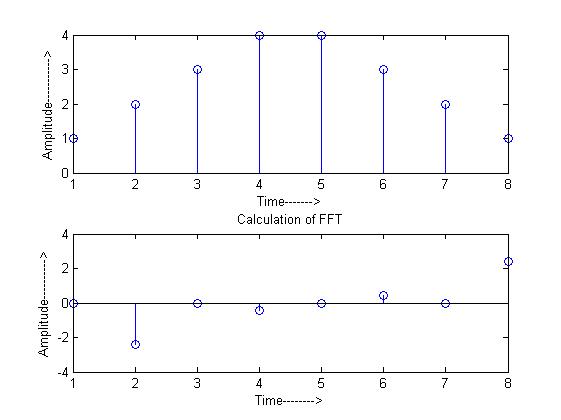
Columns 1 through 6

20.0000 -5.8284 - 2.4142i 0 -0.1716 - 0.4142i 0 -0.1716 + 0.4142i

Columns 7 through 8

0 -5.8284 + 2.4142i

>>



**Result:**

|  |  |  |
| --- | --- | --- |
| Ex. No : 4 | **IMPLEMENTATION OF DFT** | Date : |

**Aim :**

**Algorithm :**

**Program :**

%Program to calculate DFT

clc;

clear all;

close all;

x=input('Enter the sequence');

N=length(x);

for k=0:1:N-1;

for n=0:1:N-1;

p=exp(-i\*2\*pi\*k\*n/N);

y(n+1,k+1)=p;

end;

end;

y=x\*y;

disp(y);

subplot(2,1,1);

stem(x);

xlabel('Time-------->');

ylabel('Amplitude---------->');

title('Input sequence');

subplot(2,1,2);

stem(y);

xlabel('Time-------->');

ylabel('Amplitude---------->');

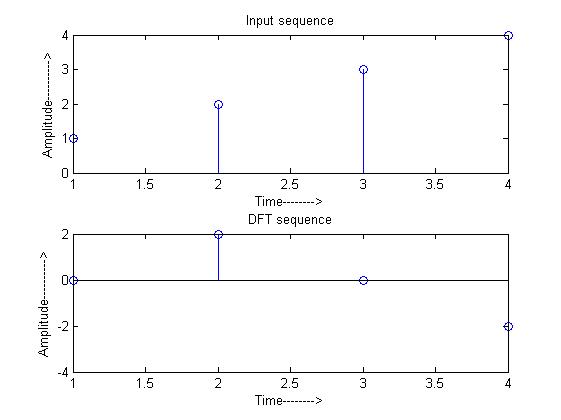
title('DFT sequence');

**Output :**

Enter the sequence[1 2 3 4]

10.0000 -2.0000 + 2.0000i -2.0000 - 0.0000i -2.0000 - 2.0000i

>>



**Result :**

**Aim :**

**Algorithm :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 5 | **LINEAR & CIRCULAR CONVOLUTION** | Date : |

**Program :**

%Program to calculate linear convolution;

clc;

clear all;

close all;

x1=input('Enter the input sequence');

x2=input('Enter the input sequence');

y=convn(x1,x2);

subplot(3,1,1);

stem(x1);

ylabel('Amplitude-------->');

xlabel('Time---------->');

title('Input sequence x1');

subplot(3,1,2);

stem(x2);

ylabel('Amplitude-------->');

xlabel('Time---------->');

title('Input sequence x2');

subplot(3,1,3);

stem(y);

ylabel('Amplitude-------->');

xlabel('Time---------->');

disp('Input sequence x1');

disp(x1);

disp('Input sequence x2');

disp(x2);

disp('Convn value y');

disp(y);

title('Convolution');

**Output :**

Enter the input sequence[0 1 2 3]

Enter the input sequence[1 2 3 4]

Input sequence x1

0 1 2 3

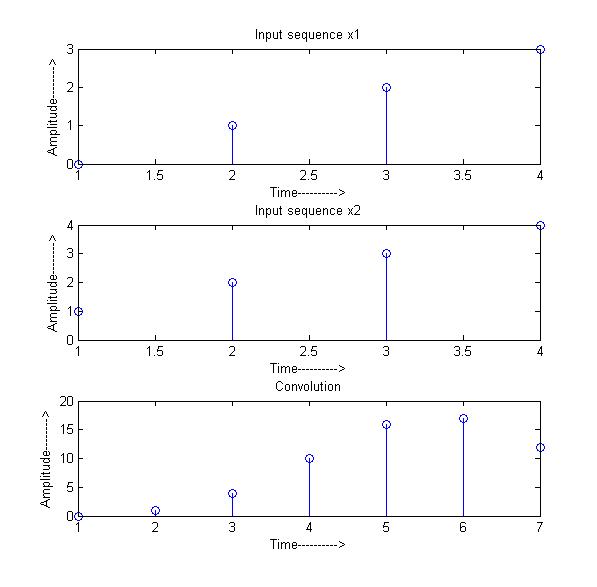
Input sequence x2

1 2 3 4

Convn value y

0 1 4 10 16 17 12

>>



**Program :**

%Program to calculate circular convolution;

clc;

clear all;

close all;

x=input('Enter the input sequence(x)');

h= input('Enter the input sequence(h)');

y=fft(x);

n=fft(h);

for i=1:length(x)

s(i)=y(i)\*n(i);

end

z=ifft(s);

subplot(3,1,1);

stem(x);

ylabel('Amplitude-------->');

xlabel('Time---------->');

title('x(n)');

subplot(3,1,2);

stem(h);

ylabel('Amplitude-------->');

xlabel('Time---------->');

title('h(n)');

subplot(3,1,3);

stem(z);

ylabel('Amplitude-------->');

xlabel('Time---------->');

disp('Input sequence x(n)');

disp(x);

disp('Input sequence');

disp(h);

disp('Convn values');

disp(z);

title('Circular Convolution');

**Output :**

Enter the input sequence(x)[0 1 2 3]

Enter the input sequence(h)[1 2 3 4]

Input sequence x(n)

0 1 2 3

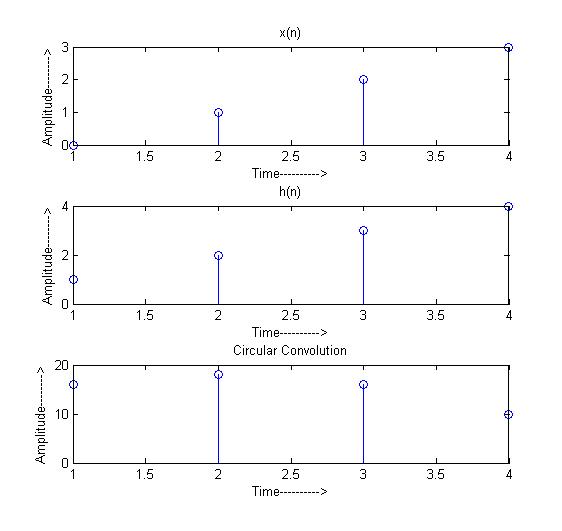
Input sequence

1 2 3 4

Convn values

16 18 16 10

>>



**Result :**

**Aim :**

**Algorithm :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 6 | **SAMPLING THEOREM** | Date : |

**Program:**

%Solution for section(a);

clc;

clear all;

close all;

f1=1/128;

f2=5/128;

n=0:255;

fc=50/128;

x=cos(2\*pi\*f1\*n)+cos(2\*pi\*f2\*n);

xa=cos(2\*pi\*fc\*n);

xamp=x.\*xa;

subplot(2,2,1);

plot(n,x);

title('x(n)');

xlabel('n------>');

ylabel('Amplitude--------?');

subplot(2,2,2);

plot(n,xa);

title('xa(n)');

xlabel('n------>');

ylabel('Amplitude--------?');

subplot(2,2,3);

plot(n,xamp);

xlabel('n-------->');

ylabel('Amplitude--------->');

%128 point DFT computation-solution for section(b);

n=0:127;

figure;

n1=128;

f1=1/128;

f2=5/128;

fc=50/128;

x=cos(2\*pi\*f1\*n)+ cos(2\*pi\*f2\*n);

xc= cos(2\*pi\*fc\*n);

xamp=x.\*xc;

xam=fft(xamp,n1);

stem(n,xam);

title('xamp(n)');

xlabel('n-------->');

ylabel('Amplitude--------->');

%128 point DFT computation-solution for section(c);

n=0:99;

figure;

n2=0:n1-1;

f1=1/128;

f2=5/128;

fc=50/128;

x=cos(2\*pi\*f1\*n)+ cos(2\*pi\*f2\*n);

xc= cos(2\*pi\*fc\*n);

xamp=x.\*xc;

for i=1:100

xamp1(i)=xamp(i);

end

xam=fft(xamp1,n1);

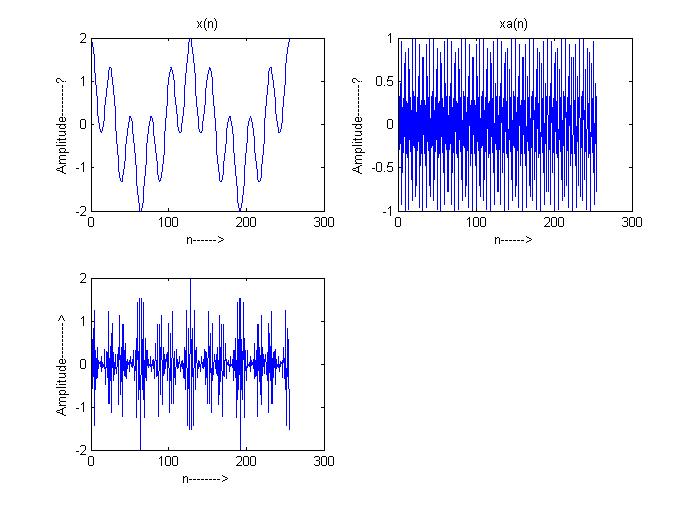
stem(n2,xam);

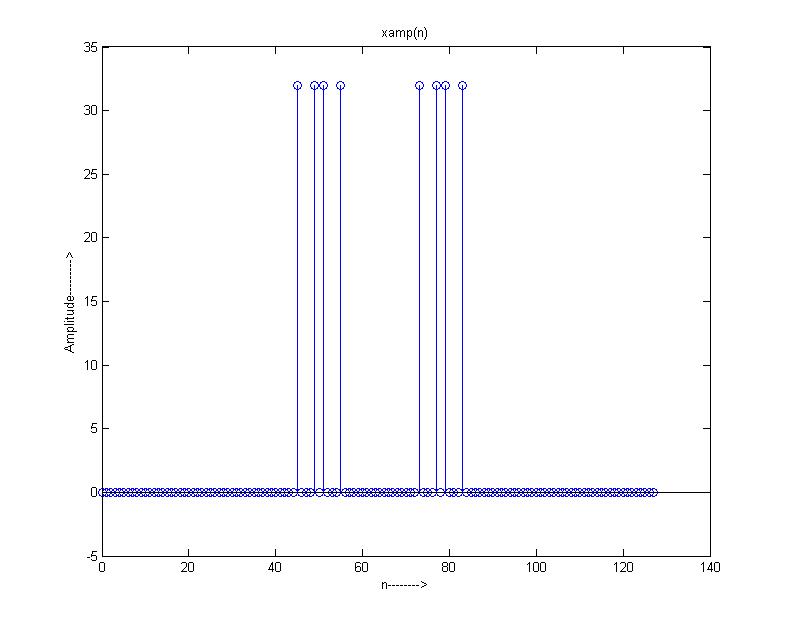
title('xamp(n)');

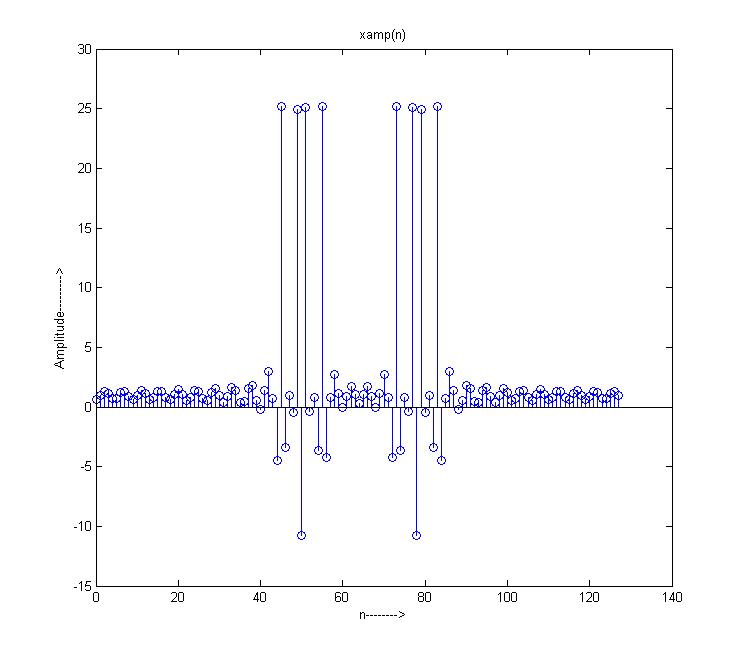
xlabel('n-------->');

ylabel('Amplitude--------->');

**Output:**







**Result :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 7 | **INTERPOLATION** | Date : |

**Aim :**

**Algorithm :**

**Program :**

%Program for interpolation

clc;

clear all;

close all;

N=input('input length=');

L=input('up-sampling factor=');

f0=input('input signal frequency=');

%generate the input sinusoidal sequence%

n=0:N-1;

x=sin(2\*pi\*f0\*n);

%generate the up sampled sequence%

y=zeros (1,L\*length(x));

y([1:L:length(y)])=x;

%to plot the input and output sequences

subplot(2,1,1);

stem(n,x);

xlabel('time index n');

ylabel('amplitude');

title('input sequence');

subplot(2,1,2);

stem(n,y (1:length(x)));

xlabel('time index n');

ylabel('amplitude');

title(['output sequence up-sampled by:num2str(L)']);

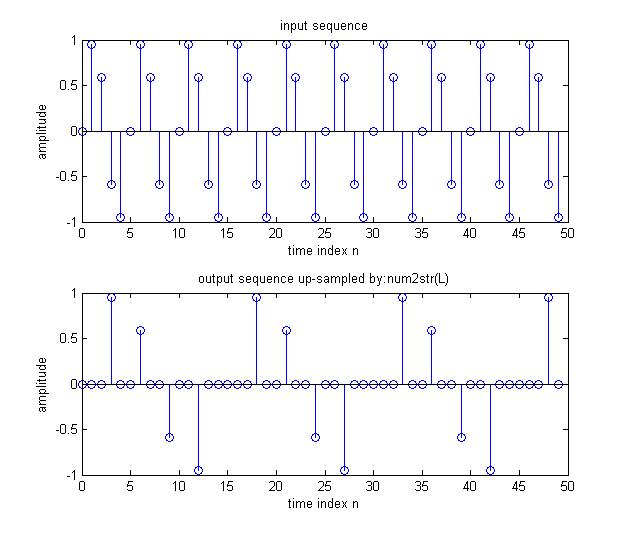
**Output :**

input length=50

up -sampling factor=3

input signal frequency=.2

>>



**Result :**

**Aim :**

**Algorithm :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 8 | **DECIMATION** | Date : |

**Program :**

%Program for decimation

clc;

close all;

clear all;

N=input('output length=');

M=input('down sampling factor=');

f0=input('input signal frequency=');

%generate the input sinusoidal sequence%

n=0 : N-1;

m=0 : N\*M-1;

x=sin(2\*pi\*f0\*m);

%generate the down sampled sequence%

y=x([1:M:length(x)]);

%to plot the input and output sequence%

subplot(2,1,1);

stem(n,x(1:N));

title('input sequence');

subplot(2,1,2);

stem(n,y);

title(['output sequence down sampled by:num2str(m)']);

xlabel('time index');

ylabel('amplitude');

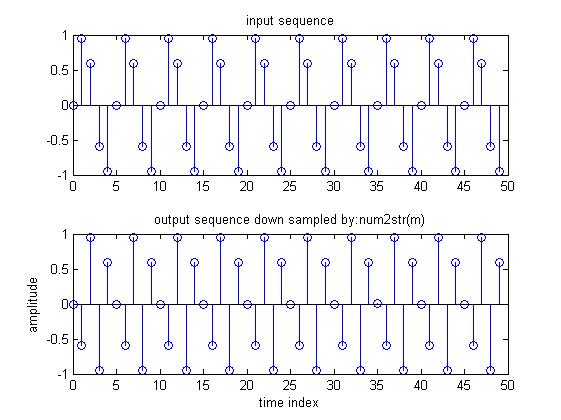
**Output :**

output length=50

down sampling factor=3

input signal frequency=.2

>>



**Result :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 9 | **FIR FILTER DESIGN-RECTANGULAR WINDOW** | Date : |

**Aim :**

**Algorithm :**

**Program :**

%Program for design FIR lowpass, bandpass, band stop filter using rectangular window;

clc;

clear all;

close all;

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter the stop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

num=-20\*log10(sqrt(rp\*rs))-13;

dem=14.6\*(fs-fp)/f;

n=ceil(num/dem);

n1=n+1;

if(rem(n,2)~=0);

n1=n;

n=n-1;

end

y=boxcar(n1);

%low pass filter;

b=fir1(n,wp,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,1);

plot(o/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%High pass filter;

b=fir1(n,wp,'high',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,2);

plot(o/pi,m);

xlabel('(b)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band pass filter;

wn=[wp,ws];

b=fir1(n,wn,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,3);

plot(o/pi,m);

xlabel('(c)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band stop filter;

b=fir1(n,wn,'stop',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,4);

plot(o/pi,m);

xlabel('(d)Normalized frequency------------->');

ylabel('Gain in dB--------->');

**Output :**

Enter the pass band ripple0.05

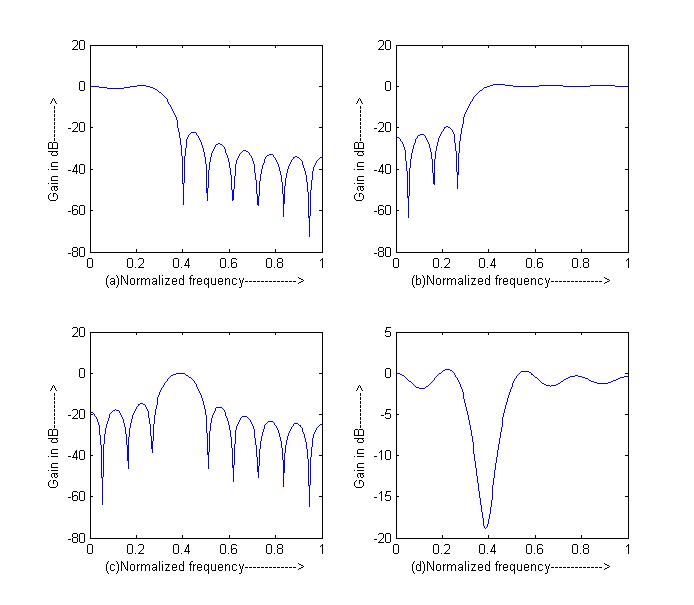
Enter the stop band ripple0.04

Enter the pass band frequency1500

Enter the stop band frequency2000

Enter the sampling frequency9000

>>



**Result :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 10 | **FIR FILTER DESIGN-HAMMING WINDOW** | Date : |

**Aim :**

**Algorithm :**

**Program :**

%Design of FIR filter using Hamming window;

clc;

clear all;

close all;

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter thestop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

num=-20\*log10(sqrt(rp\*rs))-13;

dem=14.6\*(fs-fp)/f;

n=ceil(num/dem);

n1=n+1;

if(rem(n,2)~=0);

n1=n;

n=n-1;

end

y=hamming(n1);

%low pass filter;

b=fir1(n,wp,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,1);

plot(o/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%High pass filter;

b=fir1(n,wp,'high',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,2);

plot(o/pi,m);

xlabel('(b)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band pass filter;

wn=[wp,ws];

b=fir1(n,wn,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,3);

plot(o/pi,m);

xlabel('(c)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band stop filter;

b=fir1(n,wn,'stop',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,4);

plot(o/pi,m);

xlabel('(d)Normalized frequency------------->');

ylabel('Gain in dB--------->');

**Output :**

Enter the pass band ripple0.02

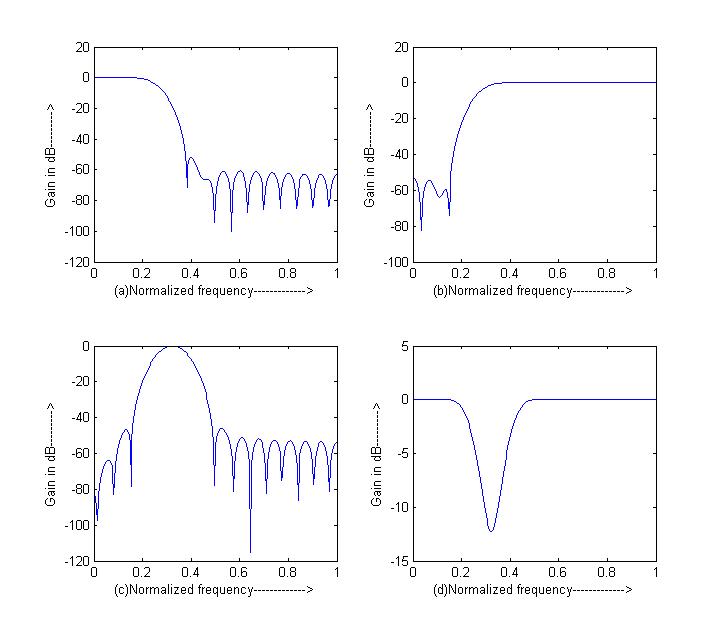
Enter the stop band ripple0.01

Enter the pass band frequency1200

Enter thestop band frequency1700

Enter the sampling frequency9000

>>



**Result :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 11 | **FIR FILTER DESIGN-HANNING WINDOW** | Date : |

**Aim :**

**Algorithm :**

**Program :**

%Design of FIR filter using Hanning window;

clc;

clear all;

close all;

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter the stop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

num=-20\*log10(sqrt(rp\*rs))-13;

dem=14.6\*(fs-fp)/f;

n=ceil(num/dem);

n1=n+1;

if(rem(n,2)~=0);

n1=n;

n=n-1;

end

y=hanning(n1);

%low pass filter;

b=fir1(n,wp,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,1);

plot(o/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%High pass filter;

b=fir1(n,wp,'high',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,2);

plot(o/pi,m);

xlabel('(b)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band pass filter;

wn=[wp,ws];

b=fir1(n,wn,y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,3);

plot(o/pi,m);

xlabel('(c)Normalized frequency------------->');

ylabel('Gain in dB--------->');

%Band stop filter;

b=fir1(n,wn,'stop',y);

[h,o]=freqz(b,1,256);

m=20\*log10(abs(h));

subplot(2,2,4);

plot(o/pi,m);

xlabel('(d)Normalized frequency------------->');

ylabel('Gain in dB--------->');

**Output:**

Enter the pass band ripple0.03

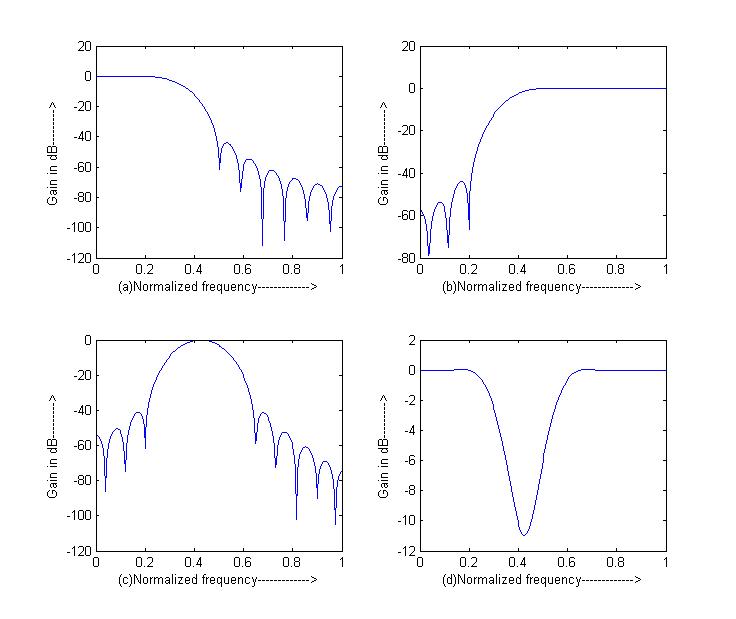
Enter the stop band ripple0.01

Enter the pass band frequency1400

Enter the stop band frequency2000

Enter the sampling frequency8000

>>



**Result :**

**Aim :**

**Algorithm :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 12 | **BUTTERWORTH DIGITAL IIR FILTERS** | Date : |

**Program :**

%Program for the design of Butterworth low pass filter;

clc;

clear all;

close all;

format long

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter the stop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

[n,wn]=buttord(wp,ws,rp,rs);

[b,a]=butter(n,wn);

w=0:.01:pi;

[h,om]=freqz(b,a,w);

m=20\*log10(abs(h));

an=angle(h);

subplot(2,1,1);

plot(om/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

subplot(2,1,2);

plot(om/pi,an);

xlabel('(b)Normalized frequency------------->');

ylabel('Phase in radians--------->');

**Output :**

Enter the pass band ripple0.5

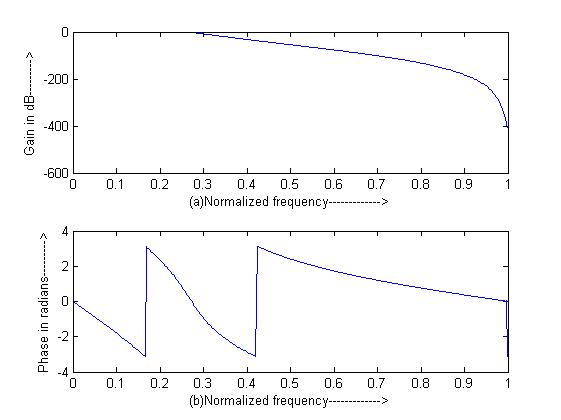
Enter the stop band ripple50

Enter the pass band frequency1200

Enter the stop band frequency2400

Enter the sampling frequency10000

>>



**Program :**

%Program for design Butterworth high pass filter;

clc;

clear all;

close all;

format long

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter the stop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

[n,wn]=buttord(wp,ws,rp,rs);

[b,a]=butter(n,wn,'high');

w=0:.01:pi;

[h,om]=freqz(b,a,w);

m=20\*log10(abs(h));

an=angle(h);

subplot(2,1,1);

plot(om/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

subplot(2,1,2);

plot(om/pi,an);

xlabel('(b)Normalized frequency------------->');

ylabel('Phase in radians--------->');

**Output :**

Enter the pass band ripple0.5

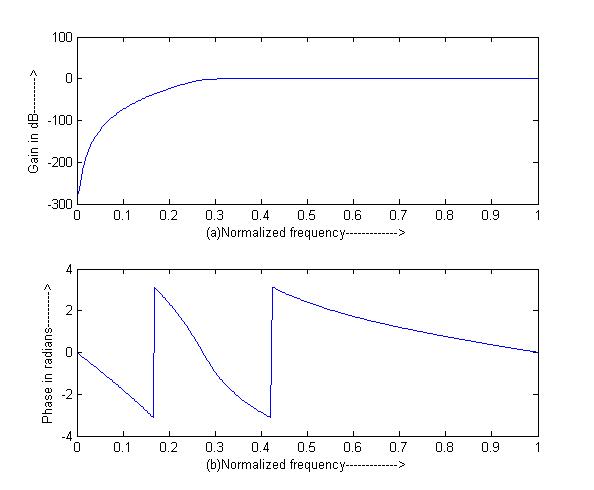
Enter the stop band ripple50

Enter the pass band frequency1200

Enter the stop band frequency2400

Enter the sampling frequency10000

>>



**Program :**

%Program for design Butterworth band pass digital filter;

clc;

clear all;

close all;

format long

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter thestop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

[n]=buttord(wp,ws,rp,rs);

[wn]=[wp ws];

[b,a]=butter(n,wn,'bandpass');

w=0:.01:pi;

[h,om]=freqz(b,a,w);

m=20\*log10(abs(h));

an=angle(h);

subplot(2,1,1);

plot(om/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

subplot(2,1,2);

plot(om/pi,an);

xlabel('(b)Normalized frequency------------->');

ylabel('Phase in radians--------->');

**Output :**

Enter the pass band ripple0.3

Enter the stop band ripple40

Enter the pass band frequency1500

Enter thestop band frequency2000

Enter the sampling frequency9000

>>



**Program :**

%Program for design Butterworth band stop digital filter;

clc;

clear all;

close all;

format long

rp=input('Enter the pass band ripple');

rs=input('Enter the stop band ripple');

fp=input('Enter the pass band frequency');

fs=input('Enter the stop band frequency');

f=input('Enter the sampling frequency');

wp=2\*fp/f;

ws=2\*fs/f;

[n]=buttord(wp,ws,rp,rs);

[wn]=[wp ws];

[b,a]=butter(n,wn,'stop');

w=0:.01:pi;

[h,om]=freqz(b,a,w);

m=20\*log10(abs(h));

an=angle(h);

subplot(2,1,1);

plot(om/pi,m);

xlabel('(a)Normalized frequency------------->');

ylabel('Gain in dB--------->');

subplot(2,1,2);

plot(om/pi,an);

xlabel('(b)Normalized frequency------------->');

ylabel('Phase in radians--------->');

**Output :**

Enter the pass band ripple0.4

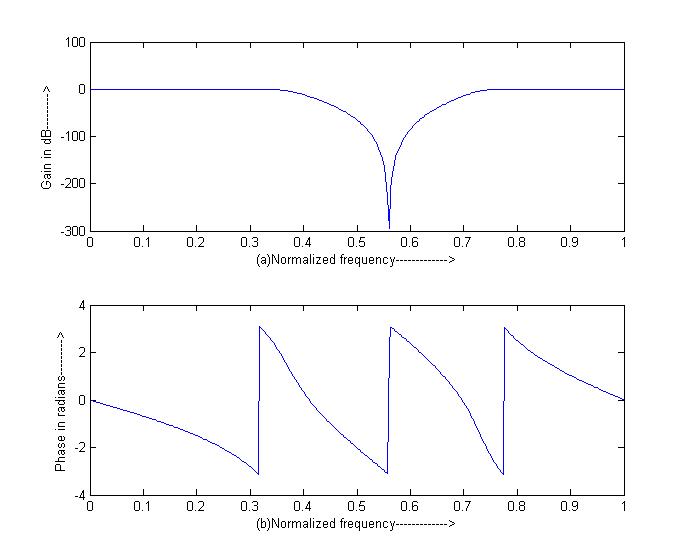
Enter the stop band ripple46

Enter the pass band frequency1100

Enter the stop band frequency2200

Enter the sampling frequency6000

>>



**Result :**

|  |  |  |
| --- | --- | --- |
| Ex. No : 13 | **INTRODUCTION TO DSP PROCESSOR** | Date : |

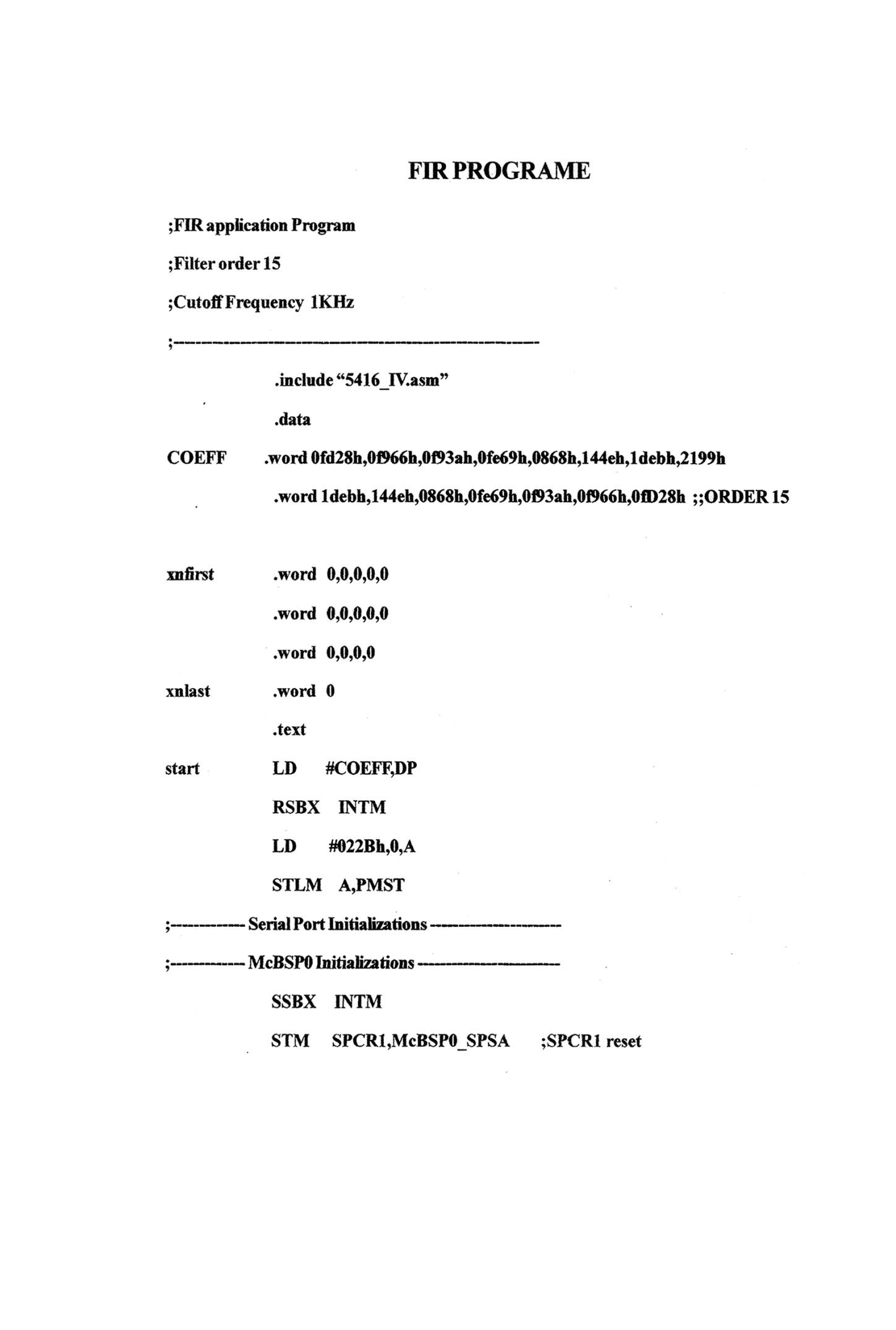
**Aim :**

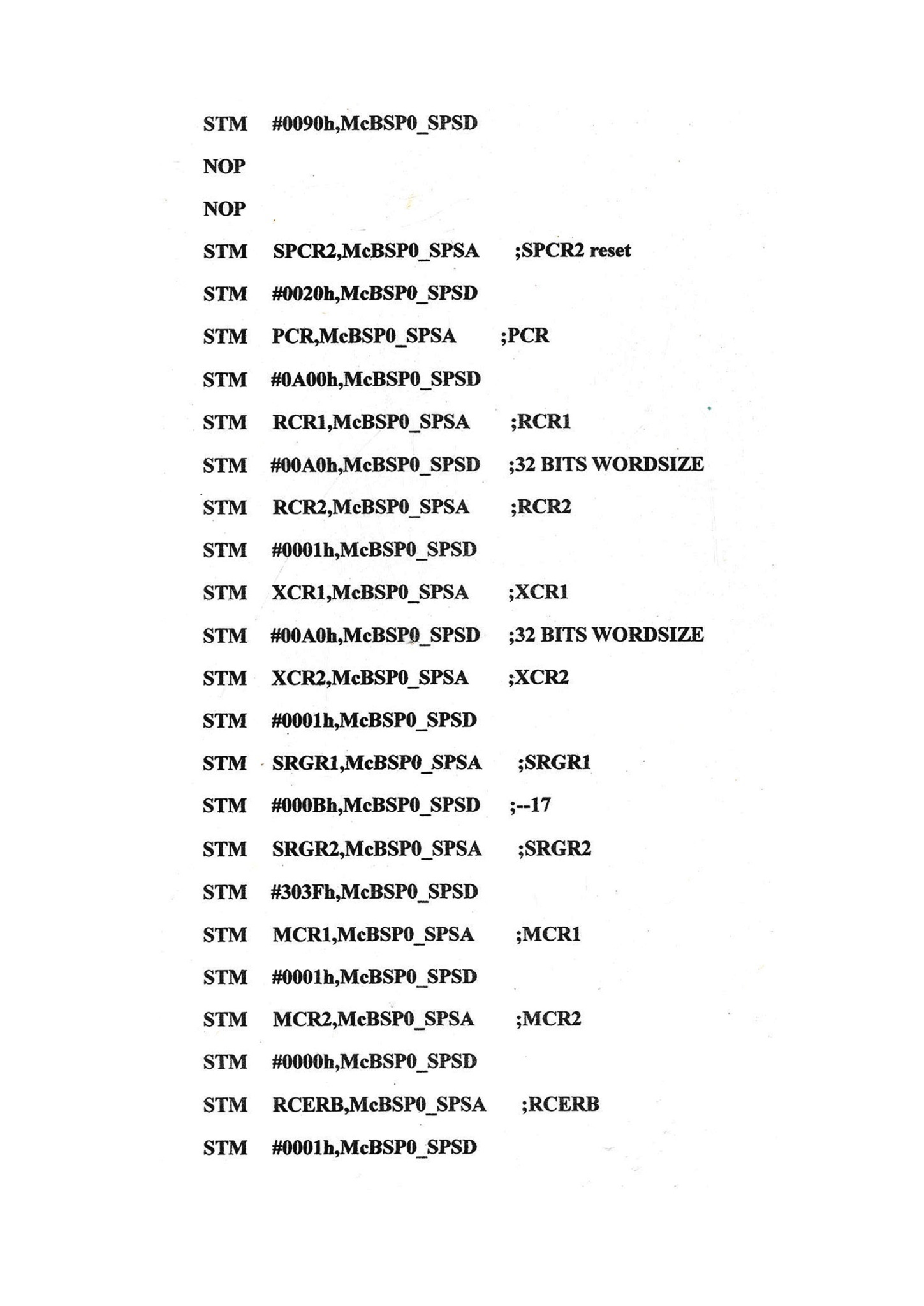
**Result :**

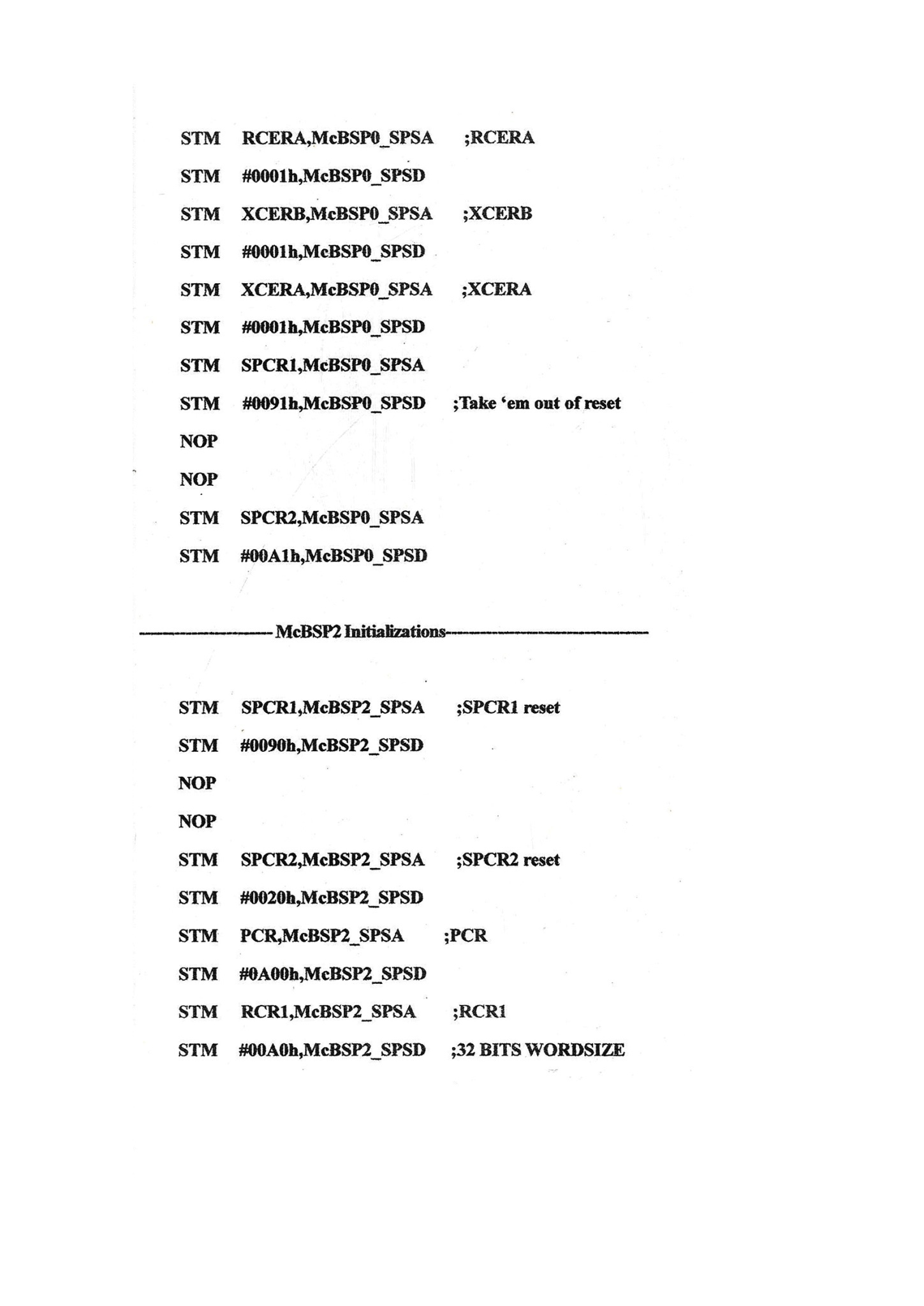
|  |  |  |
| --- | --- | --- |
| Ex. No : 14 | **FIR FILTER USING DSP PROCESSOR** | Date : |

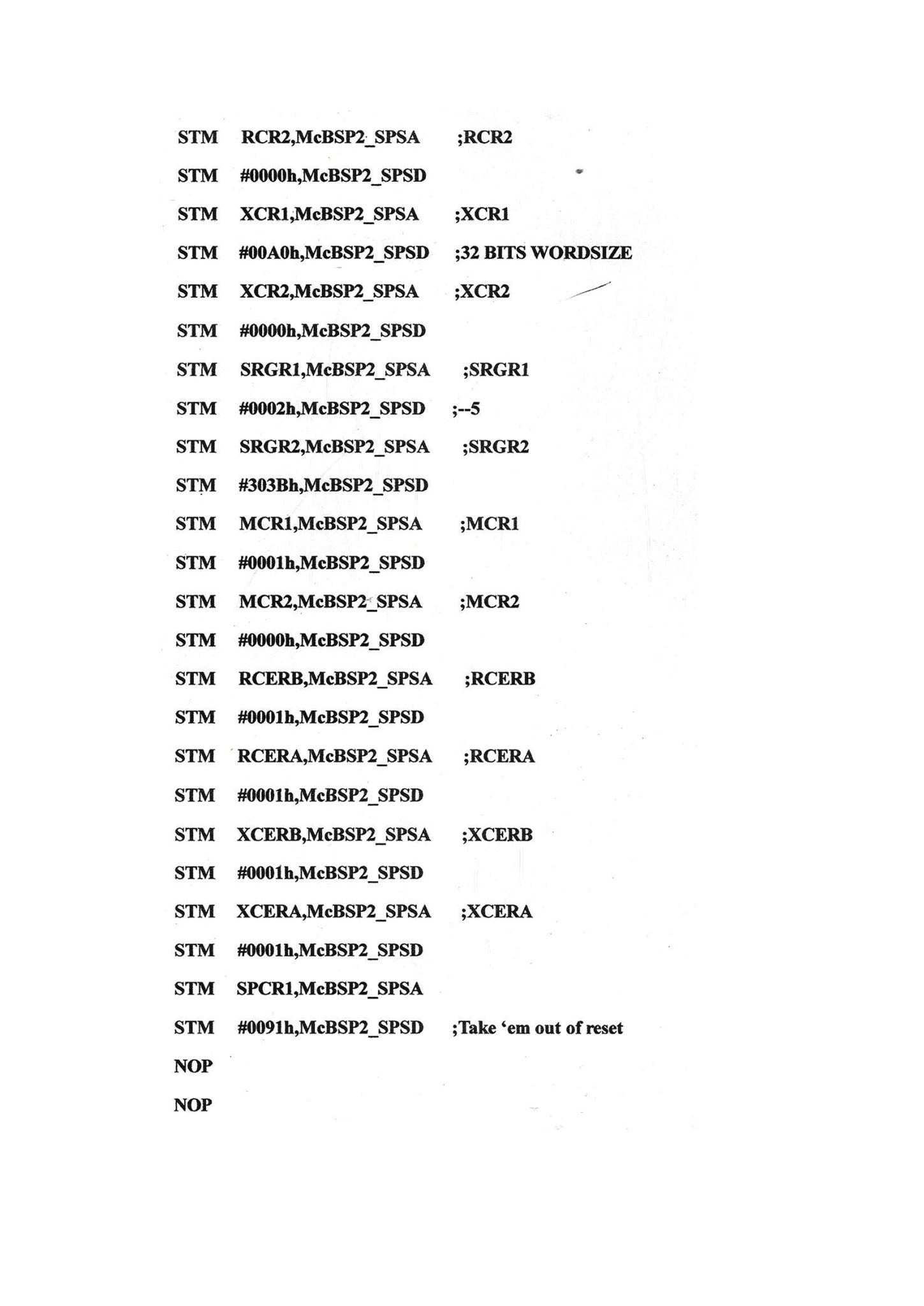
**Aim :**

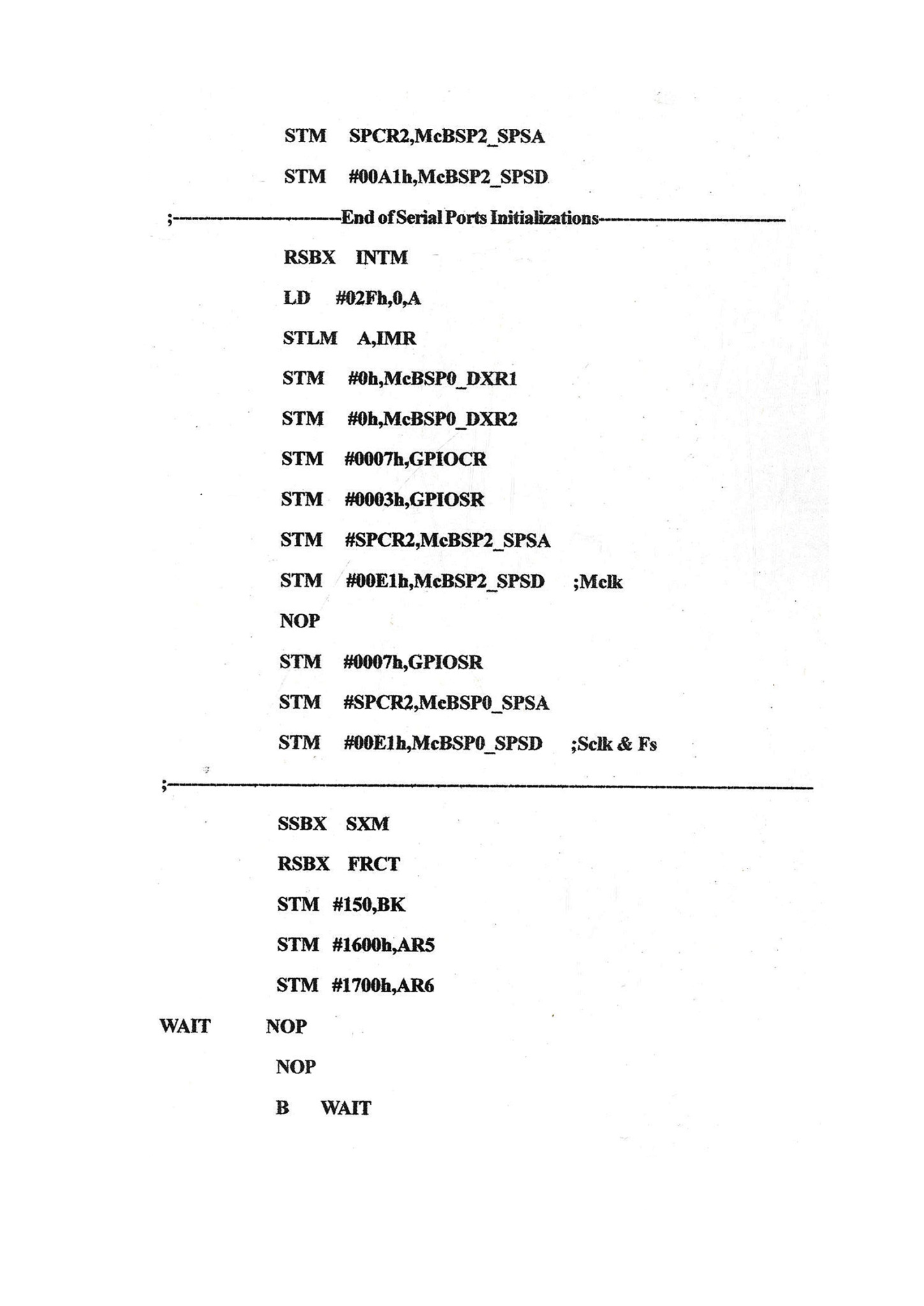
**Algorithm :**

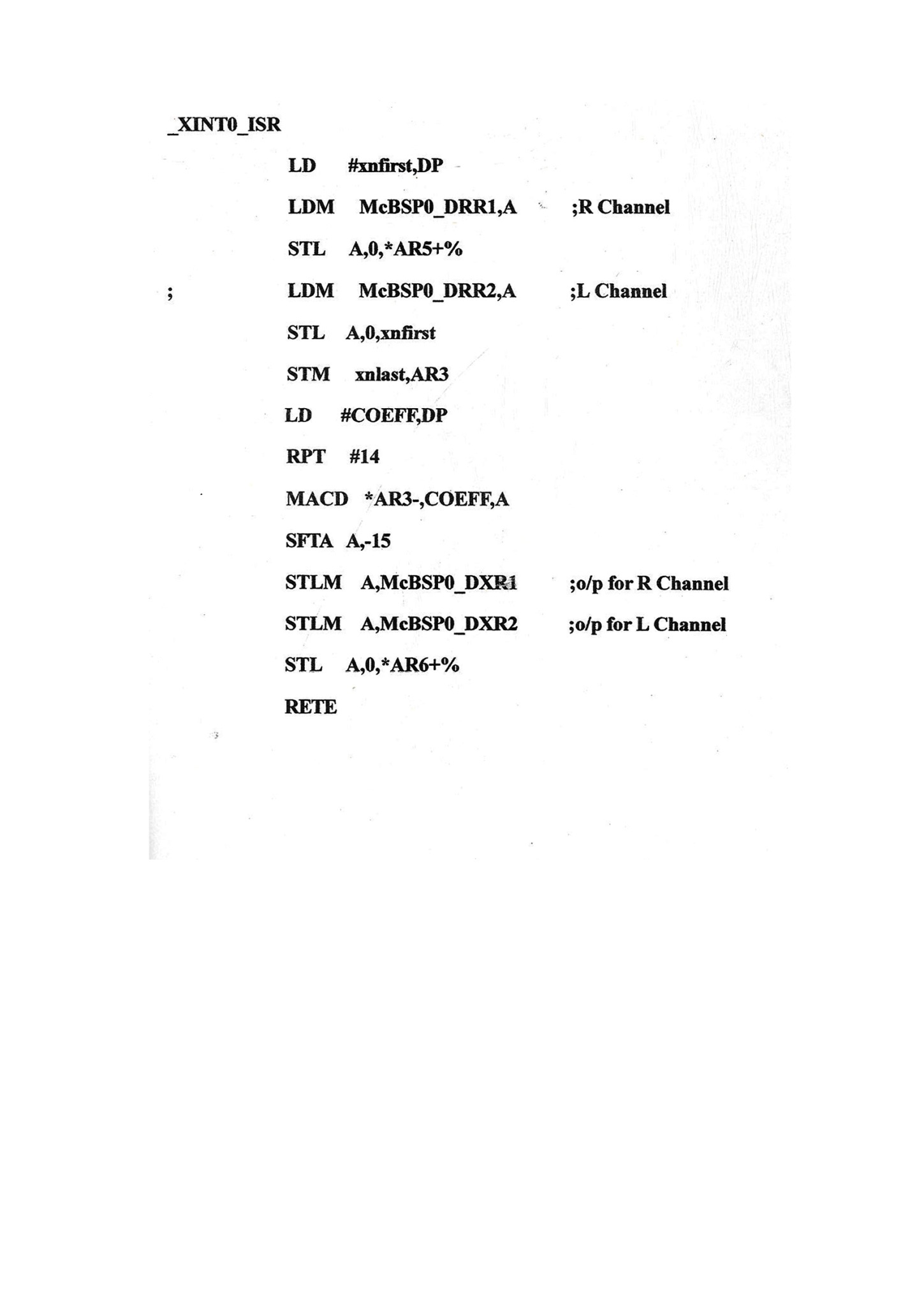
****

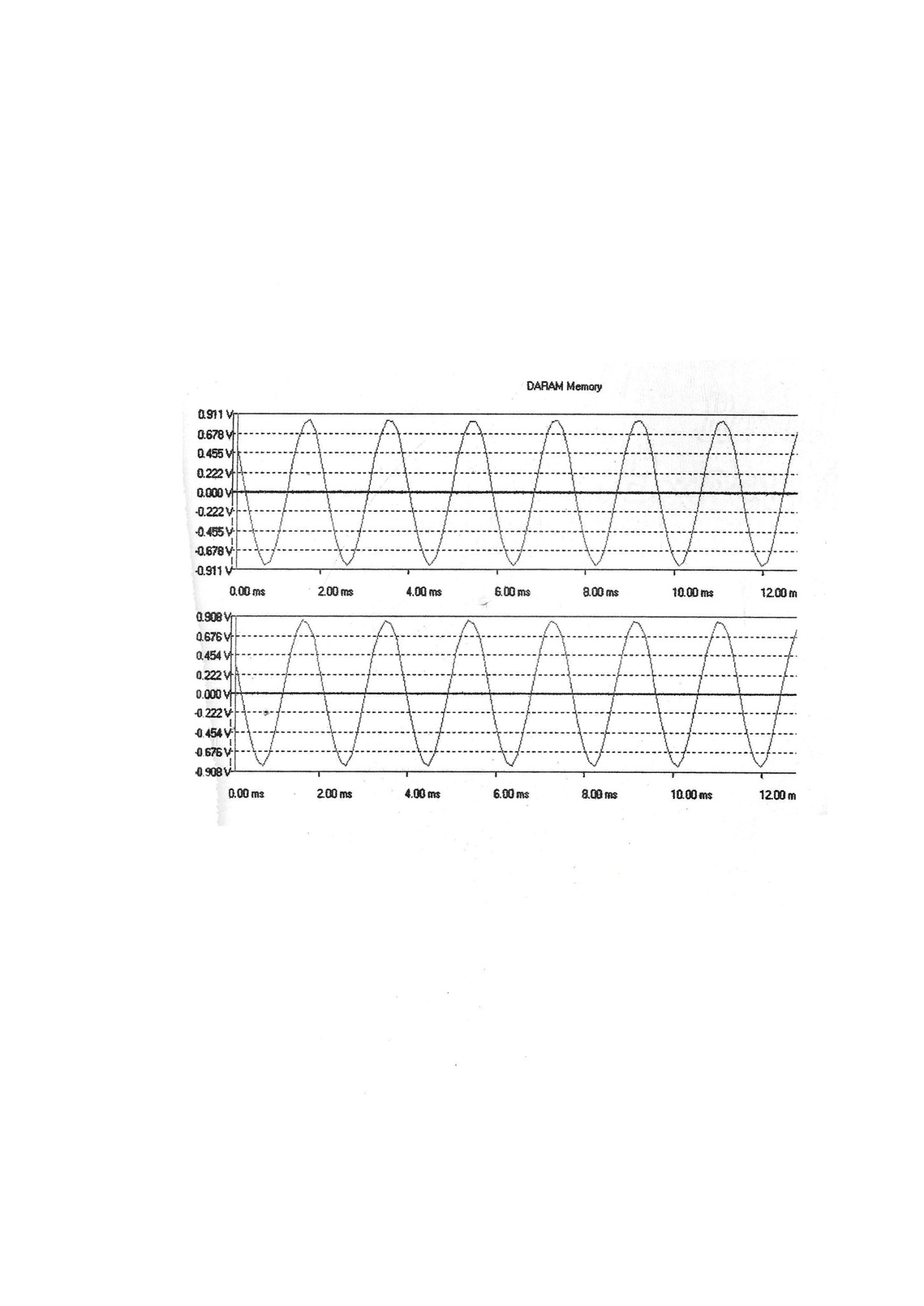
****

****

****

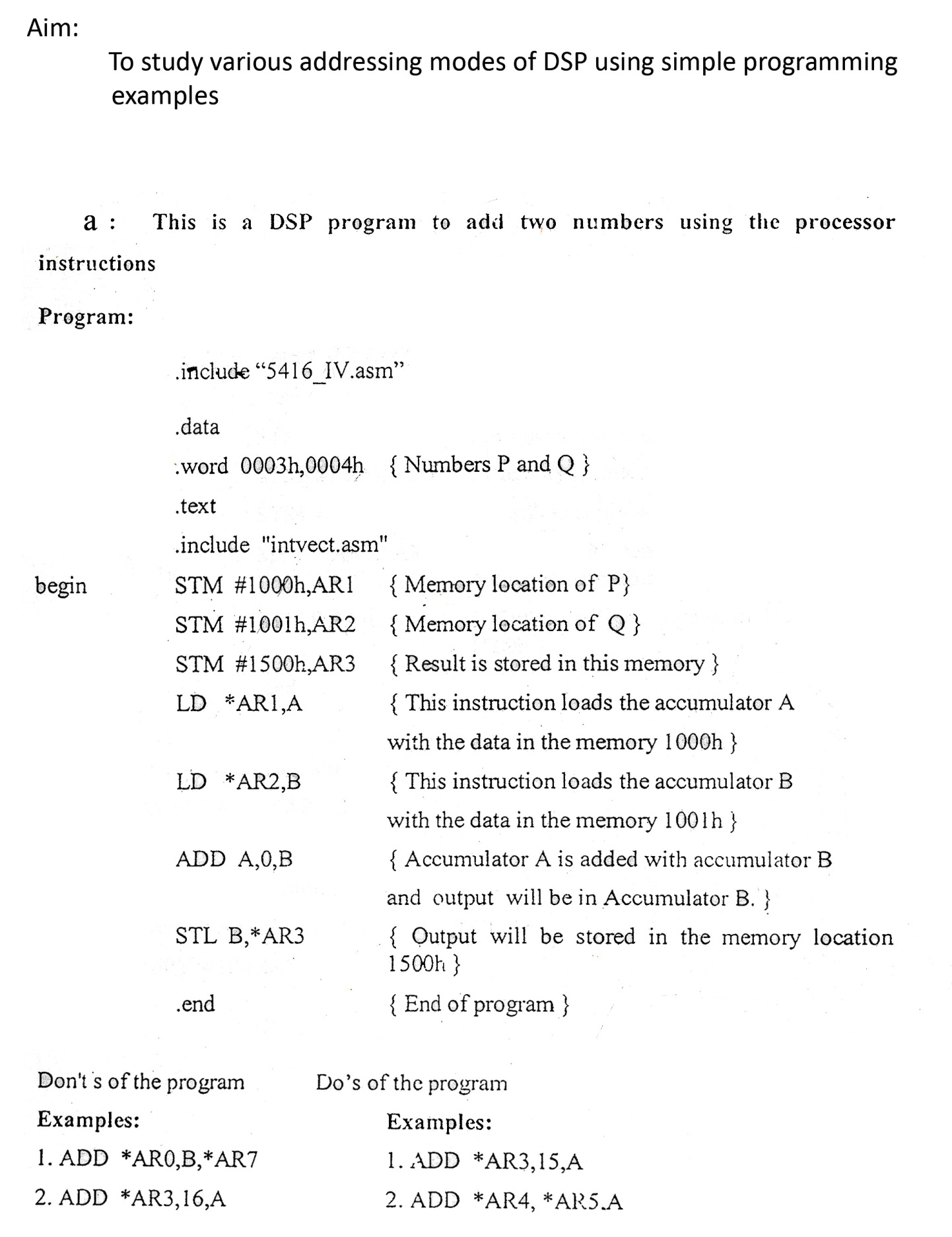
****

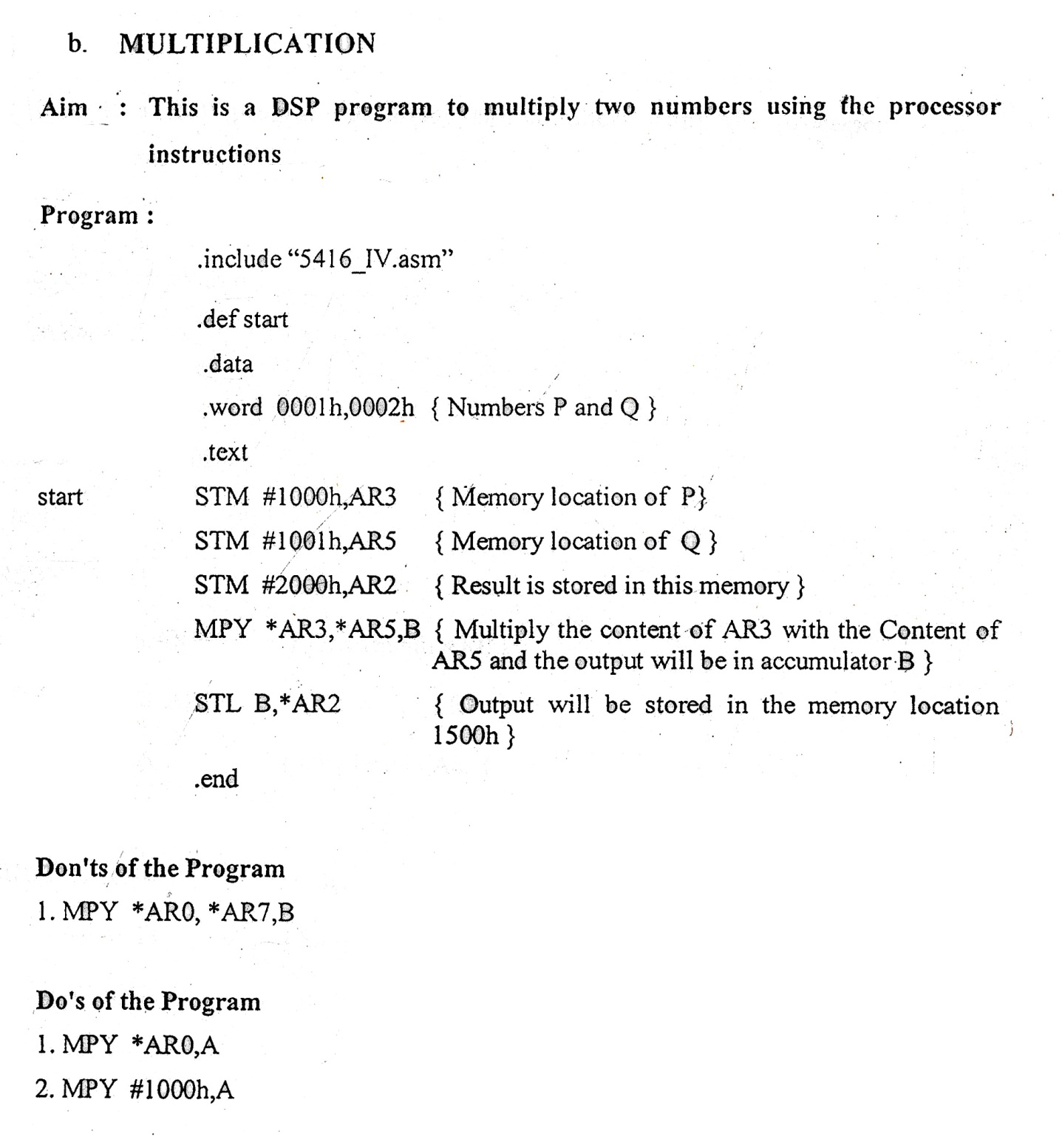
****

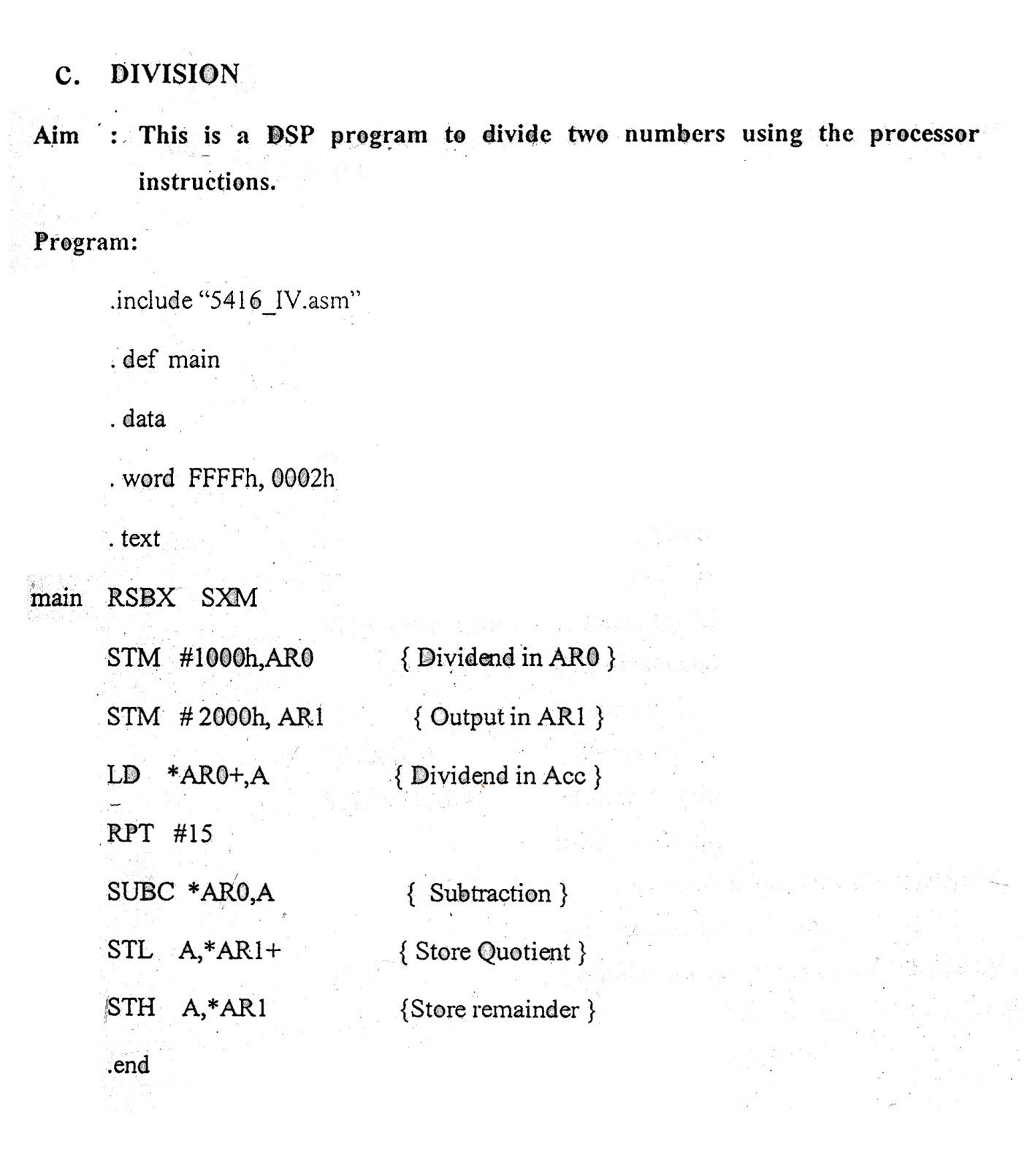
****

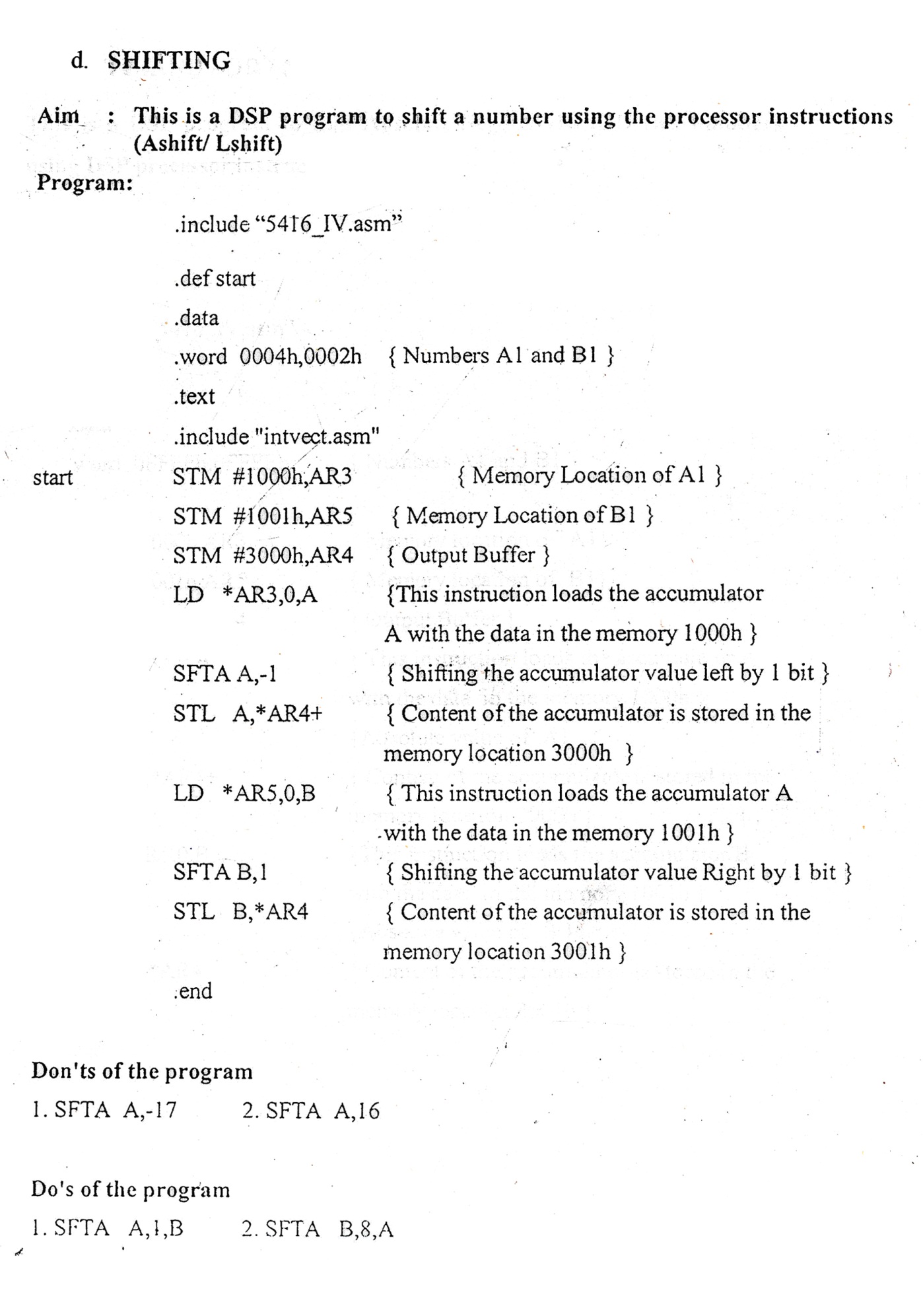
**Result :**

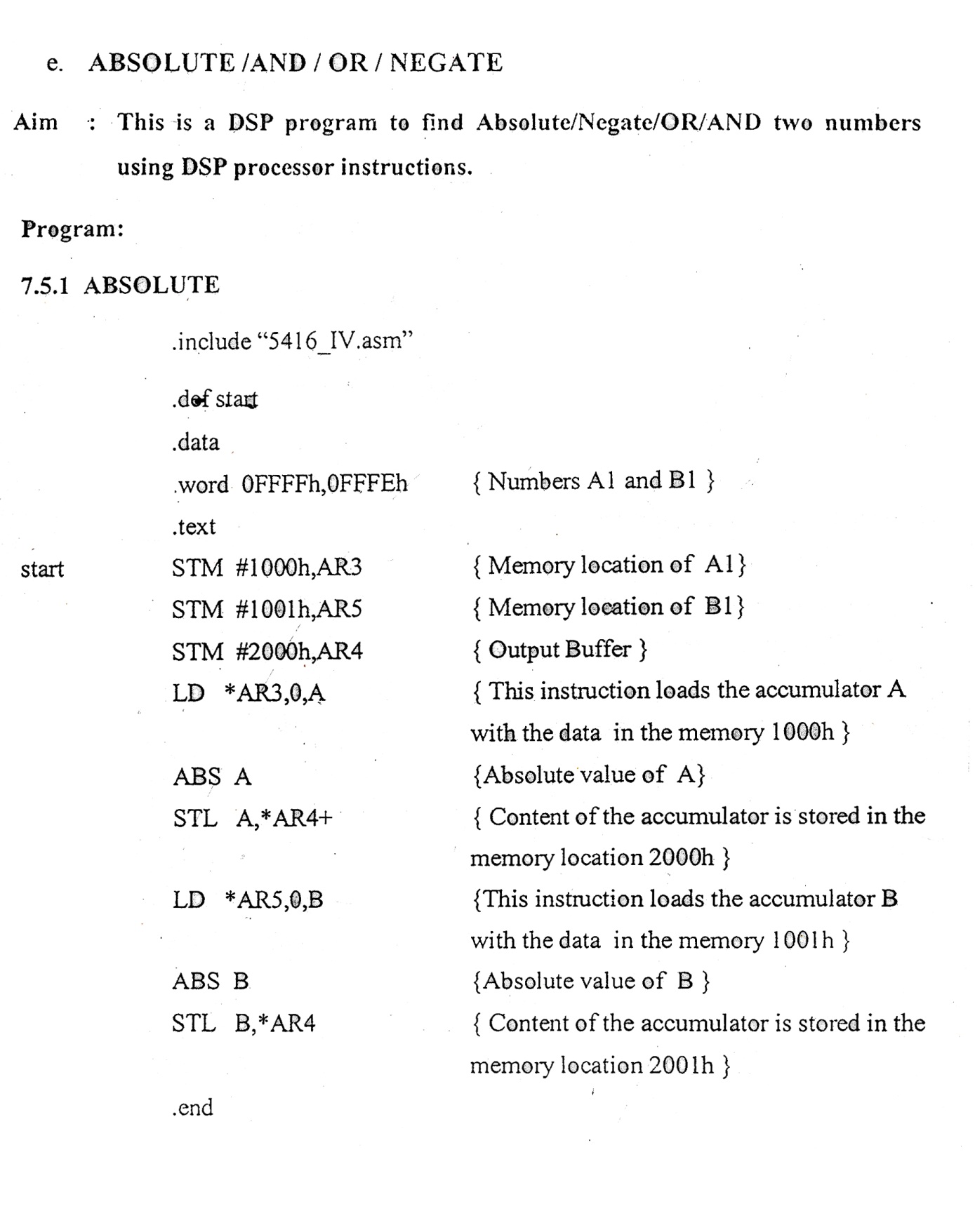
|  |  |  |
| --- | --- | --- |
| Ex. No : 15 | **STUDY OF VARIOUS ADDRESSING MODES OF DSP** | Date : |

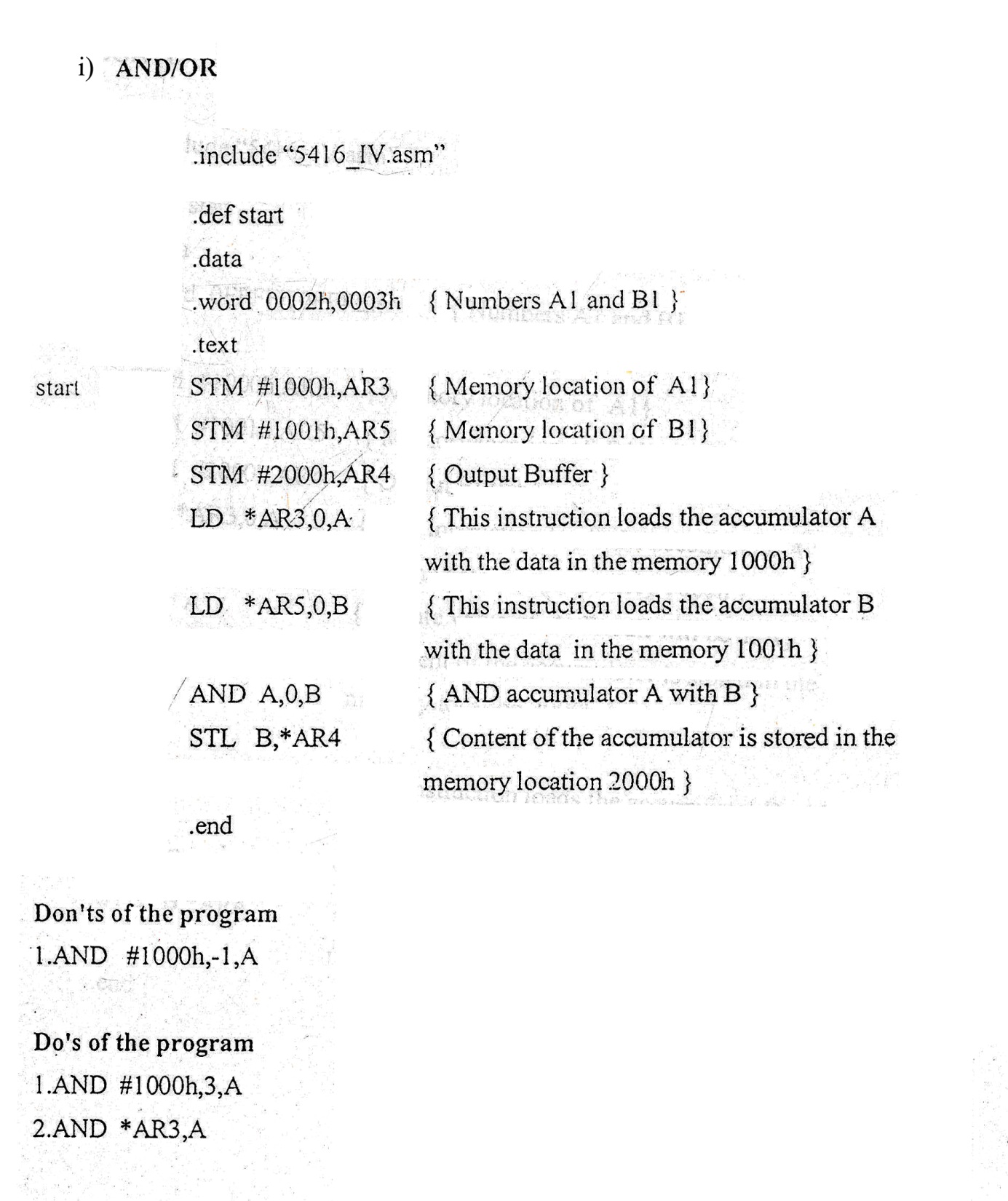
****

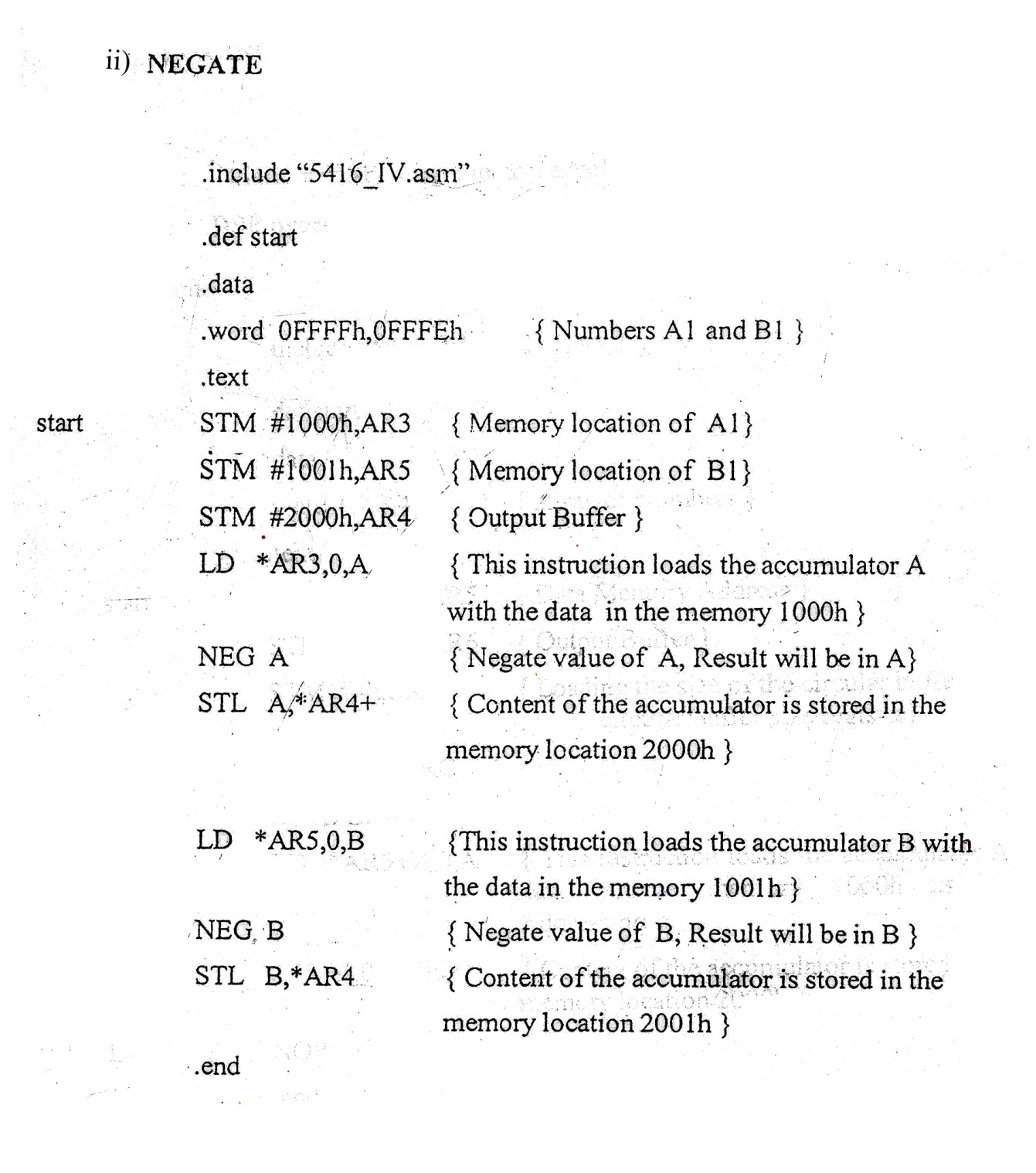
****

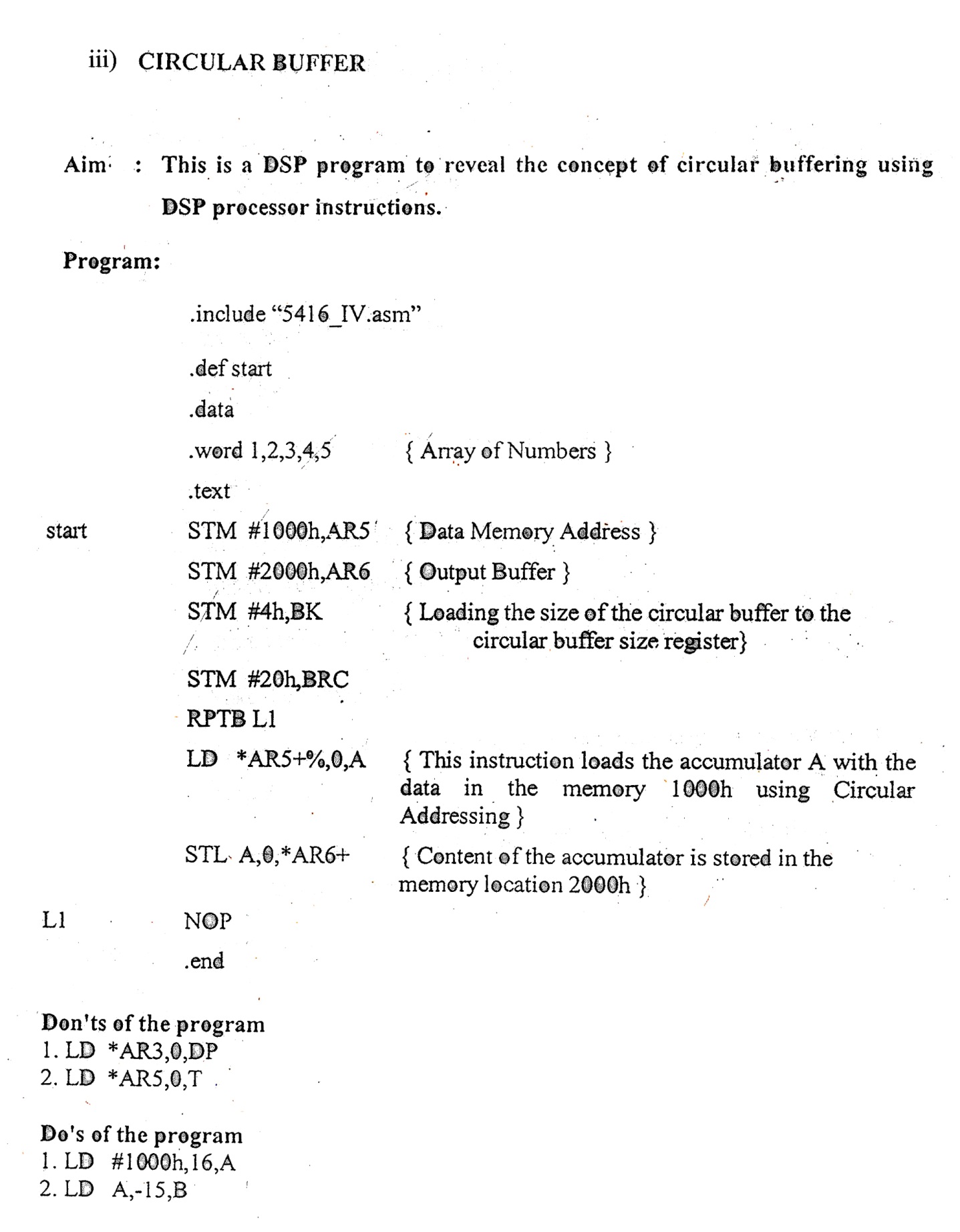
****

****

****

****

****

****

**Result :**