ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

1η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ακαδημαικό έτος 2015-2016

ΕΡΩΤΗΜΑ 2ο

ΚΩΔΙΚΟΠΟΙΗΣΗ PCM

ΣΙΜΑΚΗΣ ΠΑΝΑΓΙΩΤΗΣ

Α.Μ 5227 | simakis@ceid.upatras.gr

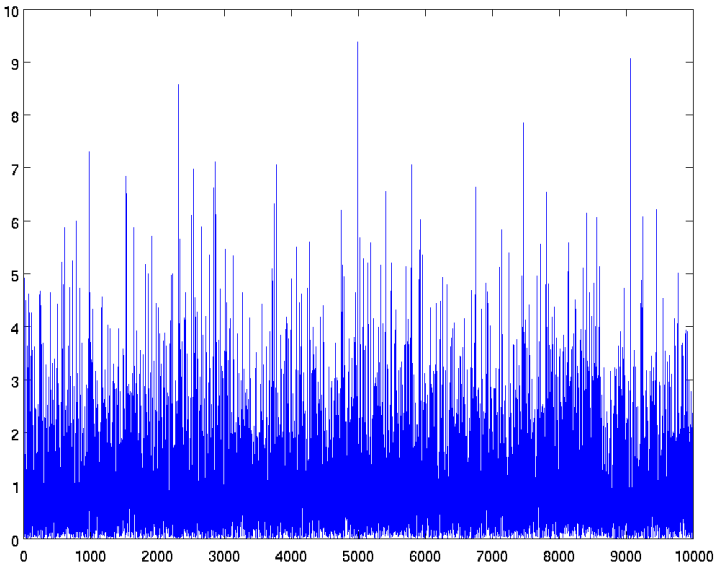
επι πτυχίο

1.

Δημιουργώ την πηγή Α με βάση την εκφώνηση:

```
t = (randn(10000,1)+sqrt(-1)*randn(10000,1))./sqrt(2);

x = abs(t).^2;
```

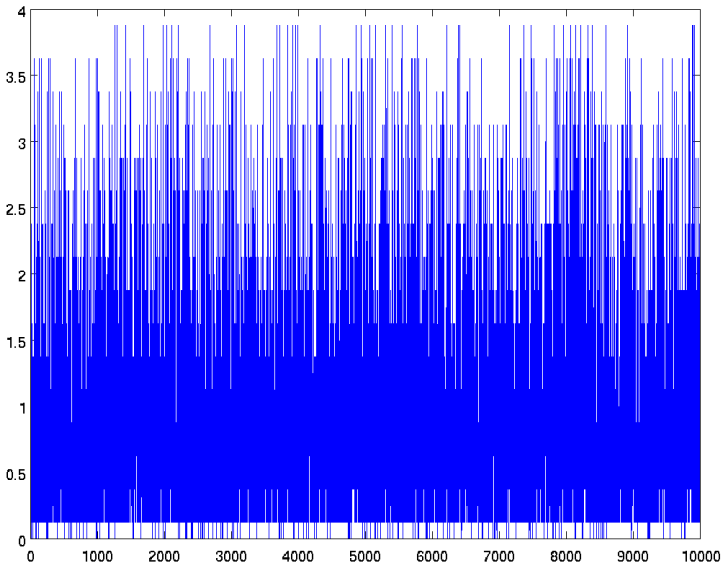
Το αρχικό σήμα φαίνεται παρακάτω:



Στην συνέχεια χρησιμοποιούμε τον ομοιόμορφο κφαντιστή που υλοποίησα για να κωδικοποιήσω την πηγή Α με min_value=0, max_value=4 και έχω:

- για N=4 bits

```
[xq, centers, p] = my_quantizer(x,4,0,4);
plot(xq);
```



στη συνέχεια με την συνάρτηση `sqnr()` υπολογίζω το SQNR στην έξοδο του κβαντιστή και την θεωρητική παραμόρφωση:

```
[my4_sqnr_exp, my4_sqnr_theor]=sqnr(x,xq,2)
```

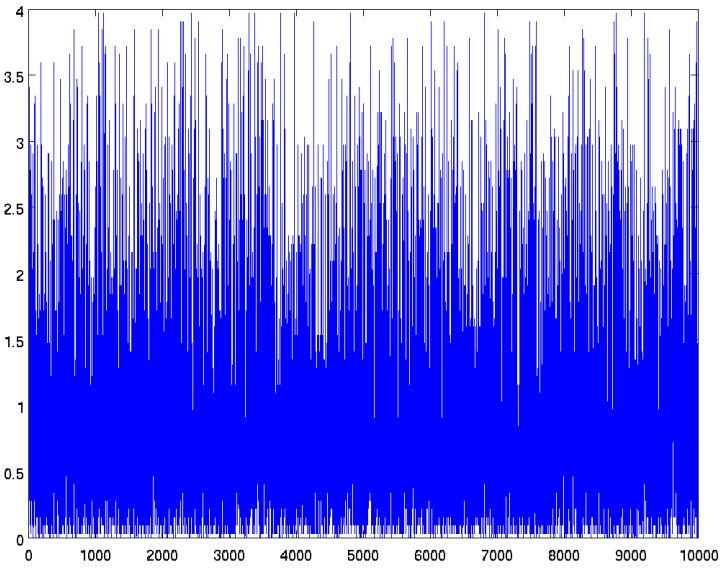
το οποίο μας επιστρέφει 6.3567 και 25.84 αντίστοιχα.

Η πιθανότητα να βρεθεί η είσοδος του κβαντιστή εκτός της δυναμικής περιοχής του είναι:

```
>> disp(p)
  Columns 1 through 9

    0.2216    0.1725    0.1387    0.1074    0.0784    0.0668    0.0495    0.0364    0.0278

  Columns 10 through 16

    0.0215    0.0187    0.0149    0.0099    0.0070    0.0057    0.0042
```

- για N = 6 bits

```
[y,fs,N] = wavread('speech.wav');
[xq, centers] = my_quantizer(y,6,0,4);
```



στη συνέχεια με την συνάρτηση `sqnr()` υπολογίζω το SQNR στην έξοδο του κβαντιστή και την θεωρητική παραμόρφωση:

```
[my6_sqnr_exp, my6_sqnr_theor]=sqnr(x,xq,6)
```

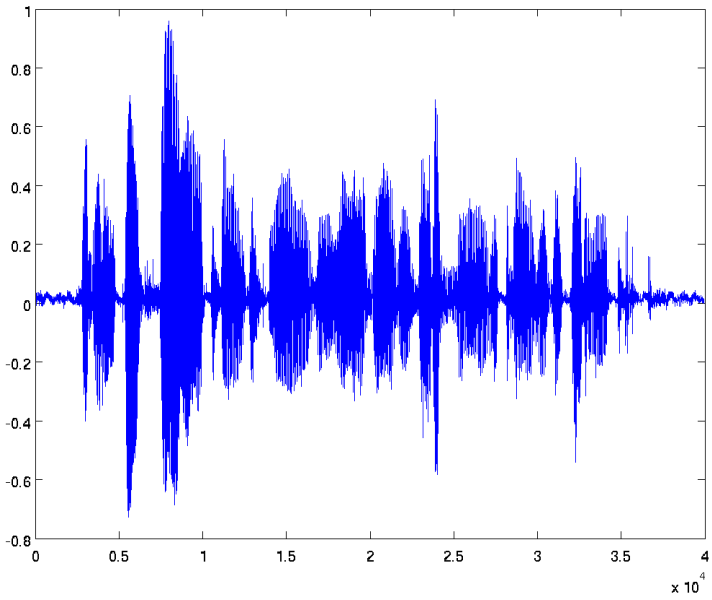
το οποίο μας επιστρέφει 6.4038 και 37.88 αντίστοιχα.

Η πιθανότητα να βρεθεί η είσοδος του κβαντιστή εκτός της δυναμικής περιοχής του είναι:

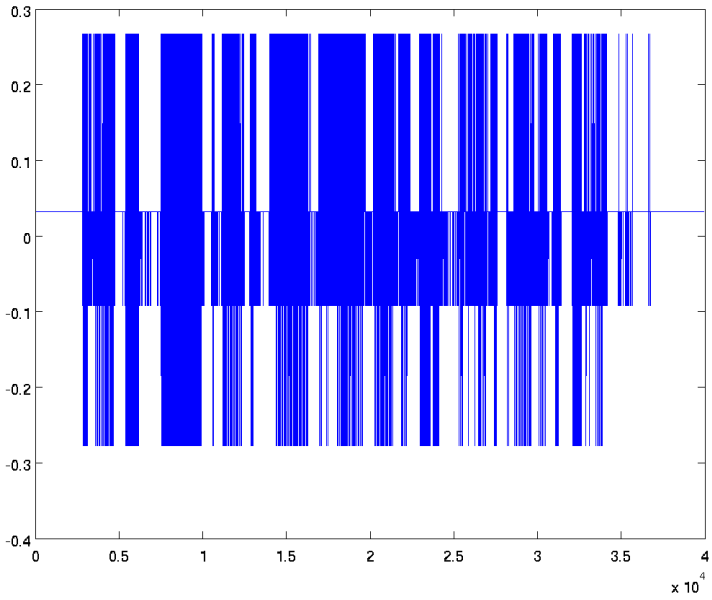
```
>> disp(p)
  Columns 1 through 9

    0.0592    0.0554    0.0536    0.0534    0.0484    0.0456    0.0403    0.0382    0.0347

  Columns 10 through 18

    0.0367    0.0370    0.0303    0.0304    0.0256    0.0269    0.0245    0.0226    0.0197

  Columns 19 through 27

    0.0206    0.0155    0.0204    0.0177    0.0150    0.0137    0.0138    0.0130    0.0104

  Columns 28 through 36

    0.0123    0.0104    0.0101    0.0084    0.0075    0.0082    0.0061    0.0074    0.0061

  Columns 37 through 45

    0.0069    0.0044    0.0058    0.0044    0.0050    0.0051    0.0046    0.0040    0.0042

  Columns 46 through 54

    0.0031    0.0034    0.0042    0.0032    0.0027    0.0021    0.0019    0.0021    0.0018

  Columns 55 through 63

    0.0016    0.0015    0.0010    0.0015    0.0015    0.0017    0.0011    0.0009    0.0011

  Column 64

    0.0011
```

2.

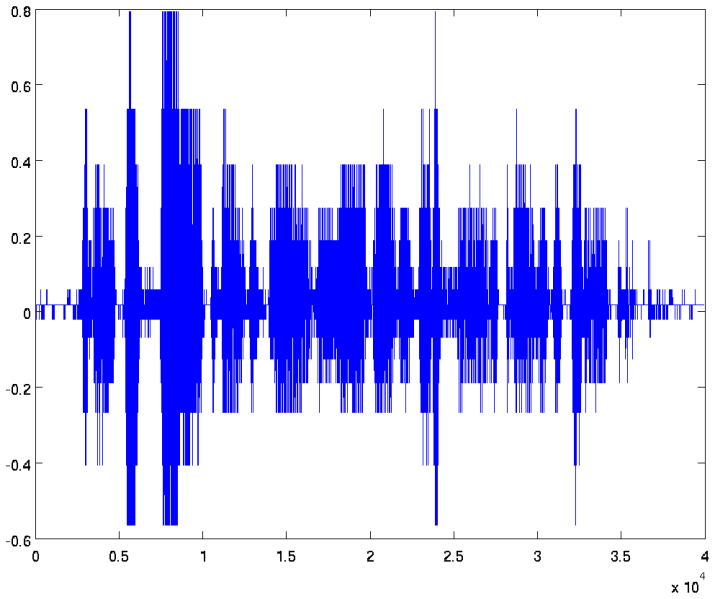
Το αρχικό σήμα της πηγής Β φαίνεται παρακάτω:



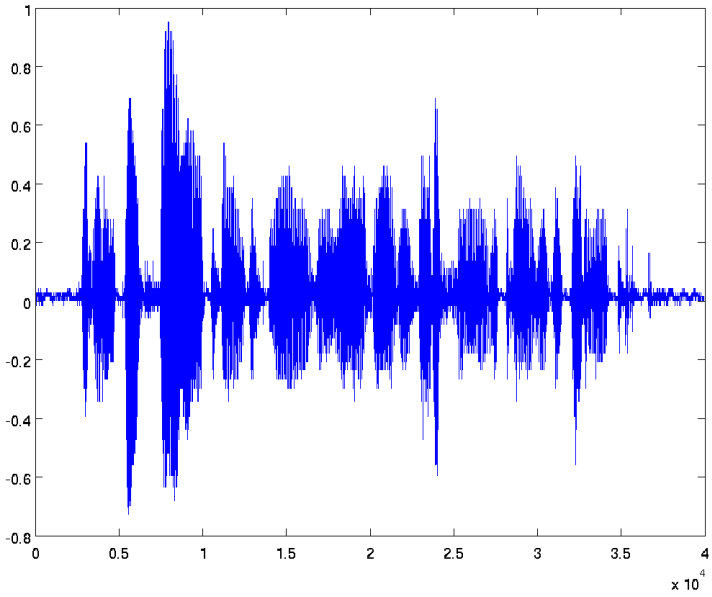
και στην συνέχεια ύστερα από την κωδικοποίηση με N=2, 4, 6 η εικόνα του σήματος είναι:



για N = 2


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.5
1
1.5
2
2.5
3
3.5
4
x 10^4

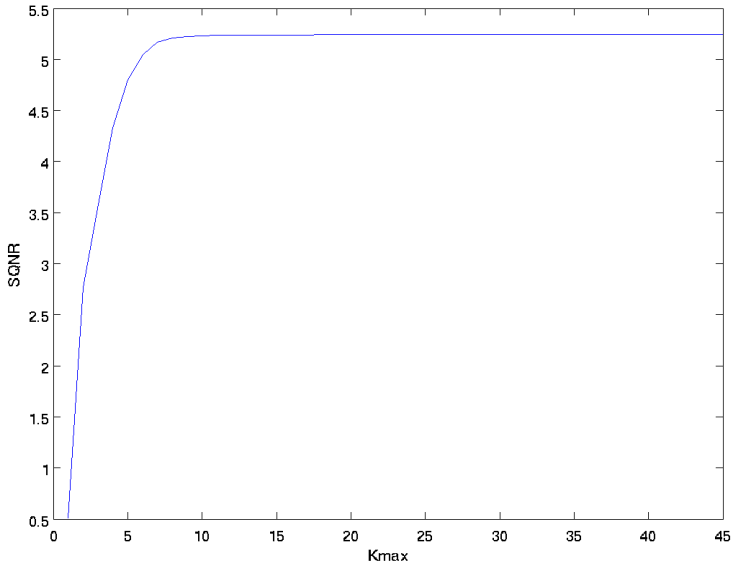
$N = 4$


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.5
1
1.5
2
2.5
3
3.5
4
x 10^4

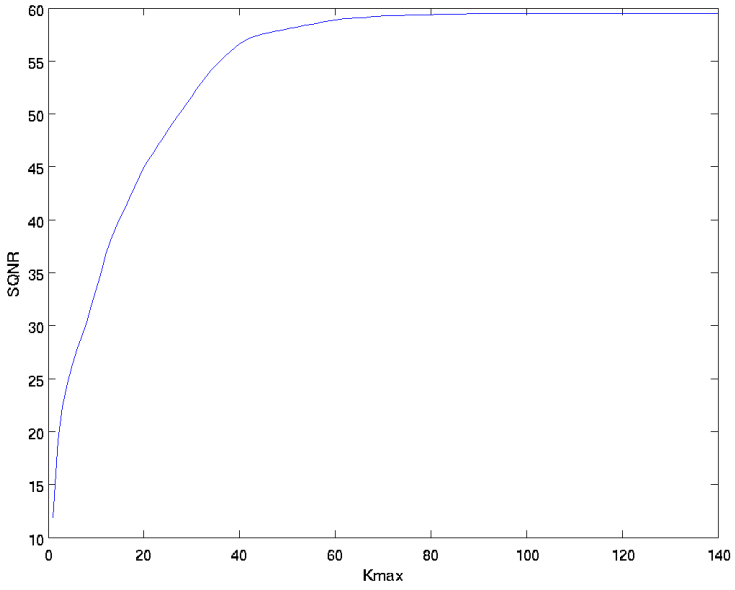
$N = 6$

a.

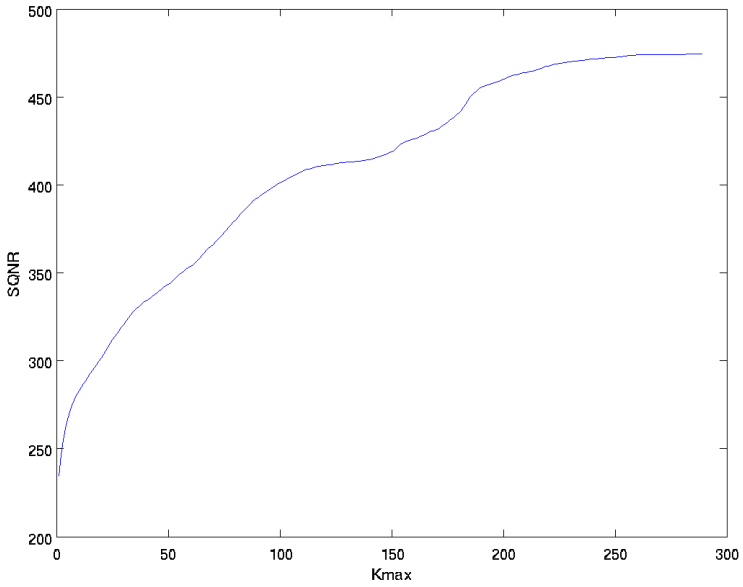
Ο ρυθμός με τον οποίο μεταβάλλεται το SQNR σε σχέση με τον αριθμό των επαναλήψεων:



$N = 2$



$N = 4$



$N = 6$

b.

Ο υπολογισμός των SQNR υλοποιείται με το script `erotima2b.m`

Αριθμός N bits	SQNR ομοιόμορφου κβαντιστή	SQNR Lloyd Max Κβαντιστή
2	-2.9001	7.1946
4	10.7555	17.7441
6	23.7050	26.7600

c.

Για να υπολογίσουμε πειραματικά την πιθανότητα εμφάνισης κάθε στάθμης του κβαντιστή αρκεί να υπολογίσουμς την πιθανότητα εμφάνισης του κάθε κέντρου κβάντισης. Η υλοποίηση γίνεται με την συνάρτηση `exp_freq()` και το script `erotima2c.m`

- για N = 2:

```
>> disp(freq2bits_exp)
    0.0594
    0.1943
    0.6321
```

```
    0.1142
```

και επαληθεύω:

```
>> sum(freq2bits_exp)

ans =

    1.0000
```

- για N = 4:

```
>> disp(freq4bits_exp)
    0.0036
    0.0076
    0.0240
    0.0452
    0.0627
    0.0790
    0.1205
    0.3549
    0.1106
    0.0801
    0.0538
    0.0352
    0.0147
    0.0065
    0.0015
```

και επαληθεύω:

```
>> sum(freq4bits_exp)

ans =

     1
```

- και τέλος για N = 6:

```
>> disp(freq6bits_exp)
    0.0000
    0.0001
    0.0002
```

```
0.0005
0.0006
0.0007
0.0013
0.0015
0.0019
0.0026
0.0032
0.0052
0.0084
0.0123
0.0165
0.0197
0.0222
0.0256
0.0269
0.0309
0.0360
0.0481
0.0553
0.1072
0.1542
0.1085
0.0586
0.0433
0.0356
0.0316
0.0274
0.0218
0.0201
0.0148
0.0140
0.0115
0.0093
0.0057
0.0041
0.0036
0.0025
0.0020
0.0015
0.0009
0.0008
```

```
    0.0004
    0.0003
    0.0002
    0.0002
    0.0001
    0.0001
    0.0002
    0.0001
    0.0002
    0.0001
```

και επαληθεύω:

```
>> sum(freq6bits_exp)


ans =


     1
```

Για τον υπολογισμό της εντροπίας από την θεωρία γνωρίζουμε οτι ο τύπος με τον οποίο υπολογίζεται είναι:

$$H = \sum p_i * \log_2 * (1/p_i)$$

όπου p είναι η πιθανότητα εμφάνησης κάθε στάθμης.

Για τον υπολογισμό της εντροπίας χρησιμοποιώ την συνάρτηση `entropy()` και τα αποτελέσματα που λαμβάνω είναι:

- για N = 2 έχω:

```
entropy2 = entropy(xq2lloyd);

>> disp(entropy2)

    1.4771
```

- για N = 4 έχω:

```
>> entropy4 = entropy(xq4lloyd);

>> disp(entropy4)

    3.0429
```

- για N = 6 έχω:

```
>> entropy6 = entropy(xq6lloyd);

>> disp(entropy6)

    4.4235
```

d.

Αρχικά υπολογίζω την αποδοτικότητα ή οποία είναι το πηλίκο $\frac{εντροπία}{N}$

Για N = 2, 4, 6 η αντίστοιχη αποδοτικότητα είναι:

```
>> entropy2/2


ans =


    0.7385
```

```
>> entropy4/4


ans =


    0.7607
```

```
>> entropy6/6


ans =


    0.7373
```

Κώδικες Matlab

my_quantizer.m

```
%% SYNARTHSH KWDIKOPOIHSH me OMOIOMORFO KVANTISTH

function [xq,centers, p] = my_quantizer(x,N,min_value,max_value)


quant_levels = 2^N; % epipeda kvantismou

vima = (max_value - min_value)/quant_levels;    % vima kvantismou

xq = zeros(length(x),1); % arxiko to dianysma me to shma eksodou me vash th
diastash toy dianismatos eisodou

centers = min_value + vima/2; % ypologizw to kentrou tou prwtou epipedou

p = zeros(1,quant_levels);  %arxiko to mhtrwo me tis pithanothtes emfanishs


% ypologizw ta kentra kvantishs

for i=1:1:quant_levels-1

    centers(i+1) = centers(i) + vima;

end


% kvantisw tis times toy shmatos

for i=1:1:length(x)

    for j=1:1:quant_levels

        if x(i)<=min_value + j*vima;

            xq(i) = centers(j);

            p(j) = p(j) + 1;

            break;

        end

    end

end


% ypologizw tis pithanothtes

p = p./length(x);

end
```

Lloyd_Max.m

```
%% SYNARTHSH KWDIKOPOIHSHS me ton algorithmo Lloyd Max
function [xq, centers, D, p] = Lloyd_Max(x,N,min_value,max_value)
kmax = 0;
quant_levels = 2^N; % ypologizw ta epipeda kvantismou me vash to N
xq = zeros(length(x),1) ; % arxikopoiw to kvantismeno shma me mhdenika
centers = zeros(quant_levels,1) ; % arxikopoiw ta kentra kvantismou me mhdenika
D(1) = 0;
Sqnr(1) = 0;
p = zeros(quant_levels,1) ; % arxikopoiw to mhtrwo pou tha ekxwrisw tis
pithanothtes
d = (max_value - min_value)/quant_levels ; % ypologizw to vima kvantismou
centers(1) = min_value + d/2 ; % ypologizw to prwto kentro


% ypologismos twn kentrwn kvantishs
for i =1:quant_levels-1
    centers(i+1) = centers(i) + d ; %ypologismos kentrwn perioxwn
end


% diadikasia kvantismou toy shmatos
while 1
    kmax = kmax + 1; % afksanw ton metrhth epanalipshs
    T(quant_levels+1) = max_value ; % arxikopoiw to panw orio
    T(1) = min_value;   % arxikopoiw to katw orio
    sum =zeros(quant_levels) ;
    counter=zeros(quant_levels) ; %arxikopoihsh counter emfanisewn
    for k=2:quant_levels
        T(k)=(centers(k-1)+centers(k))/2 ; %prosdiorismos orion
        %perioxon kvantisis
    end
    if (x(kmax)>=max_value) %Diadikasi kbantishs
    xq(kmax, 1) = 1;
    elseif (x(kmax)<min_value)
        xq(kmax, 1) = quant_levels ;
    else
        for i =1:length(x)
```

```
            for k=1:quant_levels
                if x(i)>T(k) && x(i)<=T(k+1)
                    p(k) = p(k) + 1;
                    xq(i, 1)=centers(k) ;
                    sum(k)=sum(k)+x(i) ;
                    counter(k)=counter(k)+1;
                    break;
                end
            end
        end
    end
    for k=1:quant_levels
        p(k) = p(k)/length(x) ;
        if (counter(k) > 0 )
            centers(k) = sum(k)/counter(k) ; %Ypologismos newn kentrwn
        end
    end
    % ypologizw thn paramorfosh se kathe epanalipsh
    D(kmax+1)=mean((x-xq).^2);
    Sqnr(kmax) = mean(x.^2)/ D(kmax+1);
    % kai an einai mikroterh apo to 10^-16 tote stamataei h diadikasia
    a = 10^-16;
    if (abs(D(kmax+1)-D(kmax))<a)
        break;
    end
end
```

sqnr.m

```
%% SYNARTHSH GIA TO YPOLOGISMOU THEORITIKOU KAI PEIRAMATIKOU SQNR


function [sqnr_exp, sqnr_theor] = sqnr(x, xq, N)
%prwta ypologizoume thn peiramatiki timh toy sqnr
sfalma = abs(xq-x);
sqnr_exp = 10*log10(sum(x.^2)/sum(sfalma.^2));
```

```
% kai sthn synexeia ypologizoume thn theoritikh timw toy sqnr
sqnr_theor = 1.76+6.02*N;


end
```

erotima2b.m

```
%% ypologismos twn sqnr gia omoiomofro kai lloyd max kvadisti
x = wavread('speech.wav');
xq2my = my_quantizer(x,2,-1,1);
sqnr2my = sqnr(x,xq2my,2);
xq2lloyd = Lloyd_Max(x,2,-1,1);
sqnr2lloyd = sqnr(x,xq2lloyd,2);


xq4my = my_quantizer(x,4,-1,1);
sqnr4my = sqnr(x,xq4my,4);
xq4lloyd = Lloyd_Max(x,4,-1,1);
sqnr4lloyd = sqnr(x,xq4lloyd,4);


xq6my = my_quantizer(x,6,-1,1);
sqnr6my = sqnr(x,xq6my,6);
xq6lloyd = Lloyd_Max(x,6,-1,1);
sqnr6lloyd = sqnr(x,xq6lloyd,6);
```

erotima2c.m

```
%% erotima 2c: peiramatikos ypologismos ypologismos ths pithanothtas emfanishs twn
kentrwn kvantishs
x = wavread('speech.wav');
[xq2lloyd,centers] = Lloyd_Max(x,2,-1,1);
[xq4lloyd,centers] = Lloyd_Max(x,4,-1,1);
[xq6lloyd,centers] = Lloyd_Max(x,6,-1,1);



freq2bits_exp = exp_freq(xq2lloyd);
freq4bits_exp = exp_freq(xq4lloyd);
freq6bits_exp = exp_freq(xq6lloyd);
```

entropy.m

```
%% synarthsh ypologismou ths entropias
function [entropy] = entropy(x)
p = exp_freq(x);
entropy = 0;
for i=1:length(p)
    if (p(i) ~= 0)
        entropy = entropy + p(i) * log2(1/p(i));
    end
end
```

exp_freq.m

```
%% me afth th synarthsh ypologizw thn pithanothta emfanishs toy kathe kentroy
kvantishs


function [ freq ] = exp_freq( xq )
uniq = unique(xq);
freq = [uniq,histc(xq(:),uniq)];
freq(:,2) = freq(:,2)./length(xq);
freq(:,1) = [];
disp(freq);


end
```