Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

1^{Ο} ΣΕΤ ΕΡΓΑΣΤΗΡΙΑΚΩΝ ΑΣΚΗΣΕΩΝ

ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ ΠΟΤΑΜΙΑΝΟΣ
Α.Μ. 1084537
up1084537@ac.upatras.gr

Πάτρα, ΙΑΝ.2024

Μέρος Α’ (Κώδικας)

1.

a. Από τη θεωρία του μαθήματος, γνωρίζουμε ότι στη Θεωρία Πληροφορίας, η πηγή πληροφορίας μπορεί να αναλυθεί σε διακριτά σύμβολα με συγκεκριμένες πιθανότητες εμφάνισης. Στην περίπτωση της εικόνας, τα σύμβολα αυτά αντιπροσωπεύονται από τις διαφορετικές τιμές των pixels.

Στον παραπάνω κώδικα MATLAB, πρώτα γίνεται η ανάγνωση της εικόνας "parrot.png". Στη συνέχεια, υπολογίζονται οι συχνότητες εμφάνισης κάθε μοναδικής τιμής pixel στην εικόνα. Αυτό επιτυγχάνεται μέσω της συνάρτησης unique, η οποία εντοπίζει τις μοναδικές τιμές των pixels, και της accumarray, η οποία υπολογίζει τη συχνότητα εμφάνισης κάθε τιμής.

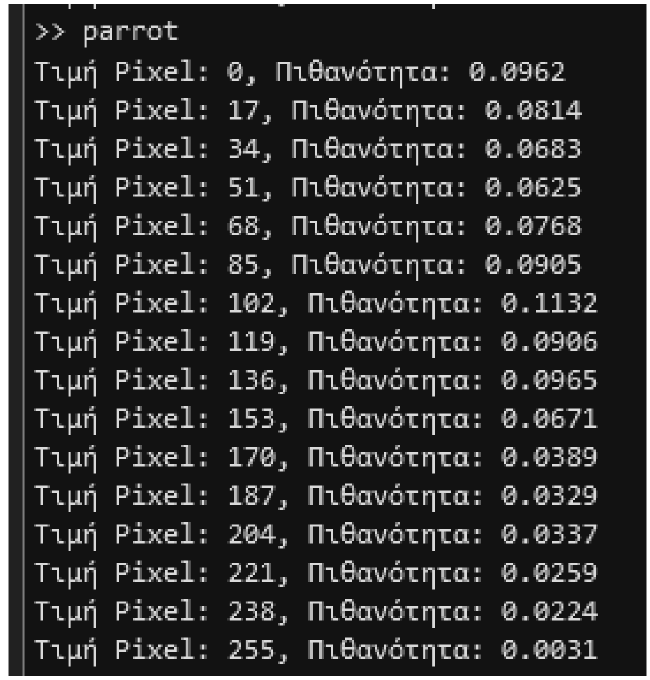
Η γραμμή κώδικα [symbols, ~, idx] = unique(I); στο MATLAB εκτελεί τρεις λειτουργίες:

i. symbols: Επιστρέφει τις μοναδικές τιμές των pixel της εικόνας I (δηλαδή της «parrot.png»). Αυτές οι τιμές αντιπροσωπεύουν τα διακριτά σύμβολα της πηγής πληροφορίας.
ii. ~: Είναι ένας τρόπος να αγνοηθούν επιστρεφόμενες τιμές που δεν χρειάζονται. Στην περίπτωση αυτή, δεν ενδιαφερόμαστε για το δεύτερο εξαγόμενο αποτέλεσμα της συνάρτησης unique.
iii. idx: Επιστρέφει ένα διάνυσμα που αντιστοιχεί στη θέση κάθε στοιχείου του I στον ταξινομημένο πίνακα symbols.

Τέλος, οι πιθανότητες για κάθε τιμή pixel υπολογίζονται διαιρώντας τις συχνότητες με τον συνολικό αριθμό των pixels στην εικόνα (probabilities = counts / numel(I);) .

ΑΠΟΤΕΛΕΣΜΑΤΑ:

Διακριτή Τιμή (Pixel Value)	Πιθανότητα Εμφάνισης
0	9.62%
17	8.14%
34	6.83%
51	6.25%
68	7.68%
85	9.05%
102	11.32%
119	9.06%
136	9.65%
153	6.71%
170	3.89%
187	3.29%
204	3.37%
221	2.59%
238	2.24%
255	0.31%



```
>> parrot
Τιμή Pixel: 0, Πιθανότητα: 0.0962
Τιμή Pixel: 17, Πιθανότητα: 0.0814
Τιμή Pixel: 34, Πιθανότητα: 0.0683
Τιμή Pixel: 51, Πιθανότητα: 0.0625
Τιμή Pixel: 68, Πιθανότητα: 0.0768
Τιμή Pixel: 85, Πιθανότητα: 0.0905
Τιμή Pixel: 102, Πιθανότητα: 0.1132
Τιμή Pixel: 119, Πιθανότητα: 0.0906
Τιμή Pixel: 136, Πιθανότητα: 0.0965
Τιμή Pixel: 153, Πιθανότητα: 0.0671
Τιμή Pixel: 170, Πιθανότητα: 0.0389
Τιμή Pixel: 187, Πιθανότητα: 0.0329
Τιμή Pixel: 204, Πιθανότητα: 0.0337
Τιμή Pixel: 221, Πιθανότητα: 0.0259
Τιμή Pixel: 238, Πιθανότητα: 0.0224
Τιμή Pixel: 255, Πιθανότητα: 0.0031
```

b.i. Για τον υπολογισμό της εντροπίας, πρώτα θα διαβάσουμε την εικόνα (έχει γίνει ήδη στο προηγούμενο βήμα), θα την μετατρέψουμε σε μονοδιάστατο πίνακα, και μετά θα υπολογίσουμε τις πιθανότητες εμφάνισης κάθε μοναδικής τιμής pixel. Η εντροπία θα υπολογιστεί χρησιμοποιώντας τον τύπο:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

όπου H(X) είναι η εντροπία της πηγής, p(x_i) είναι η πιθανότητα εμφάνισης της τιμής x_i, και η άθροιση γίνεται πάνω σε όλες τις μοναδικές τιμές των pixel.

ΑΠΟΤΕΛΕΣΜΑΤΑ: H(X) = 3.7831

Name	Value	Size	Class
counts	16x1 double	16x1	double
entropy	3.7831	1x1	double
i	16	1x1	double
idx	30000x1 double	30000x1	double
img	200x150 uint8	200x150	uint8
pixels	30000x1 uint8	30000x1	uint8
probabilities	16x1 double	16x1	double
symbols	16x1 uint8	16x1	uint8
unique_values	16x1 uint8	16x1	uint8

ii. Η διαδικασία υπολογισμού του μέσου μήκους κώδικα είναι η εξής:

<u>ΒΗΜΑ1:</u> Δημιουργία του Huffman δενδρου με την χρήση του συνόνημου αλγορίθμου με βάση τις πιθανότητες εμφάνισης κάθε μοναδικής τιμής.

<u>ΒΗΜΑ 2:</u> Υπολογισμός του μήκους κάθε κωδικού Huffman.

<u>ΒΗΜΑ 3:</u> Υπολογισμός του μέσου μήκους του κώδικα, που είναι το άθροισμα των πιθανοτήτων κάθε συμβόλου επί το μήκος του κωδικού του.

<u>ΑΠΟΤΕΛΕΣΜΑ:</u> Το δένδρο Huffman του Βήματος 1 καθώς και το μέσο μήκος κώδικα του Βήματος 3.

```
Σύμβολο: 0, Κωδικός Huffman: 1  1  0
Σύμβολο: 17, Κωδικός Huffman: 0  0  1  0
Σύμβολο: 34, Κωδικός Huffman: 0  1  0  0
Σύμβολο: 51, Κωδικός Huffman: 0  1  1  1
Σύμβολο: 68, Κωδικός Huffman: 0  0  1  1
Σύμβολο: 85, Κωδικός Huffman: 0  0  0  0
Σύμβολο: 102, Κωδικός Huffman: 1  0  0
Σύμβολο: 119, Κωδικός Huffman: 1  1  1
Σύμβολο: 136, Κωδικός Huffman: 1  0  1
Σύμβολο: 153, Κωδικός Huffman: 0  1  0  1
Σύμβολο: 170, Κωδικός Huffman: 0  0  0  1  1
Σύμβολο: 187, Κωδικός Huffman: 0  1  1  0  1
Σύμβολο: 204, Κωδικός Huffman: 0  1  1  0  0
Σύμβολο: 221, Κωδικός Huffman: 0  0  0  1  0  0
Σύμβολο: 238, Κωδικός Huffman: 0  0  0  1  0  1  0
Σύμβολο: 255, Κωδικός Huffman: 0  0  0  1  0  1  1
Μέσο μήκος κωδικού Huffman: 3.8374
>> |
```

iii. Από την θεωρία μας γνωρίζουμε πως η αποδοτικότητα ενός κώδικα ορίζεται ως:

$$\eta = \frac{H(X)}{L} = \frac{Εντροπία\ Πηγής}{Μέσο\ Μήκος\ Κώδικα} \leq 1$$

Άρα 3.7831 / 3.8374 ≈ 0.9859

ΣΧΟΛΙΑ ΓΙΑ Β:

Εντροπία της Πηγής H(X): Η εντροπία της πηγής μας είναι 3.7831 bits ανά σύμβολο. Αυτό δηλώνει ότι, κατά μέσο όρο, κάθε σύμβολο της πηγής παρέχει περίπου 3.7831 bits πληροφορίας. Η εντροπία είναι ένα μέτρο της μέσης αβεβαιότητας ή της ποικιλομορφίας της πηγής και εδώ φαίνεται να είναι αρκετά υψηλή, δείχνοντας ένα μεγάλο εύρος τιμών στα δεδομένα μας(επομένως φέρει πολύ πληροφορία και χρειάζεται περισσότερα bit για την κωδικοποίησή της).

Μέσο Μήκος του Κώδικα Huffman (L): Το μέσο μήκος του κώδικα Huffman είναι 3.8374 bits ανά σύμβολο. Αυτό σημαίνει ότι ο κώδικας Huffman, κατά μέσο όρο, απαιτεί περίπου 3.8374 bits για να κωδικοποιήσει κάθε σύμβολο της πηγής. Το μέγεθος αυτό είναι σχετικά κοντά στην εντροπία της πηγής, δείχνοντας μια αρκετά αποδοτική κωδικοποίηση.

Αποδοτικότητα του Κώδικα (Efficiency): Η αποδοτικότητα του κώδικα μας είναι 0.9859, η οποία είναι πολύ κοντά στην ιδανική τιμή 1. Αυτό δηλώνει ότι ο κώδικας Huffman που χρησιμοποιήσαμε είναι πολύ αποδοτικός στην κωδικοποίηση της πηγής μας. Με άλλα λόγια, ο κώδικας μας είναι σχεδόν τόσο συμπαγής όσο θα ήταν ένας ιδανικός κώδικας με μήκος ίσο με την εντροπία της πηγής.

Συνολικά, αυτά τα αποτελέσματα δείχνουν ότι η τεχνική κωδικοποίησης Huffman που εφαρμόσαμε είναι αρκετά αποτελεσματική για τη δεδομένη πηγή, καθώς επιτυγχάνει ένα υψηλό επίπεδο συμπίεσης με μικρή απώλεια πληροφορίας.

2.

a. Προσαρμόζω τον κώδικα της 1ης άσκησης έτσι ώστε να δημιουργούμε ζεύγη διαδοχικών τιμών pixel. Κάθε ζεύγος αποτελεί ένα "σύμβολο" δεύτερης τάξης επέκτασης. Για παράδειγμα, αν έχουμε μια σειρά τιμών pixel [p1, p2, p3, ...], τα ζεύγη θα είναι (p1, p2), (p2, p3), κ.ο.κ. Έπειτα υπολογίζουμε τη συχνότητα εμφάνισης κάθε ζεύγους στην εικόνα. Αυτό γίνεται μετρώντας πόσες φορές εμφανίζεται κάθε ζεύγος και υπολογίζοντας την πιθανότητα εμφάνισης κάθε ζεύγους με βάση το συνολικό αριθμό των ζευγών(=202).

ΑΠΟΤΕΛΕΣΜΑΤΑ:

```
>> parrot2
Ζεύγος Pixel: 0 0, Πιθανότητα: 0.0810
Ζεύγος Pixel: 0 17, Πιθανότητα: 0.0129
Ζεύγος Pixel: 0 34, Πιθανότητα: 0.0013
Ζεύγος Pixel: 0 51, Πιθανότητα: 0.0005
Ζεύγος Pixel: 0 68, Πιθανότητα: 0.0003
Ζεύγος Pixel: 0 85, Πιθανότητα: 0.0001
Ζεύγος Pixel: 0 102, Πιθανότητα: 0.0001
Ζεύγος Pixel: 17 0, Πιθανότητα: 0.0123
Ζεύγος Pixel: 17 17, Πιθανότητα: 0.0502
Ζεύγος Pixel: 17 34, Πιθανότητα: 0.0133
Ζεύγος Pixel: 17 51, Πιθανότητα: 0.0026
Ζεύγος Pixel: 17 68, Πιθανότητα: 0.0013
Ζεύγος Pixel: 17 85, Πιθανότητα: 0.0005
Ζεύγος Pixel: 17 102, Πιθανότητα: 0.0004
Ζεύγος Pixel: 17 119, Πιθανότητα: 0.0002
Ζεύγος Pixel: 17 136, Πιθανότητα: 0.0003
Ζεύγος Pixel: 17 153, Πιθανότητα: 0.0001
Ζεύγος Pixel: 17 170, Πιθανότητα: 0.0001
Ζεύγος Pixel: 17 187, Πιθανότητα: 0.0000
Ζεύγος Pixel: 34 0, Πιθανότητα: 0.0011
Ζεύγος Pixel: 34 17, Πιθανότητα: 0.0129
Ζεύγος Pixel: 34 34, Πιθανότητα: 0.0360
Ζεύγος Pixel: 34 51, Πιθανότητα: 0.0119
Ζεύγος Pixel: 34 68, Πιθανότητα: 0.0026
Ζεύγος Pixel: 34 85, Πιθανότητα: 0.0012
Ζεύγος Pixel: 34 102, Πιθανότητα: 0.0010
Ζεύγος Pixel: 34 119, Πιθανότητα: 0.0004
Ζεύγος Pixel: 34 136, Πιθανότητα: 0.0004
Ζεύγος Pixel: 34 153, Πιθανότητα: 0.0003
Ζεύγος Pixel: 34 170, Πιθανότητα: 0.0002
Ζεύγος Pixel: 34 187, Πιθανότητα: 0.0002
Ζεύγος Pixel: 34 204, Πιθανότητα: 0.0001
Ζεύγος Pixel: 51 0, Πιθανότητα: 0.0005
Ζεύγος Pixel: 51 17, Πιθανότητα: 0.0031
Ζεύγος Pixel: 51 34, Πιθανότητα: 0.0124
Ζεύγος Pixel: 51 51, Πιθανότητα: 0.0289
Ζεύγος Pixel: 51 68, Πιθανότητα: 0.0121
Ζεύγος Pixel: 51 85, Πιθανότητα: 0.0024
Ζεύγος Pixel: 51 102, Πιθανότητα: 0.0014
Ζεύγος Pixel: 51 119, Πιθανότητα: 0.0007
Ζεύγος Pixel: 51 136, Πιθανότητα: 0.0002
```

```
Ζεύγος Pixel: 51 153, Πιθανότητα: 0.0003
Ζεύγος Pixel: 51 170, Πιθανότητα: 0.0002
Ζεύγος Pixel: 51 187, Πιθανότητα: 0.0001
Ζεύγος Pixel: 51 204, Πιθανότητα: 0.0001
Ζεύγος Pixel: 51 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 51 238, Πιθανότητα: 0.0000
Ζεύγος Pixel: 68 0, Πιθανότητα: 0.0002
Ζεύγος Pixel: 68 17, Πιθανότητα: 0.0010
Ζεύγος Pixel: 68 34, Πιθανότητα: 0.0026
Ζεύγος Pixel: 68 51, Πιθανότητα: 0.0119
Ζεύγος Pixel: 68 68, Πιθανότητα: 0.0417
Ζεύγος Pixel: 68 85, Πιθανότητα: 0.0135
Ζεύγος Pixel: 68 102, Πιθανότητα: 0.0026
Ζεύγος Pixel: 68 119, Πιθανότητα: 0.0014
Ζεύγος Pixel: 68 136, Πιθανότητα: 0.0007
Ζεύγος Pixel: 68 153, Πιθανότητα: 0.0005
Ζεύγος Pixel: 68 170, Πιθανότητα: 0.0002
Ζεύγος Pixel: 68 187, Πιθανότητα: 0.0002
Ζεύγος Pixel: 68 204, Πιθανότητα: 0.0001
Ζεύγος Pixel: 68 221, Πιθανότητα: 0.0002
Ζεύγος Pixel: 68 238, Πιθανότητα: 0.0001
Ζεύγος Pixel: 85 0, Πιθανότητα: 0.0003
Ζεύγος Pixel: 85 17, Πιθανότητα: 0.0005
Ζεύγος Pixel: 85 34, Πιθανότητα: 0.0017
Ζεύγος Pixel: 85 51, Πιθανότητα: 0.0042
Ζεύγος Pixel: 85 68, Πιθανότητα: 0.0130
Ζεύγος Pixel: 85 85, Πιθανότητα: 0.0493
Ζεύγος Pixel: 85 102, Πιθανότητα: 0.0165
Ζεύγος Pixel: 85 119, Πιθανότητα: 0.0024
Ζεύγος Pixel: 85 136, Πιθανότητα: 0.0013
Ζεύγος Pixel: 85 153, Πιθανότητα: 0.0004
Ζεύγος Pixel: 85 170, Πιθανότητα: 0.0004
Ζεύγος Pixel: 85 187, Πιθανότητα: 0.0002
Ζεύγος Pixel: 85 204, Πιθανότητα: 0.0001
Ζεύγος Pixel: 85 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 85 238, Πιθανότητα: 0.0001
Ζεύγος Pixel: 102 0, Πιθανότητα: 0.0001
Ζεύγος Pixel: 102 17, Πιθανότητα: 0.0002
Ζεύγος Pixel: 102 34, Πιθανότητα: 0.0004
Ζεύγος Pixel: 102 51, Πιθανότητα: 0.0017
Ζεύγος Pixel: 102 68, Πιθανότητα: 0.0036
Ζεύγος Pixel: 102 85, Πιθανότητα: 0.0165
Ζεύγος Pixel: 102 102, Πιθανότητα: 0.0698
```

```
Ζεύγος Pixel: 102 119, Πιθανότητα: 0.0156
Ζεύγος Pixel: 102 136, Πιθανότητα: 0.0026
Ζεύγος Pixel: 102 153, Πιθανότητα: 0.0011
Ζεύγος Pixel: 102 170, Πιθανότητα: 0.0006
Ζεύγος Pixel: 102 187, Πιθανότητα: 0.0006
Ζεύγος Pixel: 102 204, Πιθανότητα: 0.0002
Ζεύγος Pixel: 102 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 102 238, Πιθανότητα: 0.0001
Ζεύγος Pixel: 102 255, Πιθανότητα: 0.0000
Ζεύγος Pixel: 119 0, Πιθανότητα: 0.0001
Ζεύγος Pixel: 119 17, Πιθανότητα: 0.0003
Ζεύγος Pixel: 119 34, Πιθανότητα: 0.0002
Ζεύγος Pixel: 119 51, Πιθανότητα: 0.0005
Ζεύγος Pixel: 119 68, Πιθανότητα: 0.0018
Ζεύγος Pixel: 119 85, Πιθανότητα: 0.0047
Ζεύγος Pixel: 119 102, Πιθανότητα: 0.0157
Ζεύγος Pixel: 119 119, Πιθανότητα: 0.0499
Ζεύγος Pixel: 119 136, Πιθανότητα: 0.0133
Ζεύγος Pixel: 119 153, Πιθανότητα: 0.0024
Ζεύγος Pixel: 119 170, Πιθανότητα: 0.0006
Ζεύγος Pixel: 119 187, Πιθανότητα: 0.0006
Ζεύγος Pixel: 119 204, Πιθανότητα: 0.0002
Ζεύγος Pixel: 119 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 119 238, Πιθανότητα: 0.0001
Ζεύγος Pixel: 119 255, Πιθανότητα: 0.0000
Ζεύγος Pixel: 136 0, Πιθανότητα: 0.0005
Ζεύγος Pixel: 136 17, Πιθανότητα: 0.0001
Ζεύγος Pixel: 136 34, Πιθανότητα: 0.0003
Ζεύγος Pixel: 136 51, Πιθανότητα: 0.0002
Ζεύγος Pixel: 136 68, Πιθανότητα: 0.0004
Ζεύγος Pixel: 136 85, Πιθανότητα: 0.0016
Ζεύγος Pixel: 136 102, Πιθανότητα: 0.0047
Ζεύγος Pixel: 136 119, Πιθανότητα: 0.0155
Ζεύγος Pixel: 136 136, Πιθανότητα: 0.0561
Ζεύγος Pixel: 136 153, Πιθανότητα: 0.0135
Ζεύγος Pixel: 136 170, Πιθανότητα: 0.0020
Ζεύγος Pixel: 136 187, Πιθανότητα: 0.0010
Ζεύγος Pixel: 136 204, Πιθανότητα: 0.0003
Ζεύγος Pixel: 136 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 136 238, Πιθανότητα: 0.0001
Ζεύγος Pixel: 136 255, Πιθανότητα: 0.0002
Ζεύγος Pixel: 153 0, Πιθανότητα: 0.0001
Ζεύγος Pixel: 153 17, Πιθανότητα: 0.0000
```

```
Ζεύγος Pixel: 153 34, Πιθανότητα: 0.0001
Ζεύγος Pixel: 153 51, Πιθανότητα: 0.0001
Ζεύγος Pixel: 153 68, Πιθανότητα: 0.0001
Ζεύγος Pixel: 153 85, Πιθανότητα: 0.0004
Ζεύγος Pixel: 153 102, Πιθανότητα: 0.0006
Ζεύγος Pixel: 153 119, Πιθανότητα: 0.0035
Ζεύγος Pixel: 153 136, Πιθανότητα: 0.0183
Ζεύγος Pixel: 153 153, Πιθανότητα: 0.0342
Ζεύγος Pixel: 153 170, Πιθανότητα: 0.0068
Ζεύγος Pixel: 153 187, Πιθανότητα: 0.0015
Ζεύγος Pixel: 153 204, Πιθανότητα: 0.0009
Ζεύγος Pixel: 153 221, Πιθανότητα: 0.0002
Ζεύγος Pixel: 153 238, Πιθανότητα: 0.0002
Ζεύγος Pixel: 153 255, Πιθανότητα: 0.0001
Ζεύγος Pixel: 170 17, Πιθανότητα: 0.0001
Ζεύγος Pixel: 170 34, Πιθανότητα: 0.0000
Ζεύγος Pixel: 170 51, Πιθανότητα: 0.0000
Ζεύγος Pixel: 170 68, Πιθανότητα: 0.0001
Ζεύγος Pixel: 170 85, Πιθανότητα: 0.0003
Ζεύγος Pixel: 170 102, Πιθανότητα: 0.0001
Ζεύγος Pixel: 170 119, Πιθανότητα: 0.0007
Ζεύγος Pixel: 170 136, Πιθανότητα: 0.0022
Ζεύγος Pixel: 170 153, Πιθανότητα: 0.0105
Ζεύγος Pixel: 170 170, Πιθανότητα: 0.0168
Ζεύγος Pixel: 170 187, Πιθανότητα: 0.0060
Ζεύγος Pixel: 170 204, Πιθανότητα: 0.0014
Ζεύγος Pixel: 170 221, Πιθανότητα: 0.0004
Ζεύγος Pixel: 170 238, Πιθανότητα: 0.0002
Ζεύγος Pixel: 170 255, Πιθανότητα: 0.0001
Ζεύγος Pixel: 187 17, Πιθανότητα: 0.0000
Ζεύγος Pixel: 187 85, Πιθανότητα: 0.0000
Ζεύγος Pixel: 187 102, Πιθανότητα: 0.0001
Ζεύγος Pixel: 187 119, Πιθανότητα: 0.0002
Ζεύγος Pixel: 187 136, Πιθανότητα: 0.0006
Ζεύγος Pixel: 187 153, Πιθανότητα: 0.0024
Ζεύγος Pixel: 187 170, Πιθανότητα: 0.0079
Ζεύγος Pixel: 187 187, Πιθανότητα: 0.0136
Ζεύγος Pixel: 187 204, Πιθανότητα: 0.0059
Ζεύγος Pixel: 187 221, Πιθανότητα: 0.0015
Ζεύγος Pixel: 187 238, Πιθανότητα: 0.0005
Ζεύγος Pixel: 187 255, Πιθανότητα: 0.0002
Ζεύγος Pixel: 204 102, Πιθανότητα: 0.0001
Ζεύγος Pixel: 204 119, Πιθανότητα: 0.0000
```

```
Ζεύγος Pixel: 204 136, Πιθανότητα: 0.0004
Ζεύγος Pixel: 204 153, Πιθανότητα: 0.0011
Ζεύγος Pixel: 204 170, Πιθανότητα: 0.0022
Ζεύγος Pixel: 204 187, Πιθανότητα: 0.0073
Ζεύγος Pixel: 204 204, Πιθανότητα: 0.0167
Ζεύγος Pixel: 204 221, Πιθανότητα: 0.0046
Ζεύγος Pixel: 204 238, Πιθανότητα: 0.0010
Ζεύγος Pixel: 204 255, Πιθανότητα: 0.0002
Ζεύγος Pixel: 221 85, Πιθανότητα: 0.0001
Ζεύγος Pixel: 221 119, Πιθανότητα: 0.0000
Ζεύγος Pixel: 221 136, Πιθανότητα: 0.0001
Ζεύγος Pixel: 221 153, Πιθανότητα: 0.0001
Ζεύγος Pixel: 221 170, Πιθανότητα: 0.0007
Ζεύγος Pixel: 221 187, Πιθανότητα: 0.0014
Ζεύγος Pixel: 221 204, Πιθανότητα: 0.0068
Ζεύγος Pixel: 221 221, Πιθανότητα: 0.0128
Ζεύγος Pixel: 221 238, Πιθανότητα: 0.0036
Ζεύγος Pixel: 221 255, Πιθανότητα: 0.0003
Ζεύγος Pixel: 238 85, Πιθανότητα: 0.0000
Ζεύγος Pixel: 238 119, Πιθανότητα: 0.0000
Ζεύγος Pixel: 238 153, Πιθανότητα: 0.0001
Ζεύγος Pixel: 238 170, Πιθανότητα: 0.0001
Ζεύγος Pixel: 238 187, Πιθανότητα: 0.0003
Ζεύγος Pixel: 238 204, Πιθανότητα: 0.0009
Ζεύγος Pixel: 238 221, Πιθανότητα: 0.0056
Ζεύγος Pixel: 238 238, Πιθανότητα: 0.0141
Ζεύγος Pixel: 238 255, Πιθανότητα: 0.0013
Ζεύγος Pixel: 255 153, Πιθανότητα: 0.0000
Ζεύγος Pixel: 255 170, Πιθανότητα: 0.0001
Ζεύγος Pixel: 255 221, Πιθανότητα: 0.0001
Ζεύγος Pixel: 255 238, Πιθανότητα: 0.0022
Ζεύγος Pixel: 255 255, Πιθανότητα: 0.0007
```

b.i.

Θα πρέπει να υπολογίσουμε την εντροπία των ζευγών τιμών pixel της εικόνας "parrot.png". Η εντροπία, στη θεωρία πληροφοριών, μετρά το επίπεδο αβεβαιότητας ή την ποσότητα της πληροφορίας που περιέχει μια πηγή δεδομένων.

Για να υπολογίσουμε την εντροπία των ζευγών τιμών pixel, θα ακολουθήσουμε τον τύπο της εντροπίας του Shannon για μια διακριτή πηγή πληροφορίας:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

όπου H(X) είναι η εντροπία της πηγής, p(x_i) είναι η πιθανότητα εμφάνισης της τιμής x_i και το άθροισμα γίνεται πάνω σε όλα τα δυνατά σύμβολα της πηγής.

Στην περίπτωσή μας, τα σύμβολα είναι τα ζεύγη τιμών pixel και θα χρησιμοποιήσουμε τις πιθανότητες που έχουν υπολογιστεί από το προηγούμενο βήμα.

Η εντροπία για τη δεύτερης τάξης επέκταση πηγής είναι 5.6520.

<u>ΑΠΟΤΕΛΕΣΜΑ:</u>

```
Εντροπία για τη δεύτερης τάξης επέκταση πηγής: 5.6520
>>
```

ii.

Το μέσο μήκος του κώδικα Huffman υπολογίζεται βάσει του μήκους κάθε κωδικού και της πιθανότητας εμφάνισης του αντίστοιχου συμβόλου (σε αυτή την περίπτωση, του κάθε ζεύγους τιμών pixel).

Ο τύπος για το μέσο μήκος L ενός κωδικού Huffman δίνεται από:

$$L = \sum_{i=1}^{n} p(x_i) * l(x_i)$$

όπου p(x_i) είναι η πιθανότητα του συμβόλου x_i και l(x_i) το μήκος του κωδικού Huffman για αυτό το σύμβολο.

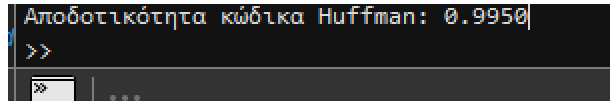
Άρα το μέσο μήκος κωδικού Huffman για τα ζεύγη είναι 5.6801.

```
>> parrot2
Μέσο μήκος κωδικού Huffman για τα ζεύγη: 5.6801
```

iii.
Από την θεωρία μας γνωρίζουμε πως η αποδοτικότητα ενός κώδικα ορίζεται ως:

$$\eta = \frac{H(X)}{L} = \frac{\text{Εντροπία Πηγής}}{\text{Μέσο Μήκος Κώδικα}} \leq 1$$

Επομένως: 5.6520/5.6801 ≈ 0.9950



ΣΧΟΛΙΑ:

Εντροπία της Πηγής (H(X)): Η εντροπία μετρά τη μέση ποσότητα πληροφορίας που περιέχεται σε κάθε σύμβολο της πηγής. Ένας αριθμός εντροπίας 5.6520 bits ανά σύμβολο δείχνει ότι η εικόνα περιέχει μία σημαντική ποικιλία στα δεδομένα της. Όσο υψηλότερη είναι η εντροπία, τόσο μεγαλύτερη είναι η αβεβαιότητα.

Μέσο Μήκος του Κώδικα Huffman (L): Το μέσο μήκος του κώδικα Huffman δείχνει τον μέσο αριθμό των bits που απαιτούνται για να αναπαραστήσουμε κάθε σύμβολο της πηγής. Ένα μέσο μήκος κώδικα 5.6801 bits ανά σύμβολο είναι πολύ κοντά στην τιμή της εντροπίας, που υποδηλώνει ότι ο κώδικας Huffman είναι αρκετά αποδοτικός στην αναπαράσταση των δεδομένων με ελάχιστη υπερβάλλουσα πληροφορία.

Αποδοτικότητα του Κώδικα (Efficiency): Η αποδοτικότητα ενός κώδικα, που ορίζεται ως το λόγο της εντροπίας προς το μέσο μήκος του κώδικα, μετρά πόσο κοντά είναι ο κώδικας στη θεωρητική βέλτιστη απόδοση. Η τιμή 0.9950 σημαίνει ότι ο κώδικας Huffman που χρησιμοποιήσαμε είναι πολύ κοντά στη βέλτιστη απόδοση, παρέχοντας έναν αποδοτικό τρόπο αναπαράστασης των δεδομένων της εικόνας.

c.

Εντροπία H(X):

- Για την άσκηση 1b (πρώτης τάξης επέκταση πηγής), η εντροπία είναι 3.7831 bits.
- Για την άσκηση 2b (δεύτερης τάξης επέκταση πηγής), η εντροπία ανέρχεται σε 5.6520 bits
  Ή 5.6520/2=2.826 bits/έξοδο πηγής(βιβλίο σ.687).

Η αύξηση της εντροπίας στη δεύτερη τάξη επέκτασης υποδηλώνει μεγαλύτερη αβεβαιότητα ή πολυπλοκότητα στα δεδομένα. Με άλλα λόγια, η εξέταση των δεδομένων σε επίπεδο ζευγών συμβόλων αποκαλύπτει μια πιο περίπλοκη δομή από την αναλογία που παρατηρείται με την εξέταση μεμονωμένων συμβόλων.

Μέσο Μήκος Κώδικα (L):

- Για την άσκηση 1b, το μέσο μήκος κώδικα είναι 3.8374 bits.
- Για την άσκηση 2b, το μέσο μήκος κώδικα αυξάνεται σε 5.6801 bits 5.6801/2=2.84005 bit/έξοδο πηγής

Αυτό είναι αναμενόμενο δεδομένης της αυξημένης εντροπίας στη δεύτερη τάξη επέκτασης. Η αυξημένη εντροπία απαιτεί περισσότερα bits ανά σύμβολο για την αποτελεσματική κωδικοποίηση.

Αποδοτικότητα:

- Για την άσκηση 1b, η αποδοτικότητα είναι 0.9859.
- Για την άσκηση 2b, η αποδοτικότητα είναι 0.9950.

Η αύξηση της αποδοτικότητας στη δεύτερη τάξη επέκτασης υποδηλώνει ότι ο κώδικας Huffman είναι πιο κοντά στη βέλτιστη απόδοση όταν λαμβάνει υπόψη τις σχέσεις μεταξύ συνεχόμενων συμβόλων. Αυτό επιβεβαιώνει την ιδέα ότι η δεύτερης τάξης επέκταση παρέχει μια πιο πλήρη αναπαράσταση της δομής των δεδομένων, οδηγώντας σε πιο αποδοτική κωδικοποίηση.

3.a.

Η βασική αιτία, που ο τύπος της επέκτασης πηγής $H(X^n)=nH(X)$ δεν ισχύει στην περίπτωσή μας, είναι ότι οι τιμές των pixel στην εικόνα (ή γενικότερα τα σύμβολα σε μια πηγή πληροφορίας) δεν είναι ανεξάρτητα και ισόπιθανα διανεμημένα. Ο τύπος $H(X^2)=2H(X)$ ισχύει μόνο όταν τα σύμβολα είναι πλήρως ανεξάρτητα και έχουν ίδια πιθανότητα εμφάνισης, δηλαδή όταν η πηγή είναι μια απλή πηγή χωρίς μνήμη (memoryless source).

Στην περίπτωση της εικόνας μας, η διαδοχή των τιμών των pixel δεν είναι ανεξάρτητη. Τα pixel σε μια εικόνα συνήθως έχουν κάποια σχετική συνέπεια ή συσχέτιση - π.χ., pixel που βρίσκονται κοντά μεταξύ τους συχνά έχουν παρόμοιες ή σχετικές τιμές. Αυτή η συσχέτιση μειώνει την αβεβαιότητα (ή την εντροπία) σε σχέση με αυτό που θα περιμέναμε αν τα pixel ήταν πλήρως ανεξάρτητα. Ως αποτέλεσμα, η εντροπία της πηγής δεύτερης τάξης δεν είναι απλώς διπλάσια της εντροπίας πρώτης τάξης, αλλά επηρεάζεται από τη συσχέτιση μεταξύ των συνεχόμενων συμβόλων.

Πηγή:«In a natural image block, the value of the current pixel is dependent on the values of some the surrounding pixels because they are part of the same object, texture, contour, etc.»

Από το:
Basic Concepts and Techniques of Video Coding and the H.261 Standard
Barry Barnett,
in Handbook of Image and Video Processing (Second Edition), 2005

3.b.

Σύμφωνα με το θεώρημα του Shannon για την κωδικοποίηση πηγής, ισχύει ότι: $H(X) \leq L < H(X)+1$

Αυτό το φράγμα δηλώνει ότι το μέσο μήκος κάθε κώδικα που είναι σχεδιασμένος για μια πηγή πληροφορίας θα είναι πάντα μεγαλύτερο ή ίσο με την εντροπία της πηγής, αλλά λιγότερο από την εντροπία συν ένα bit.

Στην περίπτωση των αποτελεσμάτων που έχουμε από τις ασκήσεις 1B και 2B:

- Για την άσκηση 1B, έχουμε $H(X)=3.7831$ και $L=3.8374$. Βλέπουμε ότι ισχύει $3.7831 \leq 3.8374 < 3.7831+1$, επομένως το μέσο μήκος του κώδικα είναι μέσα στα όρια που ορίζει το θεώρημα του Shannon.
- Για την άσκηση 2B, έχουμε $H(X)=5.6520$ και $L=5.6801$. Και πάλι βλέπουμε ότι ισχύει $5.6520 \leq 5.6801 < 5.6520+1$, επιβεβαιώνοντας τη συμμόρφωση με το θεώρημα.

4. Η εικόνα αναλύεται σε επίπεδο pixel, όπου κάθε pixel αντιπροσωπεύει ένα σύμβολο ή μια τιμή. Αυτό συνάδει με την έννοια της διακριτής πηγής στη θεωρία της πληροφορίας, όπου οι τιμές των pixel αντιστοιχούν στα σύμβολα της πηγής.

Η εντροπία, ως μέτρο της μέσης ποσότητας πληροφορίας που περιέχεται στην πηγή, υπολογίζεται βάσει των πιθανοτήτων εμφάνισης κάθε τιμής pixel. Στην περίπτωση αυτή, υπολογίζονται οι συχνότητες εμφάνισης κάθε τιμής pixel και από αυτές προκύπτει η εντροπία της εικόνας.

Η κωδικοποίηση Huffman είναι μια τεχνική αποδοτικής κωδικοποίησης που βασίζεται στην αρχή της ελαχιστοποίησης του μέσου μήκους του κώδικα, αντιστοιχίζοντας τα συχνότερα σύμβολα στους συντομότερους κώδικες. Αυτό είναι σύμφωνο με την θεωρία της κωδικοποίησης πηγής.

ο λόγος συμπίεσης J μικρότερος του 1 σημαίνει ότι η κωδικοποίηση Huffman έχει μειώσει το μέγεθος της αρχικής δυαδικής αναπαράστασης της εικόνας σχεδόν κατά το ήμισυ. Με άλλα λόγια, η κωδικοποίηση Huffman επιτρέπει την αποθήκευση ή τη μετάδοση της εικόνας χρησιμοποιώντας περίπου το 48% των bits που θα απαιτούνταν εάν δεν χρησιμοποιούσαμε συμπίεση. Αυτό δείχνει ότι η κωδικοποίηση είναι αποδοτική στη μείωση του μεγέθους των δεδομένων χωρίς απώλεια πληροφορίας.

Ο λόγος συμπίεσης J είναι 0.4797

```
>> parrot4
Λόγος συμπίεσης (J): 0.4797
>>
```

5.

Από την θεωρία του μαθήματος γνωρίζουμε:

Δυαδικό Συμμετρικό Κανάλι (BSC): Το BSC είναι ένα μοντέλο καναλιού που χαρακτηρίζεται από μια πιθανότητα σφάλματος p. Σε αυτό το μοντέλο, κάθε bit έχει μια πιθανότητα p=0.5 (λόγω συμμετρίας) να αλλάξει κατάσταση κατά την μετάδοση. Η προσομοίωση αυτού του καναλιού μας επιτρέπει να μελετήσουμε την επίδραση των σφαλμάτων στη μετάδοση δεδομένων.
Θυμόμαστε επίσης ότι η εντροπία μεγιστοποιείται για p=0.5 ενώ η χωρητικότητα του καναλιού για p=1 και p=0.

Εντροπία και Χωρητικότητα Καναλιού: Η εντροπία H(X) είναι ένα μέτρο της μέσης ποσότητας πληροφορίας που αποκομίζεται από ένα σύνολο αποτελεσμάτων. Η χωρητικότητα ενός καναλιού (σε αυτή την περίπτωση του BSC) είναι το μέγιστο ποσοστό πληροφορίας που μπορεί να μεταδοθεί με ελάχιστο σφάλμα και υπολογίζεται ως C=1−H(p)

Αμοιβαία Πληροφορία: Η αμοιβαία πληροφορία μεταξύ δύο τυχαίων μεταβλητών είναι ένα μέτρο της ποσότητας πληροφορίας που κερδίζεται για τη μία μεταβλητή μέσω της γνώσης της άλλης. Στην περίπτωσή μας, αυτό μετρά πόσο πολύ το αρχικό σήμα (πριν από την μετάδοση) επηρεάζει το ληφθέν σήμα (μετά την μετάδοση) και υπολογίζεται ως I(X;Y)=H(X)-H(p).

```
>> parrot5
Χωρητικότητα του καναλιού: 0.4754 bits
Αμοιβαία Πληροφορία: 0.46757
>>
```

Προσομοίωση Δυαδικού Συμμετρικού Καναλιού (BSC) και Υπολογισμός της πιθανότητας p:

```
y = binary_symmetric_channel(encoded_img);
p = sum(encoded_img == y) / length(encoded_img);
```

Στις γραμμές αυτές, προσομοιώνουμε τη μετάδοση της κωδικοποιημένης εικόνας μέσω ενός Δυαδικού Συμμετρικού Καναλιού. Αυτό το μοντέλο καναλιού μιμείται την αλλαγή των bits με μια συγκεκριμένη πιθανότητα, παρέχοντας μια ρεαλιστική αναπαράσταση των σφαλμάτων που μπορεί να συμβούν κατά τη μετάδοση δεδομένων. Η πιθανότητα p που υπολογίζεται αντικατοπτρίζει το ποσοστό των bits που διατηρήθηκαν αναλλοίωτα κατά την διάρκεια της μετάδοσης.

Βρίσκουμε λοιπόν το p ίσο με 0.88:

<u>Υπολογισμός Χωρητικότητας και Αμοιβαίας Πληροφορίας:</u>

```
% Υπολογισμός χωρητικότητας καναλιού
H_p = -p*log2(p) - (1-p)*log2(1-p);
C = 1 - H_p;

% Υπολογίζω πιθανότητες για 0 και 1 αντίστοιχα

p0 = sum(encoded_img == 0) / length(encoded_img);

p1 = sum(encoded_img == 1) / length(encoded_img);

% Αμοιβαία Πληροφορία

H_X = -p0 * log2(p0 + (p0 == 0)) - p1 * log2(p1 + (p1 == 0));

amoivaia_pliroforia = H_X - H_p;
```

Στο συγκεκριμένο τμήμα του κώδικα, επικεντρωνόμαστε στον υπολογισμό δύο κρίσιμων μετρήσεων: της χωρητικότητας του καναλιού C και της αμοιβαίας πληροφορίας. Η χωρητικότητα C δίνει το μέγιστο ποσοστό πληροφορίας που μπορεί να μεταδοθεί με ελάχιστο σφάλμα, ενώ η αμοιβαία πληροφορία αντικατοπτρίζει την ποσότητα της πληροφορίας που διατηρείται μετά την μετάδοση.

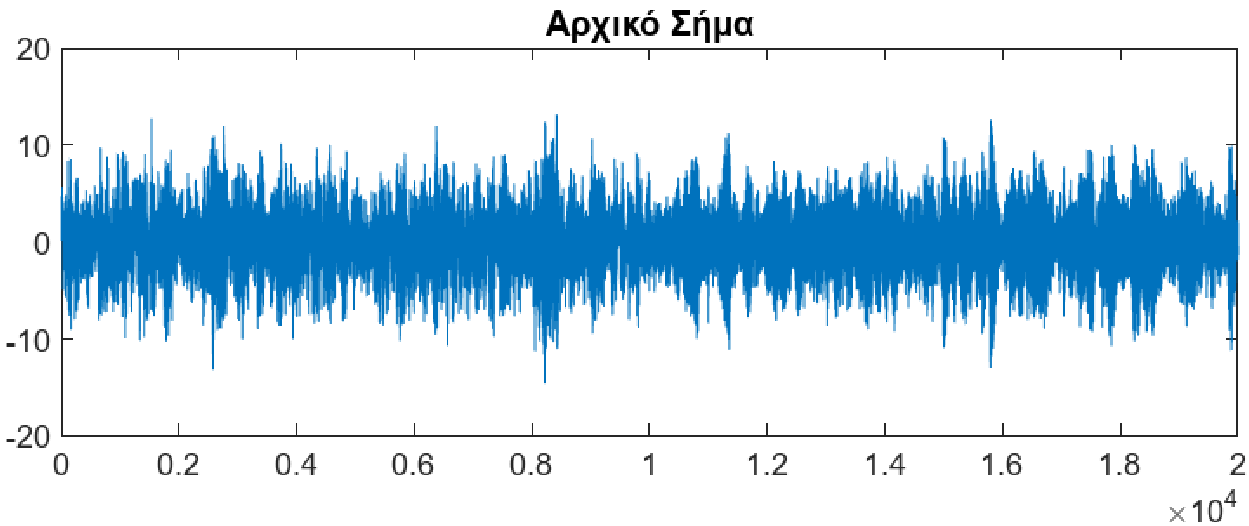
<u>Υπολογισμός Πιθανοτήτων:</u> Αρχικά, υπολογίζονται οι πιθανότητες εμφάνισης των συμβόλων 0 και 1 στην κωδικοποιημένη ακολουθία. Αυτό γίνεται με τις εντολές p0 = sum(encoded_img == 0) / length(encoded_img); και p1 = sum(encoded_img == 1) / length(encoded_img); αντίστοιχα, όπου μετράμε τον αριθμό των μηδενικών και των μονάδων αντίστοιχα, και τον διαιρούμε με το συνολικό μήκος της ακολουθίας για να βρούμε τις σχετικές πιθανότητες.

<u>Υπολογισμός της Εντροπίας H(X):</u> Η εντροπία υπολογίζεται χρησιμοποιώντας τον τύπο $H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$, όπου p(x) είναι η πιθανότητα κάθε συμβόλου στην πηγή.
Η εντολή 'log2' χρησιμοποιείται για τον υπολογισμό του λογαρίθμου βάσεως 2. Η 'sum' στο τέλος υπολογίζει το άθροισμα των όρων της εντροπίας.

ΜΕΡΟΣ Β’ (Κώδικας)

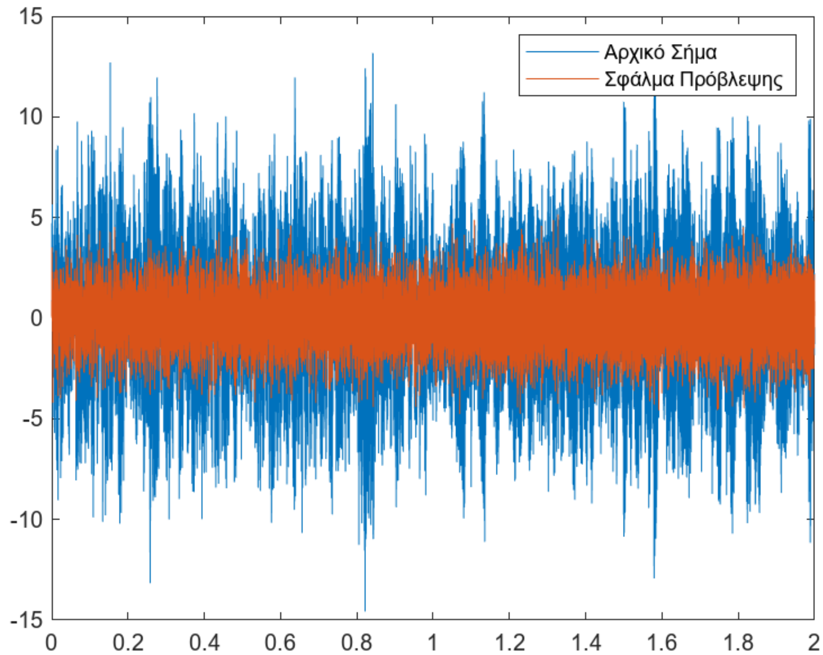
1.

Το πρώτο σήμα περιέχει σημαντική ποσότητα πληροφορίας, με διακυμάνσεις που κατανέμονται σε όλο το εύρος των τιμών του. Αυτό είναι τυπικό για ένα σήμα που δεν έχει υποστεί κωδικοποίηση και μπορεί να περιέχει τόσο χρήσιμη πληροφορία όσο και θόρυβο.



Στο δεύτερο σχήμα (για p=8,N=8), παρουσιάζεται το ανακατασκευασμένο σήμα μετά την εφαρμογή του DPCM. Εδώ, παρατηρούμε ότι το σήμα έχει μικρότερη διακύμανση, που υποδηλώνει ότι η διαδικασία της DPCM μείωσε τη δυναμική περιοχή του σήματος. Αυτό αντικατοπτρίζει την απομάκρυνση μη σημαντικών πληροφοριών ή θορύβου μέσω της κωδικοποίησης πρόβλεψης.

Η μείωση της δυναμικής περιοχής στο ανακατασκευασμένο σήμα υποδηλώνει επίσης ότι η πληροφορία που χάθηκε κατά τη διαδικασία της κβάντισης ίσως δεν ήταν σημαντική για την αναπαράσταση του αρχικού σήματος, πράγμα που συχνά είναι ένας επιθυμητός στόχος στη συμπίεση δεδομένων.



Σχόλια για την υλοποίηση:
Όπως φαίνεται το $y(\boldsymbol{n})$ = ***my_quantizer***$(\boldsymbol{y}(\boldsymbol{n}),\ \boldsymbol{N},\ \boldsymbol{min\_value},\ \boldsymbol{max\_value})$ υλοποιείται σε συνάρτηση της MATLAB όπως ζητείται.
Η δομή του μπορεί να χωριστεί στα εξής 5 μέρη:

1. **Περιορισμός Δυναμικής Περιοχής:** Το σήμα εισόδου y περιορίζεται στη δυναμική περιοχή μεταξύ των τιμών min_value και max_value. Αυτό εξασφαλίζει ότι το σήμα δεν θα ξεπεράσει τις τιμές αυτές, διατηρώντας την ακεραιότητα του κβαντιστή.

2. **Υπολογισμός Βήματος Κβάντισης (Δ):** D = max_value/(2^(N-1)): Αυτός ο τύπος υπολογίζει το βήμα κβάντισης, δηλαδή την ελάχιστη διαφορά μεταξύ δύο διαδοχικών επιπέδων κβάντισης. Ο αριθμός N είναι ο αριθμός των bits που χρησιμοποιούνται για την κβάντιση. Όσο μεγαλύτερος ο αριθμός των bits, τόσο μικρότερο το βήμα κβάντισης και ακριβέστερη η αναπαράσταση του αρχικού σήματος.

3. **Δημιουργία Διανύσματος Κέντρων Περιοχών Κβάντισης:** Το διάνυσμα centers περιέχει τα κέντρα κάθε περιοχής κβάντισης. Τα κέντρα αυτά αποτελούν τις αντιπροσωπευτικές τιμές για κάθε επίπεδο κβάντισης. Η διαδικασία αυτή βοηθά στην αντιστοίχιση κάθε δείγματος στην πλησιέστερη αντιπροσωπευτική τιμή.

   Έστω y το δείγμα εισόδου και Δ το βήμα κβαντισμού. Ο κβαντιστής αντιστοιχίζει το y σε ένα επίπεδο κβάντισης i βασισμένο στην εγγύτητα του y σε κάθε κέντρο κβάντισης c_i. Αυτό μπορεί να εκφραστεί μαθηματικά ως:

$y_{quantized}$=i, όπου i είναι τέτοιο ώστε $(c_i - {}^{\Delta}/_{2}) \leq y \leq (c_i + {}^{\Delta}/_{2})$

Στην πράξη, αυτό σημαίνει ότι ο κβαντιστής βρίσκει το επίπεδο κβάντισης i του οποίου το κέντρο c_i είναι το πλησιέστερο στο δείγμα εισόδου y, λαμβάνοντας υπόψη το εύρος της κάθε περιοχής κβάντισης που ορίζεται από το βήμα κβαντισμού Δ.

4. **Εύρεση Κατάλληλης Περιοχής Κβάντισης για το Δείγμα Εισόδου:** Αυτό το κομμάτι του κώδικα ελέγχει σε ποια περιοχή κβάντισης ανήκει το δείγμα εισόδου y. Βρίσκει το κέντρο περιοχής που είναι πλησιέστερο στο y και θέτει την τιμή y_quantized ίση με τον δείκτη αυτής της περιοχής.

5. **Αντιστοίχιση του Κωδικοποιημένου Δείγματος με το Κέντρο Περιοχής Κβάντισης:** Τέλος, το y_quantized αντιστοιχίζεται με την τιμή του κέντρου της περιοχής κβάντισης, που βρίσκεται στο διάνυσμα centers. Αυτό δίνει την τελική κβαντισμένη τιμή του δείγματος.

Ανάλυση κώδικα source.m:

Υπολογισμός Πινάκων για το Φίλτρο Πρόβλεψης

Γραμμές 13-14: Δημιουργούνται οι πίνακες R και r, οι οποίοι θα χρησιμοποιηθούν για τον υπολογισμό των συντελεστών του φίλτρου πρόβλεψης.
Γραμμές 15-24: Υπολογίζουν τις τιμές των πινάκων R και r βασισμένοι στα δεδομένα του σήματος t και στις οδηγίες της άσκησης στην σελίδα 6.
π.χ: $R_x(i-j) = \frac{1}{N-p}\sum_{n=p+1}^{N} x(n-j)x(n-i)\ \ , 1 \leq i,j \leq \mathrm{p}$ μεταφράζεται σε «R(i, j) = R(i, j) + t(n - j) * t(n - i);» στην γλώσσα της MATLAB και προσαρμοσμένο στις απαιτήσεις της άσκησής μας. Αυτός ο υπολογισμός βασίζεται στην αυτοσυσχέτιση του σήματος.

Υπολογισμός Συντελεστών Φίλτρου και Κβαντισμένων Τιμών

Γραμμές 27-30: Υπολογίζουν τους συντελεστές του φίλτρου πρόβλεψης λύνοντας το γραμμικό σύστημα R·a=r και στη συνέχεια κβαντίζουν αυτούς τους συντελεστές χρησιμοποιώντας την προηγουμένως ορισμένη συνάρτηση κβάντισης.

Κωδικοποιητής DPCM

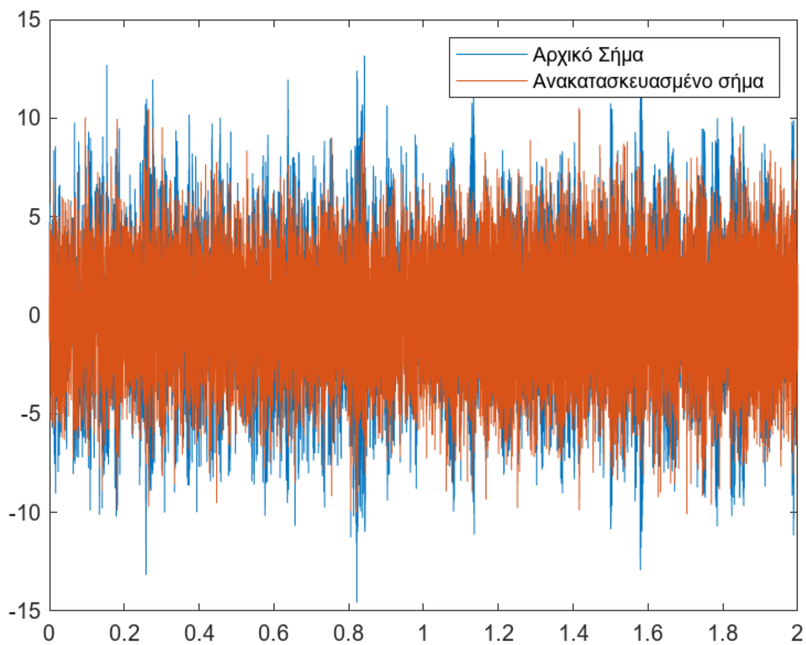
Γραμμές 33-44: Υλοποιούν τον κωδικοποιητή DPCM, όπου κάθε δείγμα του σήματος εισόδου μετατρέπεται σε ένα δείγμα σφάλματος πρόβλεψης και στη συνέχεια κβαντίζεται. Η μνήμη του φίλτρου πρόβλεψης ενημερώνεται σε κάθε βήμα.

Αποκωδικοποιητής DPCM
Γραμμές 46-55: Υλοποιούν τον αποκωδικοποιητή DPCM. Κάθε κβαντισμένο δείγμα μετατρέπεται πίσω σε ένα εκτιμώμενο δείγμα του αρχικού σήματος με τη χρήση του φίλτρου πρόβλεψης.

2.(Κώδικας)

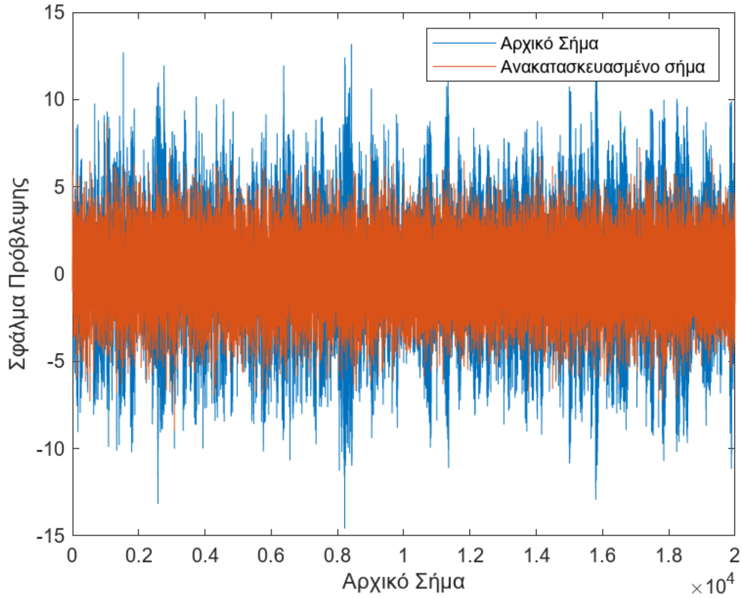
Διάγραμμα 1: p=5,N=1 bit

- Το αρχικό σήμα δείχνει τυπικές μεταβολές που θα περιμέναμε από ένα διακριτοποιημένο σήμα.
- Το σφάλμα πρόβλεψης είναι αρκετά μεγάλο, κάτι που δείχνει ότι με μόνο 1 bit κβάντισης, η ακρίβεια του συστήματος είναι χαμηλή



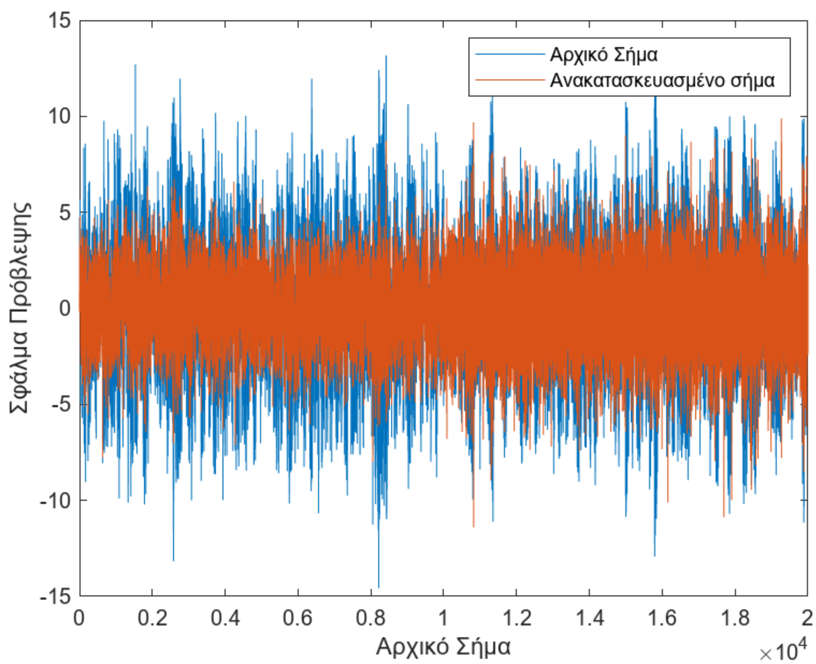
Διάγραμμα 2: p=10,N=1 bit

- Παρόμοια με το Διάγραμμα 1, η αύξηση της τάξης του προβλέπτη δεν φαίνεται να έχει μεγάλη επίδραση λόγω της χαμηλής ανάλυσης κβάντισης.



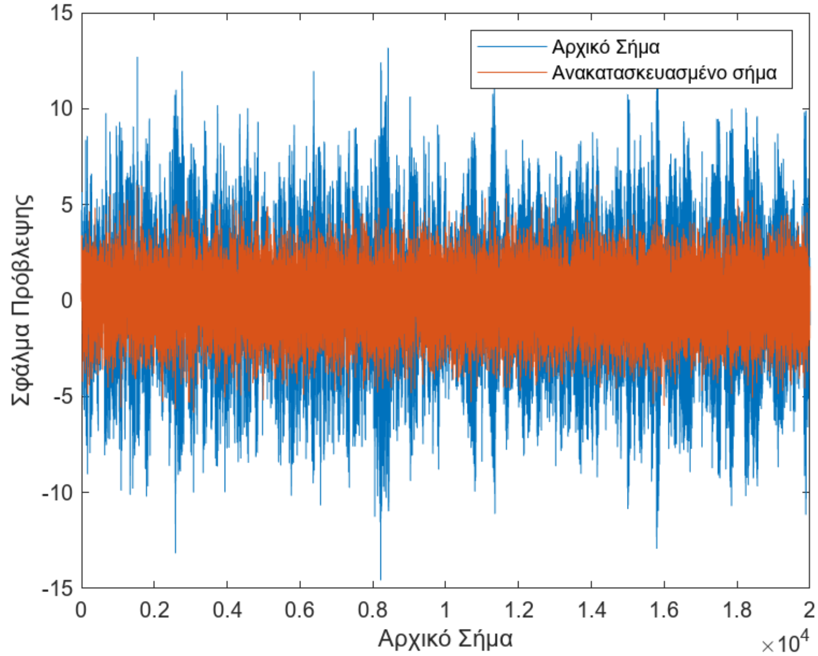
Διάγραμμα 3: p=5,N=2 bits

- Το σφάλμα πρόβλεψης μειώνεται σημαντικά σε σχέση με το Διάγραμμα 1, δείχνοντας την επίδραση της καλύτερης κβάντισης.



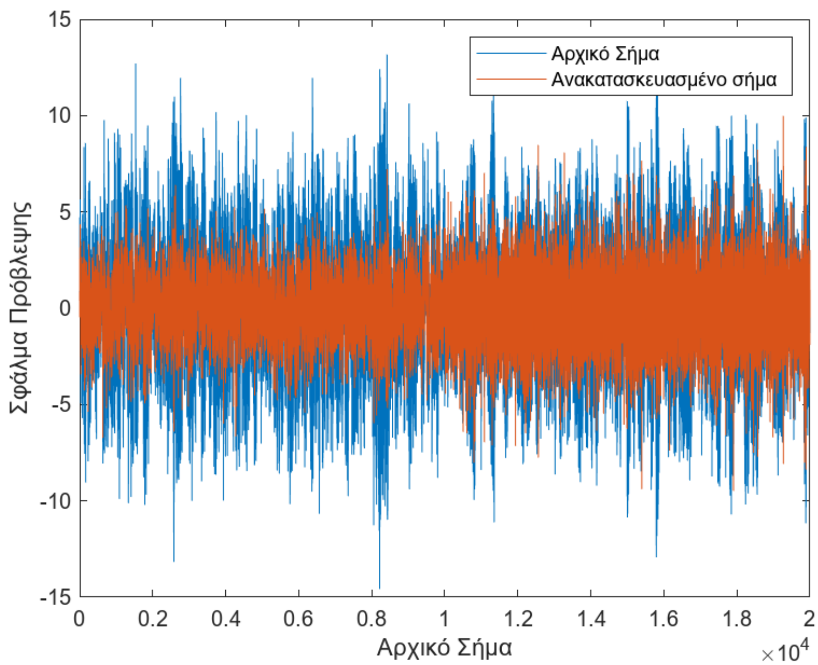
Διάγραμμα 4: p=10,N=2 bits

- Το σφάλμα πρόβλεψης μειώνεται σημαντικά σε σχέση με το Διάγραμμα 2, δείχνοντας την επίδραση της καλύτερης κβάντισης.



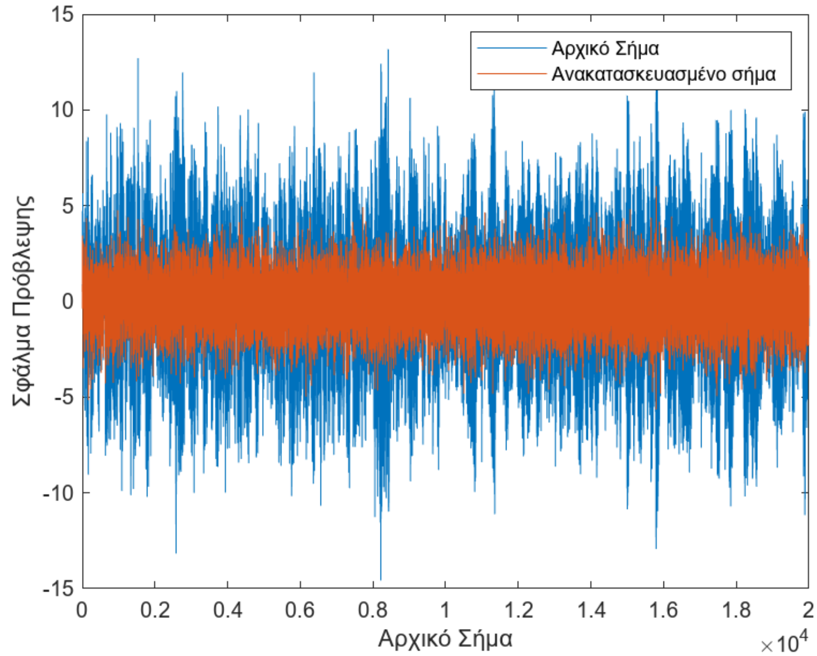
Διάγραμμα 5: p=5,N=3 bits

- Με ακόμα μεγαλύτερη ανάλυση κβάντισης, το σφάλμα πρόβλεψης μειώνεται περαιτέρω, δείχνοντας την θετική επίδραση της αύξησης των bits κβάντισης.



Διάγραμμα 6: p=10,N=3 bits

- Αυτό το διάγραμμα δείχνει το καλύτερο αποτέλεσμα μεταξύ όλων, με το σφάλμα πρόβλεψης να είναι το χαμηλότερο.



ΣΥΜΠΕΡΑΣΜΑ:

Η ανάλυση των αποτελεσμάτων του συστήματος κωδικοποίησης/αποκωδικοποίησης DPCM συνδέεται άμεσα με τις παραμέτρους p (τάξη του προβλέπτη) και N (αριθμός δυαδικών ψηφίων (bit) για την κβάντιση). Με βάση τη θεωρία, είναι δυνατό να κατανοήσουμε γιατί η αύξηση των τιμών του p και N οδηγεί σε καλύτερη αναπαράσταση του αρχικού σήματος στην έξοδο του αποκωδικοποιητή.

Επίδραση του N (Αριθμός Δυαδικών Ψηφίων): Καθώς αυξάνεται το N, η ανάλυση του κβαντιστή βελτιώνεται, προσφέροντας περισσότερες περιοχές κβάντισης. Αυτό μειώνει το σφάλμα κβάντισης, επιτρέποντας στο σήμα πρόβλεψης να προσεγγίσει πιο στενά το αρχικό σήμα. Επομένως, οι διαφορές ή ανωμαλίες μεταξύ του ανακατασκευασμένου και του αρχικού σήματος μειώνονται.

Επίδραση του p (Τάξη του Προβλέπτη): Η αύξηση του p σημαίνει ότι χρησιμοποιούνται περισσότερα προηγούμενα δείγματα για την πρόβλεψη της τιμής του τρέχοντος δείγματος. Αυτό επιτρέπει στο σύστημα να κάνει πιο ακριβείς προβλέψεις, μειώνοντας το σφάλμα πρόβλεψης και βελτιώνοντας την ακρίβεια του ανακατασκευασμένου σήματος.

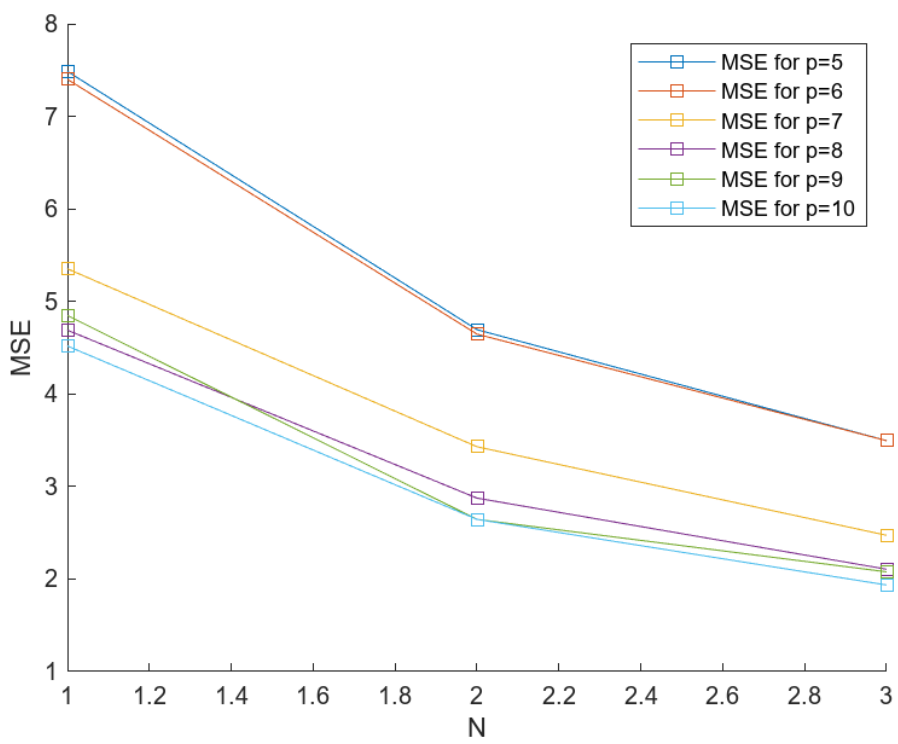
Η επίδραση των διάφορων τιμών του p για σταθερό N φαίνεται να έχει μικρότερη επίδραση στο ανακατασκευασμένο σήμα. Αυτό μπορεί να οφείλεται στο ότι η συνολική βελτίωση της ακρίβειας πρόβλεψης που προκύπτει από την αύξηση του p είναι λιγότερο σημαντική από τη βελτίωση που προσφέρει η αύξηση του N λόγω της μειωμένης ανάλυσης του κβαντιστή.

Συνοψίζοντας, για να πετύχουμε την καλύτερη δυνατή απόδοση στο σύστημα DPCM, είναι κρίσιμο να ρυθμίσουμε κατάλληλα τόσο τον αριθμό των δυαδικών ψηφίων (N) όσο και την τάξη του προβλέπτη (p). Η αύξηση του N θα μειώσει το σφάλμα κβάντισης, ενώ η αύξηση του p θα βελτιώσει την ακρίβεια των προβλέψεων, και αμφότερα θα συμβάλλουν στην παραγωγή ενός ανακατασκευασμένου σήματος

3.

Τα αναμενόμενα αποτελέσματα βάσει της θεωρίας μας είναι ότι:

Με αύξηση του αριθμού των bits κβάντισης, το MSE θα πρέπει να μειώνεται, καθώς ένας κβαντιστής με περισσότερα bits μπορεί να παράγει μια πιο ακριβή αναπαράσταση του αρχικού σήματος.



Ορίζουμε οι τιμές του N (1, 2, 3) και οι αντίστοιχες τιμές του MSE για διαφορετικές τιμές του p (από 5 έως 10). Με την εντολή

«MSE=mean(y.^2)» υπολογίζουμε το Μέσο Τετραγωνικό Σφάλμα για κάθε έναν από τους συνδυασμούς τιμών N = 1:3 και p = 5:10 που επιλέξαμε. Καταλήγουμε στα εξής αποτελέσματα:

	N = 1	N = 2	N = 3
P = 5	7.4863	4.6948	3.4949
P = 6	7.4009	4.6464	3.4945
P = 7	5.3528	3.4289	2.4718
P = 8	4.6891	2.8737	2.1051
P = 9	4.8479	2.6439	2.0767
P = 10	4.5177	2.4630	1.9344

Μπορούμε επομένως να καταλήξουμε στα εξής συμπεράσματα:

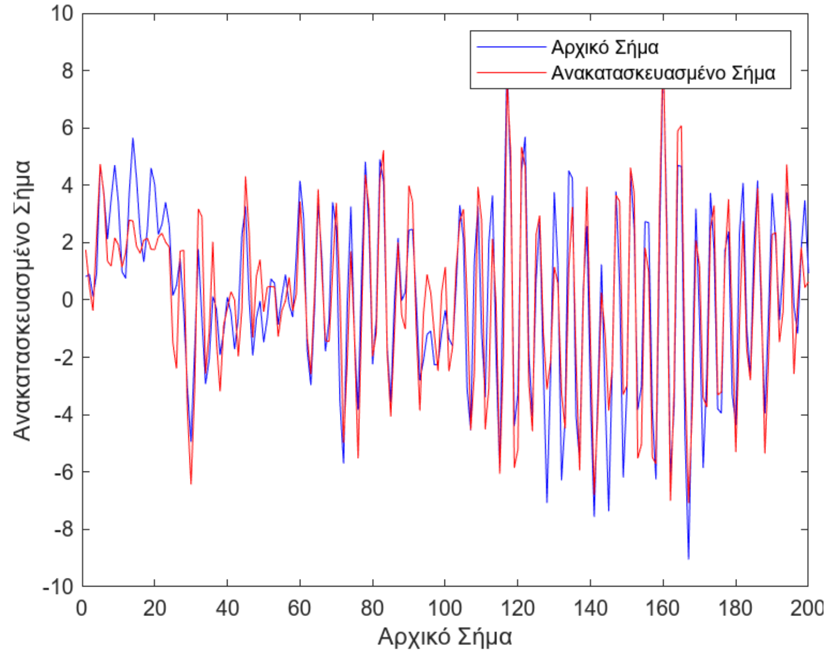
Μεταβολή του MSE με την αλλαγή του p: Παρατηρούμε ότι η αλλαγή της τιμής του p (τάξη του προβλέπτη) προκαλεί μικρές μεταβολές στο MSE. Αυτές οι μεταβολές δεν ακολουθούν κάποια συγκεκριμένη αύξουσα ή φθίνουσα τάση (γενικά όμως έχει φθίνουσα τάση, παρατηρούνται όμως κάποιες ανωμαλίες). Αντίθετα, υπάρχουν διάφορες εναλλαγές στις μέγιστες ή ελάχιστες τιμές του MSE για την ίδια τιμή του N καθώς αλλάζει η τιμή του p.

Επίδραση του N στο MSE: Οι ουσιαστικές μεταβολές στην τιμή του MSE προκύπτουν κυρίως από τις αλλαγές στις τιμές του N, δηλαδή του αριθμού των δυαδικών ψηφίων που χρησιμοποιεί ο κβαντιστής. Αυτό σημαίνει ότι η αύξηση του N βελτιώνει την απόδοση του συστήματος DPCM, μειώνοντας το σφάλμα κβάντισης και επομένως και το σφάλμα πρόβλεψης.

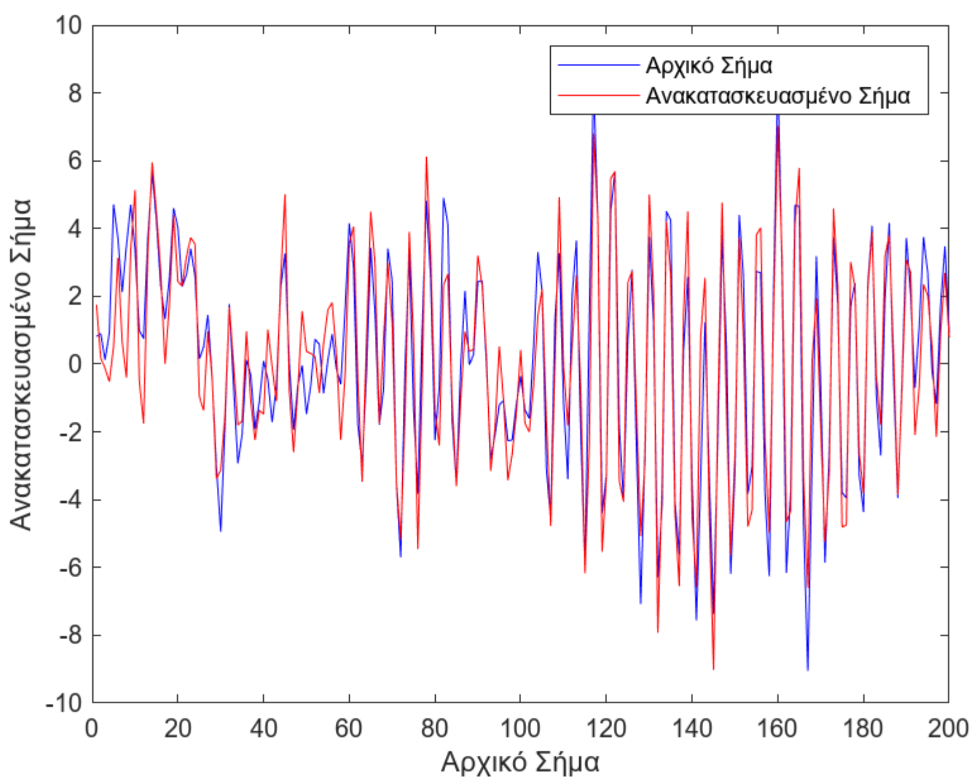
Συμπεράσματα για την Τιμή του p: Παρόλο που η αύξηση της τιμής του p μπορεί να μειώσει το MSE, από ένα σημείο και μετά (για το p=8 συγκεκριμένα), η περαιτέρω αύξηση του p δεν συμβάλλει σημαντικά στη μείωση του σφάλματος. Αυτό σημαίνει ότι υπάρχει μια τιμή του p από την οποία και πέρα η βελτίωση στην ακρίβεια της πρόβλεψης δεν είναι αρκετή για να μειώσει περαιτέρω το MSE.

4.(Κώδικας)

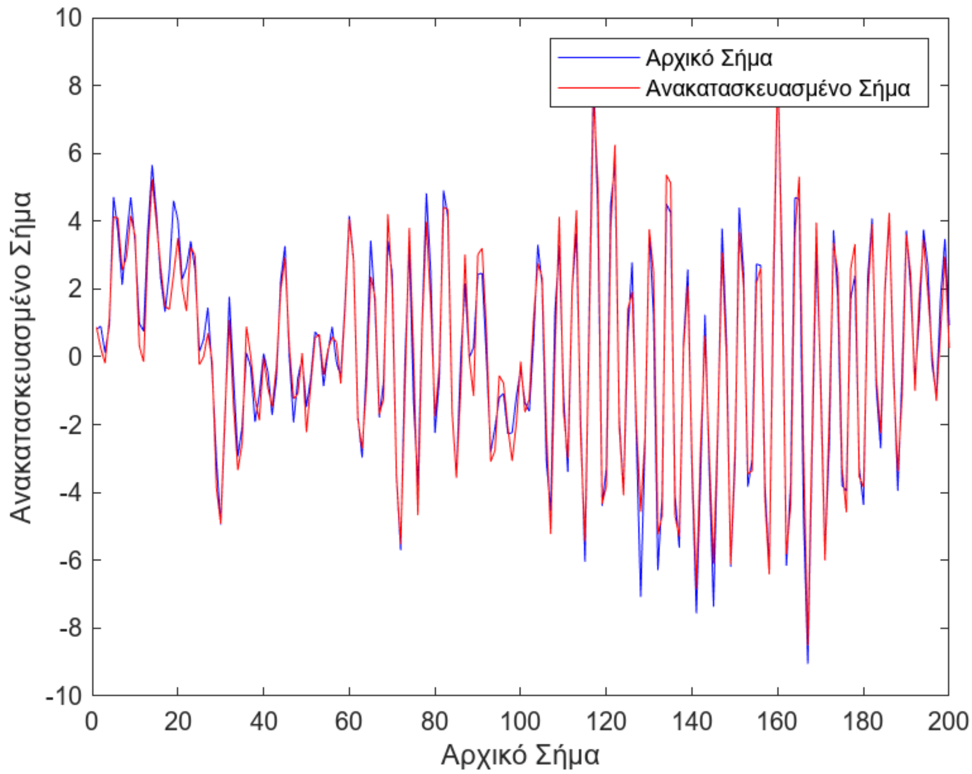
Διάγραμμα 1: p=5,N=1 bit



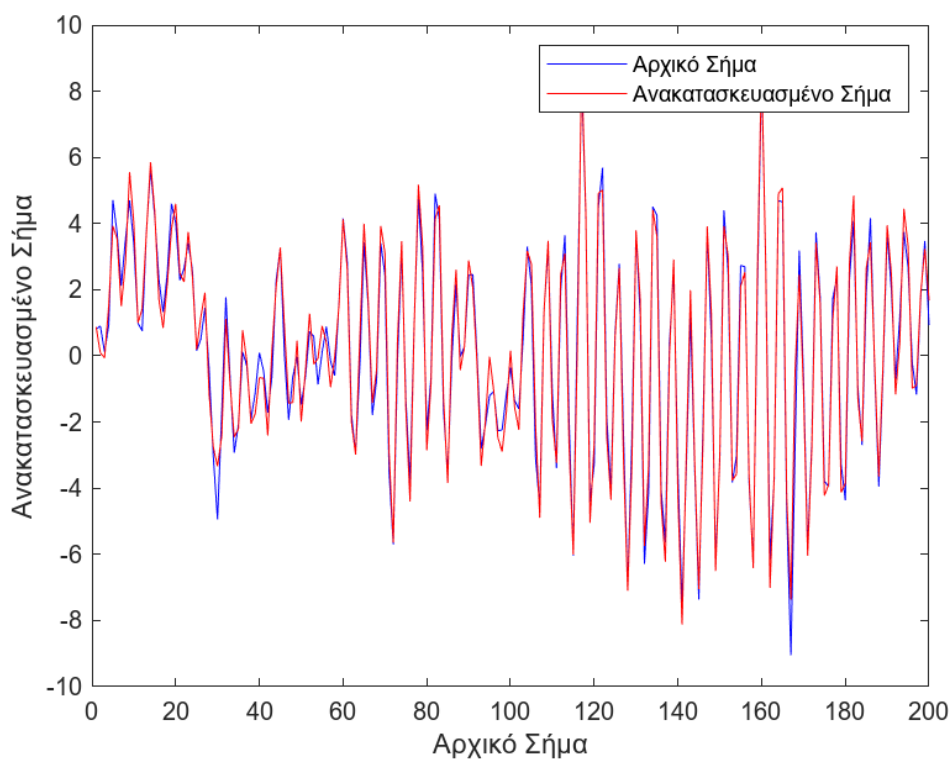
Διάγραμμα 2: p=10,N=1 bit



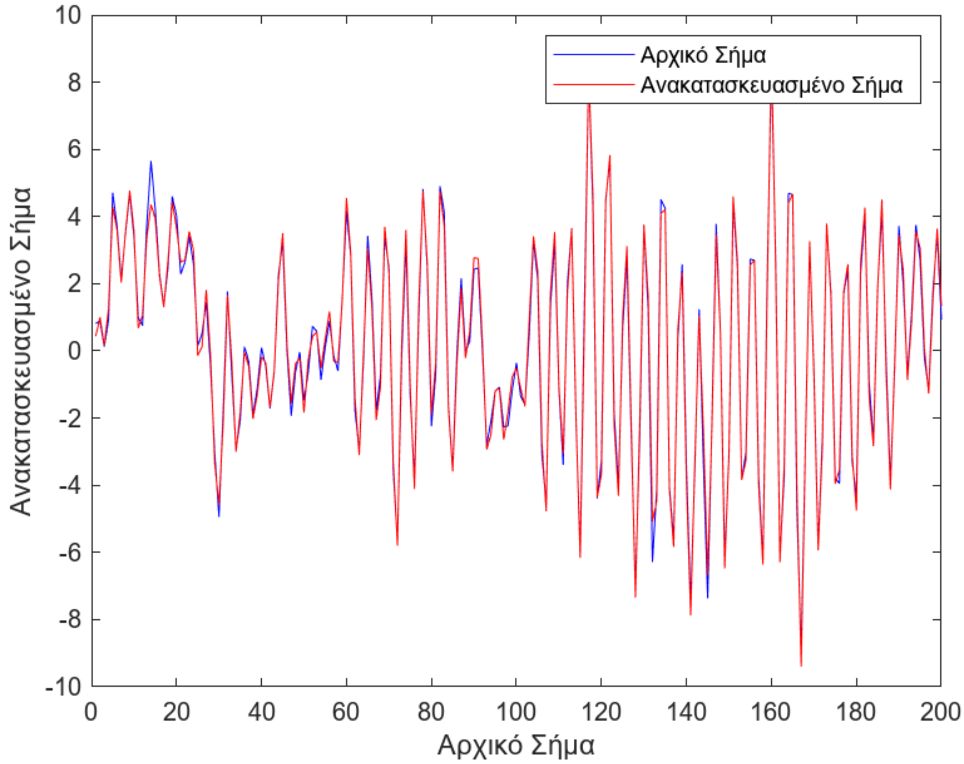
Διάγραμμα 3: p=5,N=2 bits



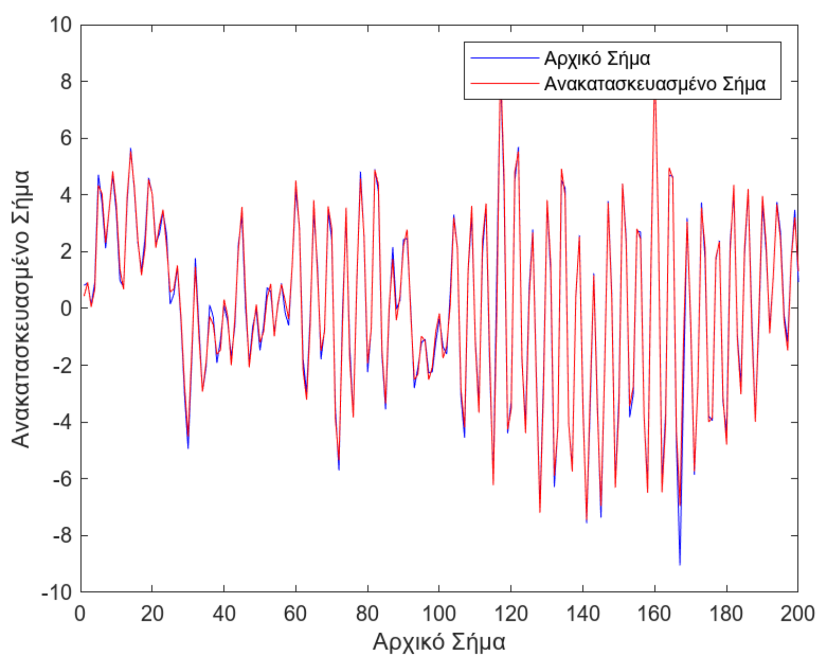
Διάγραμμα 4: p=10,N=2 bits



Διάγραμμα 5: p=5,N=3 bits



Διάγραμμα 6: p=10,N=3 bits



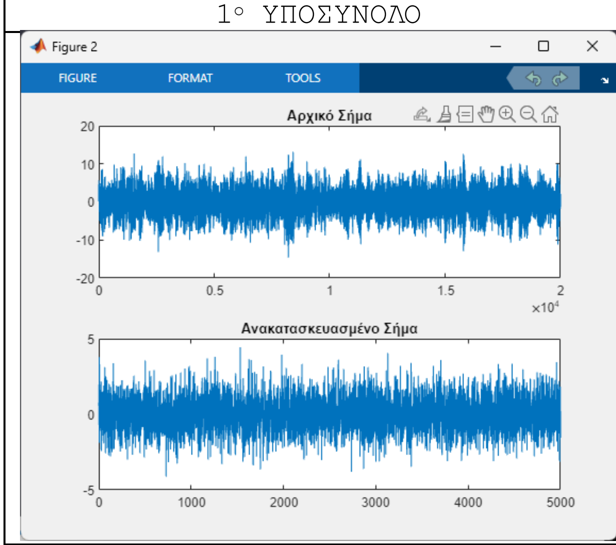
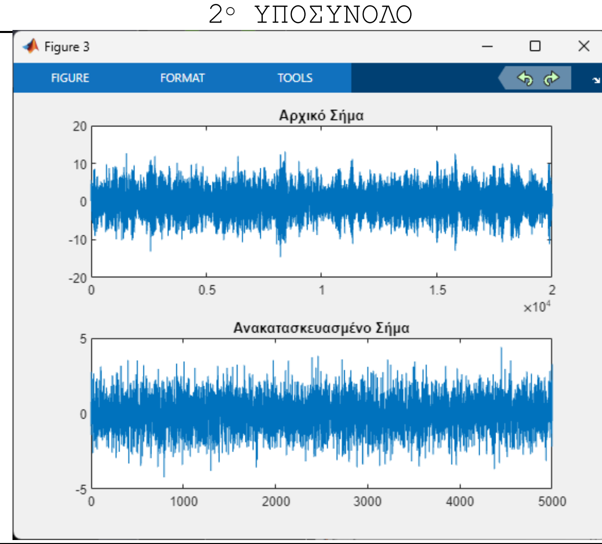
Επίδραση του N: Όταν αυξάνουμε την τιμή του N, διαπιστώνουμε μια σημαντική βελτίωση στην ποιότητα του ανακατασκευασμένου σήματος. Το οποίο προσεγγίζει το αρχικό σήμα (t), καθώς τα σφάλματα και οι ανωμαλίες μειώνονται. Αυτό συμβαίνει επειδή με την αύξηση του N, ο κβαντιστής διαθέτει περισσότερες περιοχές κβάντισης, κάτι που επιτρέπει πιο ακριβή αναπαράσταση των τιμών του σήματος.

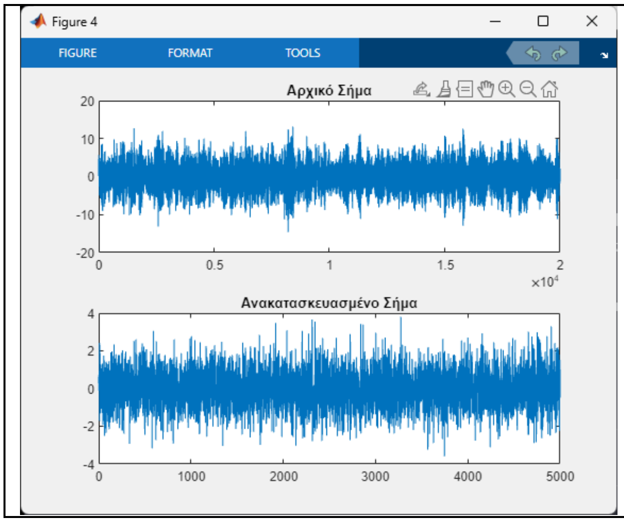
Επίδραση του p: Οι διάφορες τιμές του p φαίνεται ότι έχουν λιγότερο έντονη επίδραση στο ανακατασκευασμένο σήμα, όταν το N παραμένει σταθερό. Αυτό υποδηλώνει ότι, αν και η αύξηση του p (που σημαίνει περισσότερα παρελθόντα δείγματα χρησιμοποιούνται για την πρόβλεψη του τρέχοντος δείγματος) μπορεί να βελτιώσει την ακρίβεια της πρόβλεψης, οι βελτιώσεις αυτές δεν είναι τόσο προφανείς όσο οι βελτιώσεις που προκύπτουν από την αύξηση του N.

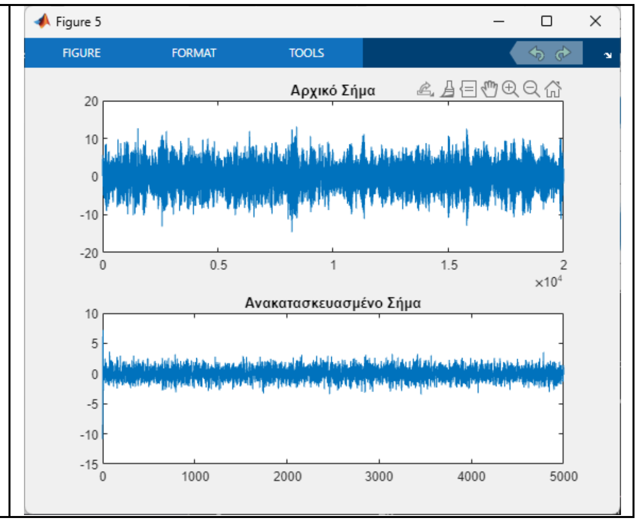
Συνολική Επίδραση στο Σήμα: Συνδυάζοντας τις παραμέτρους p και N, καταλήγουμε στο συμπέρασμα ότι για την καλύτερη αναπαράσταση του αρχικού σήματος στην έξοδο του αποκωδικοποιητή, είναι προτιμότερο να επιλέγουμε υψηλότερες τιμές τόσο για το N όσο και για το p. Ωστόσο, η βελτίωση που προκύπτει από την αύξηση του p είναι περιορισμένη μετά από ένα σημείο, ενώ η αύξηση του N φαίνεται να έχει σταθερά θετική επίδραση στη μείωση του σφάλματος κβάντισης και, κατ' επέκταση, στην ποιότητα του ανακατασκευασμένου σήματος.

Χωρίστε τα δείγματα της πηγής εισόδου σε ισομεγέθη μη επικαλυπτόμενα υποσύνολα των 5000 δειγμάτων. Επαναλάβετε τα παραπάνω ερωτήματα για κάθε ένα από τα υποσύνολα. Σχολιάστε τα αποτελέσματα που λάβατε στα υποσύνολα των 5000 δειγμάτων και συγκρίνετε με αυτά των 20000 δειγμάτων. Τι παρατηρείτε;

1.

1ο ΥΠΟΣΥΝΟΛΟ	2ο ΥΠΟΣΥΝΟΛΟ
	
3ο ΥΠΟΣΥΝΟΛΟ	4ο ΥΠΟΣΥΝΟΛΟ





Στις εικόνες των υποσυνόλων, η διαδικασία κβάντισης και ανακατασκευής του σήματος έχει πραγματοποιηθεί ξεχωριστά για κάθε υποσύνολο. Αυτό οδηγεί σε εμφανείς διακυμάνσεις και διαφορές στην ανακατασκευή μεταξύ των υποσυνόλων. Παρατηρούμε ότι το ποιοτικότερο αποτέλεσμα είναι αυτό του $3^{ου}$ υποσυνόλου, ενώ ενδιαφέρον φέρει και το $4^{ο}$ υποσύνολο. Τα υποσύνολα 1 και 2 είναι πολύ κοντινά στο πλήρη ανακατασκευασμένο σήμα.

Στην εικόνα του πλήρους σήματος των 20000 δειγμάτων, το ανακατασκευασμένο σήμα φαίνεται να έχει μια πιο ομαλή και συνεχή μορφή σε σύγκριση με τα αντίστοιχα υποσύνολα.

ΚΩΔΙΚΑΣ

ΜΕΡΟΣ Α'

1.Α.

```
I = imread('parrot.png');

% Συχνότητες εμφάνισης κάθε τιμής pixel
[symbols, ~, idx] = unique(I);
counts = accumarray(idx(:), 1);
probabilities = counts / numel(I);

% Εμφάνιση των πιθανοτήτων
for i = 1:length(symbols)
    fprintf('Τιμή Pixel: %d, Πιθανότητα: %.4f\n', symbols(i), 
probabilities(i));
end
```

1.B.i
Προσθέτω:

```
% Μετατροπή της εικόνας σε μονοδιάστατο πίνακα
pixels = I(:);

% Υπολογισμός της εντροπίας
entropy = -sum(probabilities .* log2(probabilities));
```

ii.

```
% ΒΗΜΑ 1: Δημιουργία του Huffman δέντρου
dict = huffmandict(symbols, probabilities);

% ΒΗΜΑ 2: Υπολογισμός του μήκους κάθε κωδικού Huffman
for i = 1:length(dict)
    fprintf('Σύμβολο: %d, Κωδικός Huffman: %s\n', dict{i, 1},
num2str(dict{i, 2}));
end

% ΒΗΜΑ 3: Υπολογισμός του μέσου μήκους του κώδικα
avglen = 0;
for i = 1:length(dict)
    avglen = avglen + probabilities(i) * length(dict{i, 2});
end
fprintf('Μέσο μήκος κωδικού Huffman: %.4f\n', avglen);
```

iii.

```
% Υπολογισμός της αποδοτικότητας του κώδικα Huffman
efficiency = entropy / avglen;

% Εκτύπωση της αποδοτικότητας
fprintf('Αποδοτικότητα κώδικα Huffman: %.4f\n', efficiency);
```

2.a.

```
% Δημιουργία ζευγών συνεχόμενων pixels
pairs = [pixels(1:end-1), pixels(2:end)];

% Υπολογισμός των συχνοτήτων εμφάνισης κάθε ζεύγους
[symbols_pairs, ~, idx_pairs] = unique(pairs, 'rows');
counts_pairs = accumarray(idx_pairs, 1);
probabilities_pairs = counts_pairs / (numel(pixels)-1);

% Εμφάνιση των πιθανοτήτων για τα ζεύγη
for i = 1:length(symbols_pairs)
    fprintf('Ζεύγος Pixel: %d %d, Πιθανότητα: %.4f\n',
symbols_pairs(i, 1), symbols_pairs(i, 2), probabilities_pairs(i));
end
```

2.b.i.

```
% Υπολογισμός της εντροπίας για τα ζεύγη τιμών pixel
entropy_pairs = -sum(probabilities_pairs .*
log2(probabilities_pairs));

% Εκτύπωση της εντροπίας
fprintf('Εντροπία για τη δεύτερης τάξης επέκταση πηγής: %.4f\n',
entropy_pairs);
```

ii.

```
% Δημιουργία του Huffman δέντρου για τις γραμμικές διευθύνσεις
dict_pairs = huffmandict(unique_pairs, probabilities_pairs);

% Υπολογισμός του μέσου μήκους του κώδικα Huffman για τις γραμμικές
διευθύνσεις
avglen_pairs = 0;
for i = 1:length(dict_pairs)
    symbol_length = length(dict_pairs{i, 2});
    symbol_probability = probabilities_pairs(i);
avglen_pairs = avglen_pairs + symbol_probability * symbol_length;
end

% Εκτύπωση του μέσου μήκους του κώδικα Huffman
fprintf('Μέσο μήκος κωδικού Huffman για τα ζεύγη: %.4f\n',
avglen_pairs);
```

iii.

```
% Υπολογισμός της αποδοτικότητας του κώδικα Huffman
efficiency = entropy_pairs / avglen_pairs;

% Εκτύπωση της αποδοτικότητας
fprintf('Αποδοτικότητα κώδικα Huffman: %.4f\n', efficiency);
```

4.

```
% Κωδικοποίηση της εικόνας με Huffman
encoded_img = huffmanenco(pixels, dict);

% Υπολογισμός του συνολικού μήκους του κωδικοποιημένου σήματος
encoded_length = length(encoded_img);

% Υπολογισμός του λόγου συμπίεσης J
% Το μέγεθος της αρχικής εικόνας σε bits (κάθε pixel είναι 8 bits)
original_bits = numel(I) * 8;

% Λόγος συμπίεσης
```

```
J = encoded_length / original_bits;

% Εμφάνιση του λόγου συμπίεσης
fprintf('Λόγος συμπίεσης (J): %.4f\n', J);
```

5.

```
% Προσομοίωση Δυαδικού Συμμετρικού Καναλιού και υπολογισμός της
πιθανότητας p
y = binary_symmetric_channel(encoded_img);
p = sum(encoded_img == y) / length(encoded_img);

% Υπολογισμός χωρητικότητας καναλιού και αμοιβαίας πληροφορίας
H_p = -p*log2(p) - (1-p)*log2(1-p);
C = 1 - H_p;
% Calculate the probabilities
p0 = sum(encoded_img == 0) / length(encoded_img);
p1 = sum(encoded_img == 1) / length(encoded_img);

% Calculate the entropy

H_X = -p0 * log2(p0 + (p0 == 0)) - p1 * log2(p1 + (p1 ==0));
amoivaia_pliroforia = H_X - H_p;

% Εμφάνιση των αποτελεσμάτων
fprintf('Χωρητικότητα του καναλιού: %.4f bits\n', C);
fprintf(['Αμοιβαία Πληροφορία: ', num2str(amoivaia_pliroforia)]);
fprintf('\np: %.2f’, p);
```

Μέρος Β:

source.m

```
function source()
    % Αρχικοποίηση παραμέτρων
    p = 10; % Παράμετρος πρόβλεψης
    N = 3; % Αριθμός bit
    minValue = -3.5; % Ελάχιστη τιμή για κβάντιση
    maxValue = 3.5; % Μέγιστη τιμή για κβάντιση

    % Φόρτωση δεδομένων πηγής
    load('source.mat');
    signalLength = length(t);

    % Υπολογισμός πινάκων για το φίλτρο πρόβλεψης
    R = zeros(p, p);
    r = zeros(p, 1);
    for i = 1:p
        for n = (p + 1):signalLength
```

```
            r(i) = r(i) + t(n) * t(n - i);
            for j = 1:p
                R(i, j) = R(i, j) + t(n - j) * t(n - i);
            end
        end
        r(i) = r(i) / (signalLength - p);
        R(i, :) = R(i, :) / (signalLength - p);
    end

    % Υπολογισμός συντελεστών φίλτρου και κβαντισμένων τιμών
    a = R \ r;
    for i = 1:p
        a(i) = my_quantizer(a(i), 8, -2, 2); % Χρήση ορισμένων τιμών
    end

    % Κωδικοποιητής DPCM
    memory = zeros(p, 1);
    y = zeros(signalLength, 1);
    y_hat = zeros(signalLength, 1);
    y_toned = 0;

    for i = 1:signalLength
        y(i) = t(i) - y_toned;
        y_hat(i) = my_quantizer(y(i), N, minValue, maxValue);
        y_hat_toned = y_hat(i) + y_toned;
        memory = [y_hat_toned; memory(1:p - 1)];
        y_toned = a' * memory;
    end

    % Αποκωδικοποιητής DPCM
    memory = zeros(p, 1);
    y_toned = 0;
    y_quantized = zeros(signalLength, 1);

    for i = 1:signalLength
        y_quantized(i) = y_hat(i) + y_toned;
        memory = [y_quantized(i); memory(1:p - 1)];
        y_toned = a' * memory;
    end


    % Ερώτημα 2 - Σχεδιασμός αρχικού σήματος και σφάλματος πρόβλεψης
    plot(t) % Γράφημα του αρχικού σήματος
    hold on
    plot(y) % Γράφημα του σφάλματος πρόβλεψης
    xlabel('Αρχικό Σήμα')
    ylabel('Σφάλμα Πρόβλεψης')
    legend('Αρχικό Σήμα', 'Ανακατασκευασμένο σήμα')
```

```
    % hold off

    % % Ερώτημα 3 - Υπολογισμός μέσου τετραγωνικού σφάλματος
    MSE = mean(y.^2)
    N      = [1 2 3];
    MSEP5  = [7.4863 4.6948 3.4949];
    MSEP6  = [7.4009 4.6464 3.4945];
    MSEP7  = [5.3528 3.4289 2.4718];
    MSEP8  = [4.6891 2.8737 2.1051];
    MSEP9  = [4.8479 2.6439 2.0767];
    MSEP10 = [4.5177 2.6439 1.9344];

      hold on;
      plot1 = plot(N,MSEP5);
      plot2 = plot(N,MSEP6);
      plot3 = plot(N,MSEP7);
      plot4 = plot(N,MSEP8);
      plot5 = plot(N,MSEP9);
      plot6 = plot(N,MSEP10);
      set(plot1,'Marker','square');
      set(plot2,'Marker','square');
      set(plot3,'Marker','square');
      set(plot4,'Marker','square');
      set(plot5,'Marker','square');
      set(plot6,'Marker','square');
      hold off;
      xlabel('N')
      ylabel('MSE')
      legend('MSE for p=5','MSE for p=6','MSE for p=7','MSE for
p=8','MSE for p=9','MSE for p=10')

     % Ερώτημα 4 - Σύγκριση αρχικού και ανακατασκευασμένου σήματος
     plot(t(1:200), 'b') % Αρχικό σήμα
     hold on
     plot(y_quantized (1:200), 'r') % Ανακατασκευασμένο σήμα
     xlabel('Αρχικό Σήμα')
     ylabel('Ανακατασκευασμένο Σήμα')
     legend('Αρχικό Σήμα', 'Ανακατασκευασμένο Σήμα')
     hold off
end
```

Συνάρτηση my_quantizer:

```
function y_quantized = my_quantizer(y,N,min_value,max_value)

% Ο κβαντιστής θα πρέπει να περιορίζει τη δυναμική περιοχή του
σφάλματος πρόβλεψης στις τιμές [min_value : max_value]
```

```
if y < min_value
    y = min_value;
end
if y > max_value
    y = max_value;
end

% ο κβαντιστής υπολογίζει το βήμα κβαντισμού Δ
D = max_value/(2^(N-1));

% centers: διάνυσμα με τα κέντρα των περιοχών κβάντισης
centers = zeros(2^N,1);
centers(1) = max_value - D/2;
centers(2^N) = min_value + D/2;
for i=2:2^N-1
   centers(i) = centers(i-1) - D;
end

% υπολογίζει σε ποια περιοχή ανήκει το δείγμα εισόδου
for i=1:2^N
    if ( (y <= centers(i)+D/2) && (y>= centers(i)-D/2) )
        y_quantized = i;
    end
end

%  το κωδικοποιημένο δείγμα y^(n)
y_quantized = centers(y_quantized);

end
```

Κώδικας για την τμηματοποίηση σε 4*5000:

```
Ίδιος με source.m . . .
.
.
.
% Διαίρεση του σήματος σε ισομεγέθη υποσύνολα
    numSubsets = ceil(signalLength / subsetLength);

    % Επεξεργασία κάθε υποσυνόλου ξεχωριστά
     for s = 1:numSubsets
        startIndex = (s - 1) * subsetLength + 1;
        endIndex = min(s * subsetLength, signalLength);
        currentSubset = t(startIndex:endIndex);
        subsetSignalLength = length(currentSubset);
.
.
```

```
.
% Σχεδίαση του αρχικού και του ανακατασκευασμένου σήματος για σύγκριση
    figure;
    subplot(2, 1, 1);
    plot(t);
    title('Αρχικό Σήμα');
    subplot(2, 1, 2);
    plot(y);
    title('Ανακατασκευασμένο Σήμα');

    end
```