Μέρος Α

(Ι) Μεταβλητές που χρησιμοποίησα σε όλο το Α Μέρος:

- data: αρχικός πίνακας με τα αρχικά δεδομένα του προβήματος.

- spam: Περιέχει τα στοιχεία των spam E-Mail.

- notspam: Περιέχει τα στοιχεία των notspam E-Mail.

- kefalaiaspam: Περιέχει μόνο τη στήλη των κεφαλαίων χαρακτήρων στα spam E-Mail.

- kefalaianotspam: Περιέχει μόνο τη στήλη των κεφαλαίων χαρακτήρων στα κανονικά E-Mail.

- allilouxiaspam: Περιέχει μόνο τη στήλη της μεγαλύτερης μη διακοπτόμενης αλληλουχίας κεφαλαίων χαρακτήρων στα spam E-Mail.

- allilouxianotspam: Περιέχει μόνο τη στήλη της μεγαλύτερης μη διακοπτόμενης αλληλουχίας κεφαλαίων χαρακτήρων στα κανονικά E-Mail.

- n0kefalaiaspam: Περιέχει μόνο τη στήλη των κεφαλαίων χαρακτήρων των spam E-Mail σε αύξουσα σειρά και χωρίς επαναλλήψεις.

- fkefalaiaspam: Περιέχει την συχνότητα εμφάνισης κεφαλαίων χαρακτήρων στα spam E-Mail.

- n0kefalaianotspam: Περιέχει μόνο τη στήλη των κεφαλαίων χαρακτήρων των κανονικών E-Mail σε αύξουσα σειρά και χωρίς επαναλλήψεις.

- fkefalaianotspam: Περιέχει την συχνότητα εμφάνισης κεφαλαίων χαρακτήρων στα κανονικά E-Mail.

- n0allilouxiaspam: Περιέχει μόνο τη στήλη της μεγαλύτερης μη διακοπτόμενης αλληλουχίας κεφαλαίων χαρακτήρων στα spam E-Mail κατά αύξουσα σειρά και χωρίς επαναλλήψεις.

- fallilouxiaspam: Περιέχει τη συχνότητα εμφάνισης μη διακοπτόμενων αλληλουχιών κεφαλαίων χαρακτήρων στα spam E-Mail.

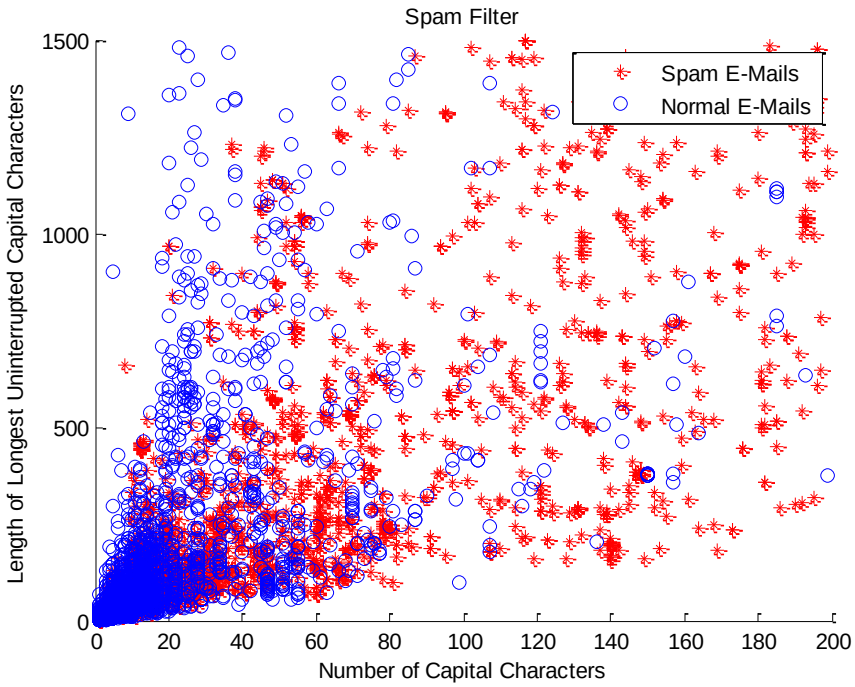
- n0allilouxianotspam: Περιέχει μόνο τη στήλη της μεγαλύτερης μη διακοπτόμενης αλληλουχίας κεφαλαίων χαρακτήρων στα κανονικά E-Mail κατά αύξουσα σειρά και χωρίς επαναλλήψεις.

- fallilouxianotspam: Περιέχει τη συχνότητα εμφάνισης μη διακοπτόμενων αλληλουχιών κεφαλαίων χαρακτήρων στα κανονικά E-Mail.

Έναρξη Προγράμματος

```
>> data=dlmread('spam_data.dat');

>> spam=data(data(:,8)==1,:); % χωρισμός του πίνακα data στα στοιχεία των Spam E-Mail...
```

```
>> notspam=data(data(:,8)==0,:); % ...και στα στοιχεία των κανονικών E-Mail

>> kefalaiaspam=spam(:,6); % Δημιουργία πίνακα κεφαλαίων χαρακτήρων από τα spam E-Mail

>> allilouxiaspam=spam(:,7); % Δημιουργία πίνακα (δείτε πάνω για επεξήγηση των πινάκων)

>> kefalaianotspam=notspam(:,6);

>> allilouxianotspam=notspam(:,7);

>> scatter(kefalaiaspam,allilouxiaspam,'r*') % Δημιουργία γραφικής παράστασης τύπου scatter που απεικονίζει τους κεφαλαίους χαρακτήρες σε σχέση με τις μη διακοπτόμενες αλληλουχίες κεφαλαίων χαρακτήρων στα spam E-Mail, με επισήμανση κόκκινων αστερίσκων.

>> hold on

>> scatter(kefalaianotspam,allilouxianotspam,'bo') % Δημιουργία γραφικής παράστασης τύπου scatter που απεικονίζει τους κεφαλαίους χαρακτήρες σε σχέση με τις μη διακοπτόμενες αλληλουχίες κεφαλαίων χαρακτήρων στα κανονικά E-Mail, με επισήμανση μπλέ κύκλων.

>> xlabel('Number of Capital Characters')

>> ylabel('Length of Longest Uninterrupted Capital Characters')

>> legend('Spam E-Mails', 'Normal E-Mails')

>> title('Spam Filter')
```



Παρατηρώ ότι τα spam E-Mail έχουν περισσότερες μη διακοπτόμενες αλληλουχίες κεφαλαίων χαρακτήρων από τα κανονικά E-Mail, και ότι ο αριθμός των κεφαλαίων χαρακτήρων ιδιαίτερα στα κανονικά E-Mail (τα οποία είναι και κατά 900 περίπου περισσότερα από τα Spam E-Mail) μειώνεται με την αύξηση των μη διακοπτόμενων αλληλουχιών κεφαλαίων χαρακτήρων. Συμπεράσματα και τα

δύο λογικά σύμφωνα με τα δεδομένα του προβήματος και με την κοινή λογική και εμπειρία περί E-Mail.

(II)

Συνεχίζοντας στον ήδη υπάρχων κώδικα του μέρους **(I)**...

(α) >> n0kefalaiaspam=unique(kefalaiaspam); % (Δείτε παραπάνω για επεξήγηση των πινάκων)

>> for i=1:length(n0kefalaiaspam) % Τοποθέτησα μετρητή i ο οποίος θα ψάχνει...

fkefalaiaspam(i)=sum(n0kefalaiaspam(i)==kefalaiaspam); % ...μία μία τη γραμμή του πίνακα n0kefalaiaspam και ύστερα θα εξετάζει όλον τον πίνακα kefalaiaspam για να βρεί ποιές γραμμές του είναι ίσες με το περιεχόμενο της εκάστωτε γραμμή του n0kefalaiaspam. Όσες είναι ίσες θα πάρουν την τιμή 1 και ύστερα θα αθροιστούν όλες με το sum, ώστε να απεικονιστεί το πλήθος τους. Τα αποτελέσματα καταχωρούνται στον πίνακα fkefalaiaspam για την κάθε γραμμή του n0kefalaiaspam, επομένως ο αριθμός των γραμμών του θα είναι ίσος με αυτόν του n0kefalaiaspam. Ακριβώς αντίστοιχη διαδικασία πραγματοποιείται και για τα β,γ,δ.

end

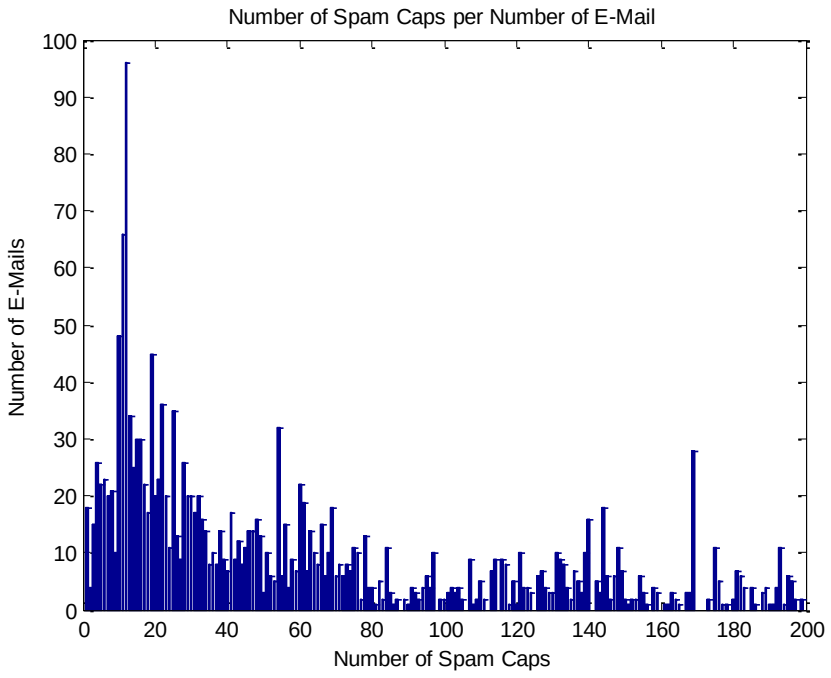
>> figure

>> bar(n0kefalaiaspam,fkefalaiaspam)

>> xlabel('Number of Spam Caps')

>> ylabel('Number of E-Mails')

>> title('Number of Spam Caps per Number of E-Mail')



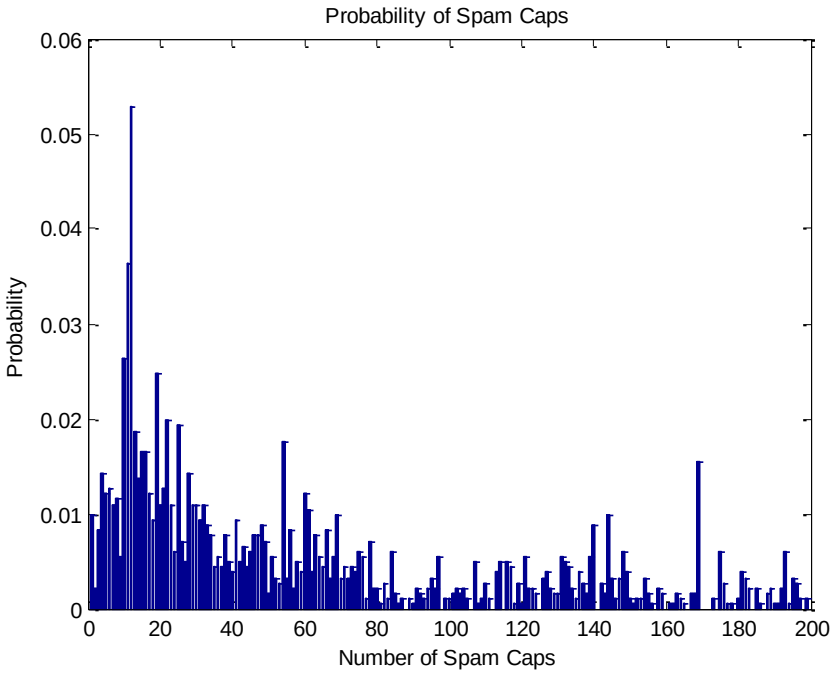
>> figure

>> bar(n0kefalaiaspam,fkefalaiaspam/sum(fkefalaiaspam)) % και εδώ κανονικοποιείται το διάγραμμα ώστε να αναπαριστάνει την πιθανότητα να υπάρχουν 1, 2,...,199 κεφαλαίοι χαρακτήρες κατά τον αριθμό των E-Mail. Παρατηρούμε όπως είναι λογικό ότι αυτή η πιθανότητα μειώνεται με την αύξηση του πλήθους των κεφαλαίων χαρακτήρων. Ακριβώς αντίστοιχη διαδικασία γίνεται και στα β,γ,δ.

>> xlabel('Number of Spam Caps')

>> ylabel('Probability')

>> title('Probability of Spam Caps')



(β) συνεχίζοντας τον κώδικα...

>> n0kefalaianotspam=unique(kefalaianotspam);

>> for i=1:length(n0kefalaianotspam)

fkefalaianotspam(i)=sum(n0kefalaianotspam(i)==kefalaianotspam);

end

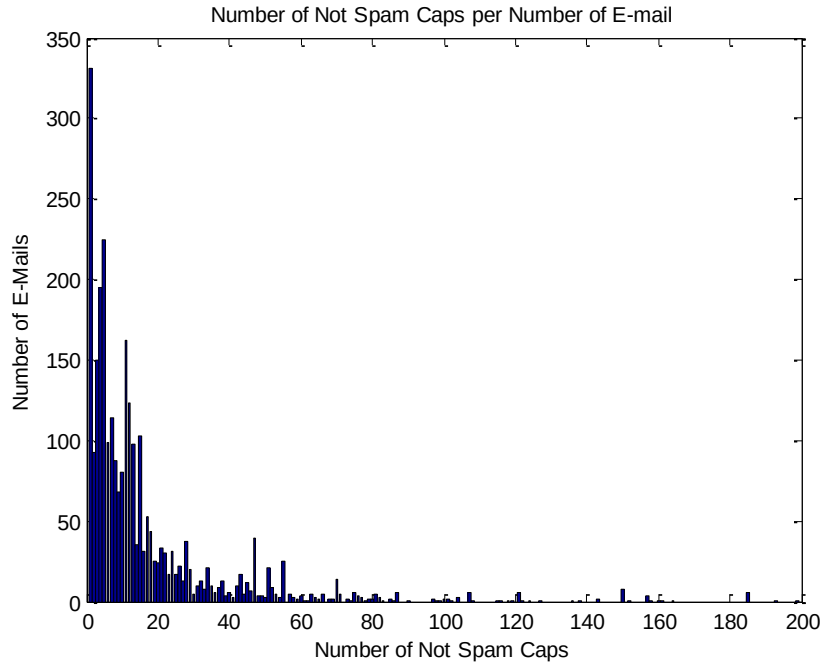
>> figure

>> bar(n0kefalaianotspam,fkefalaianotspam)

>> xlabel('Number of Not Spam Caps')

>> ylabel('Number of E-Mails')

>> title('Number of Not Spam Caps per Number of E-mail')



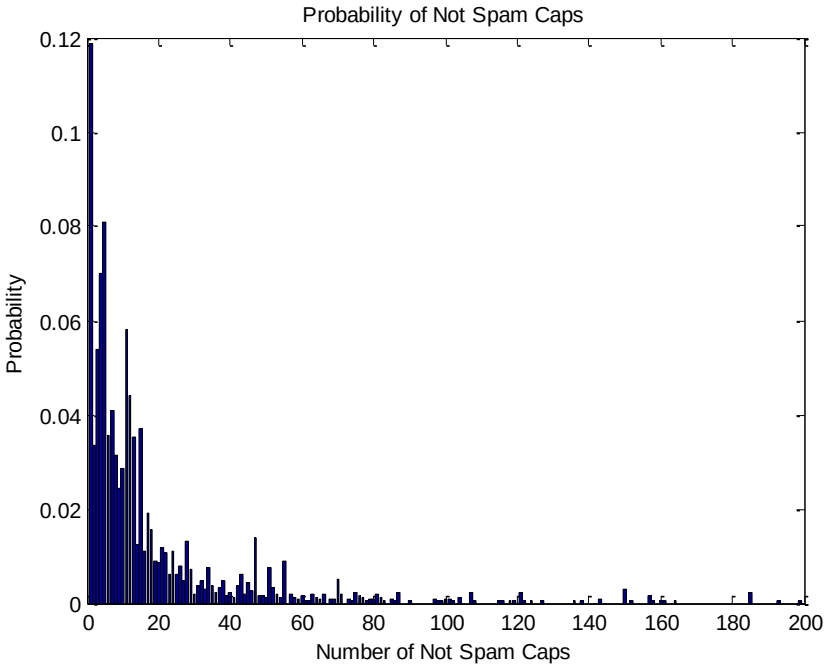
>> figure

>> bar(n0kefalaianotspam,fkefalaianotspam/sum(fkefalaianotspam))

>> xlabel('Number of Not Spam Caps')

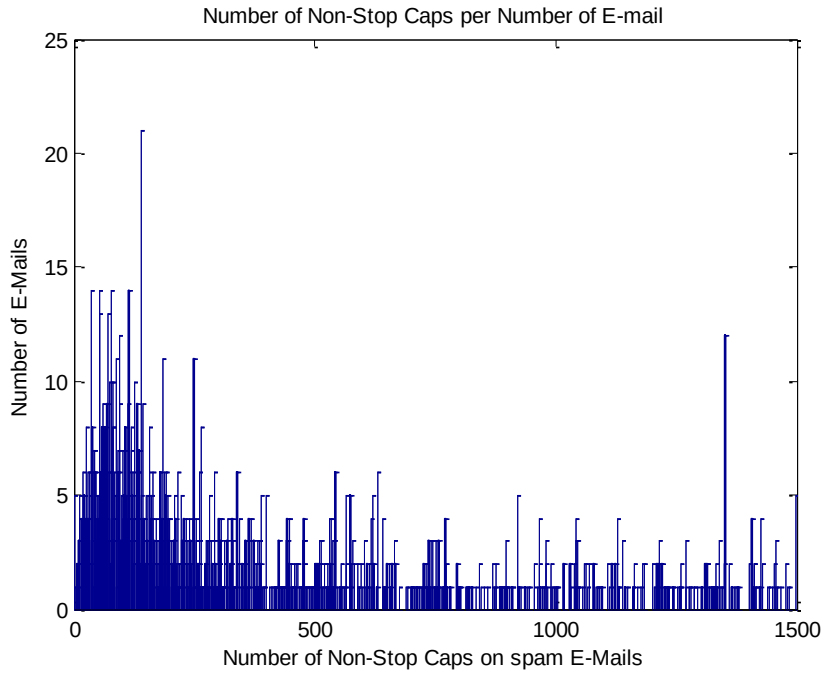
>> ylabel('Probability')

>> title('Probability of Not Spam Caps')

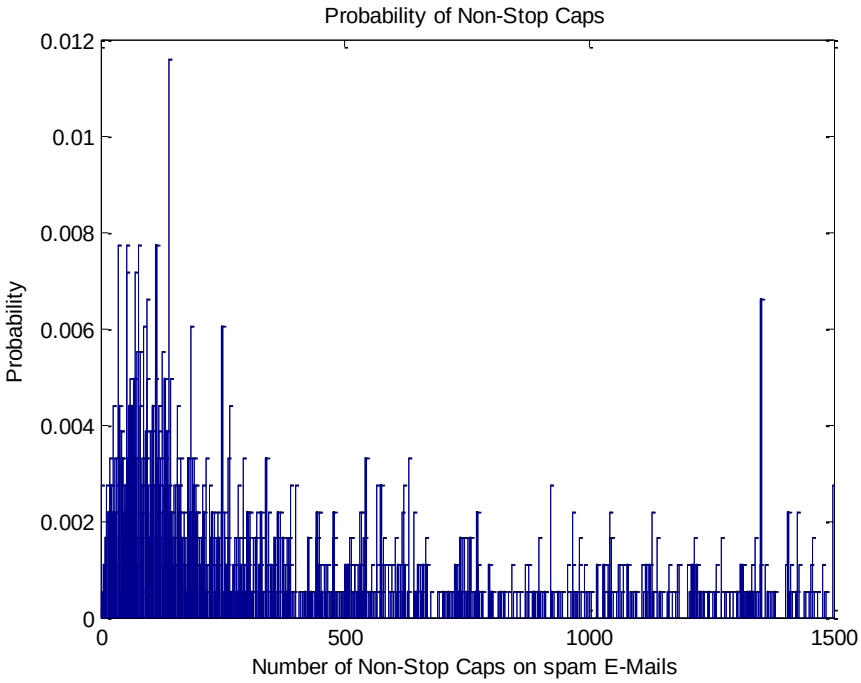


(γ) συνεχίζοντας τον κώδικα...

```
>> n0allilouxiaspam=unique(allilouxiaspam);
>> for i=1:length(n0allilouxiaspam)
fallilouxiaspam(i)=sum(n0allilouxiaspam(i)==allilouxiaspam);
end
>> figure
>> bar(n0allilouxiaspam,fallilouxiaspam)
>> xlabel('Number of Non-Stop Caps on Spam E-Mail')
>> ylabel('Number of E-Mails')
>> title('Number of Non-Stop Caps per Number of E-mail')
```

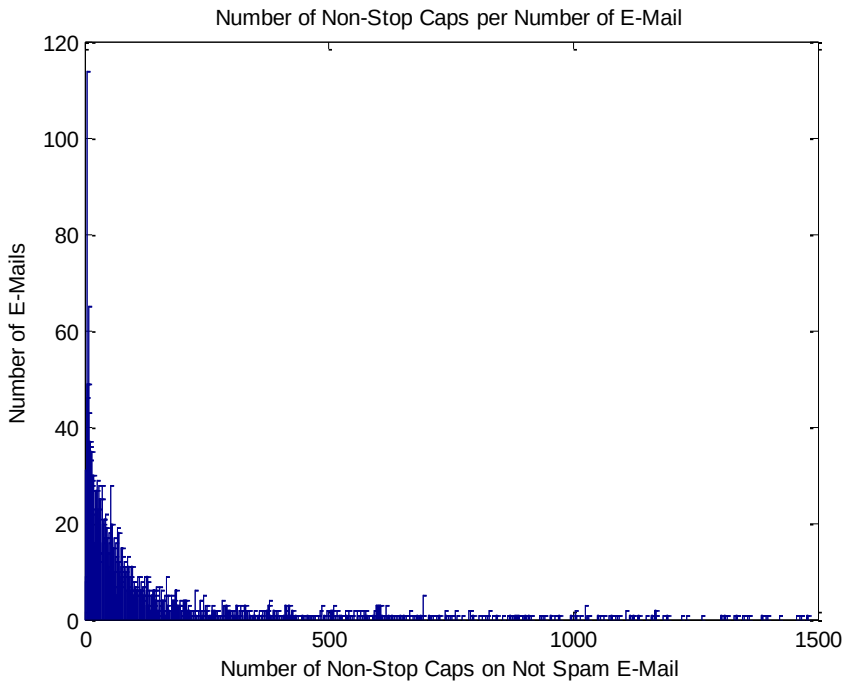



```
>> figure
>> bar(n0allilouxiaspam,fallilouxiaspam/sum(fallilouxiaspam))
>> xlabel('Number of Non-Stop Caps on spam E-Mails')
>> ylabel('Probability')
>> title('Probability of Non-Stop Caps')
```

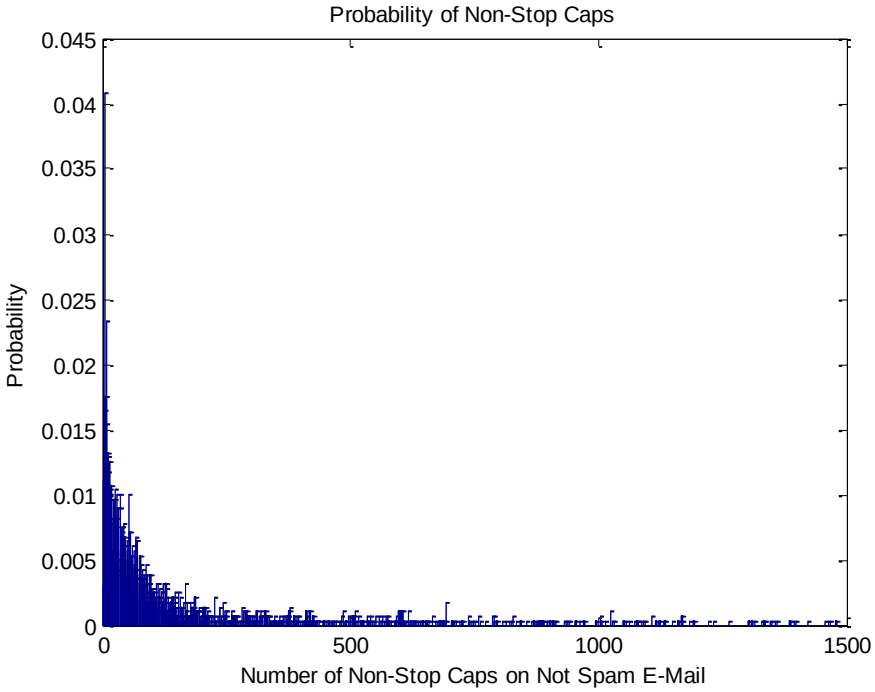


(δ) συνεχίζοντας τον κώδικα...

```
>> n0allilouxianotspam=unique(allilouxianotspam);
>> for i=1:length(n0allilouxianotspam)
fallilouxianotspam(i)=sum(n0allilouxianotspam(i)==allilouxianotspam);
end
>> figure
>> bar(n0allilouxianotspam,fallilouxianotspam)
>> xlabel('Number of Non-Stop Caps on Not Spam E-Mail')
>> ylabel('Number of E-Mails')
>> title('Number of Non-Stop Caps per Number of E-Mail')
```




```
>> figure

>> bar(n0allilouxianotspam,fallilouxianotspam/sum(fallilouxianotspam))

>> xlabel('Number of Non-Stop Caps on Not Spam E-Mail')

>> ylabel('Probability')

>> title('Probability of Non-Stop Caps')
```



Μέρος Β

(Β1) Η συνάρτηση γράφεται στο Editor του Matlab:

```
function [sRow,sCol,sTot,oneT,zeroT]=ProjectB1(A)
if A==logical(A) % Η συνάρτηση πρέπει να δέχεται ως είσοδο λογικό πίνακα.
    sRow=sum(A,2); % Χρησιμοποιώντας την ιδιότητα του sum(A,dim) αθροίζω την κάθε γραμμή
του A.
    sCol=sum(A); % Χρησιμοποιώντας το sum στην πιο απλή της μορφή, αθροίζω την κάθε στήλη
του A.
    sTot=sum(sum(A)); % Αθροίζω ΟΛΑ τα στιχεία του A.
    oneT=sum(sum((A)==1)); % Αθροίζω το πλήθος ΟΛΩΝ των άσσων του πίνακα.
    zeroT=sum(sum((A)==0)); % Αθροίζω το πλήθος ΟΛΩΝ των μηδενικών του πίνακα.
else % Εάν η συνάρτηση δεν είναι λογική η κάθε έξοδος θα λάθει τον κενό πίνακα. Δεν
ζητούσατε ακριβώς αυτό, αλλά κατάλαβα πώς επειδή θέλατε ώς είσοδο μόνο λογικό πίνακα, στην
περίπτωση που η είσοδος δεν είναι λογικός πίνακας το πρόγραμμα δεν θα έπρεπε να λειτουργήσει
(όλοι οι έξοδοι θα έπαιρναν τον κενό πίνακα)
    sRow=[];
    sCol=[];
    sTot=[];
    oneT=[];
    zeroT=[];
end
```

..και η κλήση της συνάρτησης γίνεται από το Command Window του Matlab, ως εξής:

>> [X1,X2,X3,X4,X5]=ProjectB1(A)

Αν στο Command Window δεν έχουμε ορίσει από πρίν τον πίνακα A δεν θα μας βγάλει αποτέλεσμα η εκτέλεση αυτής της εντολής. Γι'αυτό όταν την εκτελέσουμε θα πρέπει ως είσοδο να βάλουμε πίνακα που έχουμε ορίσει από πρίν στο Command Window, ή απλά να τοποθετήσουμε έναν νέο πίνακα πχ.:

>> [X1,X2,X3,X4,X5]=ProjectB1([1,0,1,1;0,0,1,1;1,1,0,1])

όπου [[1,0,1,1;0,0,1,1;1,1,0,1]] : τυχαίος πίνακας εισόδου MxN.

Στην περίπτωση που ως είσοδο θέσουμε μη λογικό πίνακα οι έξοδοι θα αποκτήσουν ως αποτελέσματα τους κενούς πίνακες.

(Β2) Η συνάρτηση γράφεται στο Editor του Matlab:

```
function [out1,out2,out3] = ProjectB2(A,B)
if isreal(A) & isreal(B) % Εξακρίβωση ότι οι πίνακες εισόδου είναι πραγματικοί. Δεν
είχαμε κάνει μαζί την εντολή isreal αλλά δεν πιστεύω πώς υπήρχε πρόβλημα που την
χρησιμοποίησα.
    if size(A)==size(B) % Εξακρίβωση ότι οι πίνακες εισόδου έχουν την ίδιες διαστάσεις.
        out1=find(A==B); % Το out1 θα περιέχει τις θέσεις όπου οι πίνακες A και B έχουν
τις ίδιες τιμές. Τα A==B τοποθετούν άσσους σ'αυτή την περίπτωση και το find ύστερα πάει και
βρίσκει την θέση αυτών των άσσων παραλείποντας την θέση των μηδενικών.
        out2=intersect(A,B); % Αυτήν την εντολή δεν την είχαμε διδαχθεί μαζί. Αλλά την
βρήκα εγώ και την τοποθέτησα διότι μας δίνει τον πιο απλό τρόπο να παραστήσουμε τις τιμές που
είναι κοινές στους πίνακες A και B (δεν πιστεύω να υπάρχει πρόβλημα που την χρησιμοποιήσα)
```

```
        halfadd=floor(((max(max(A))+max(max(B)))/2)); % Δημιουργία
μεταβλητής εσωτερικά της συνάρτησης που θα περιέχει το ημιάθροισμα των
μεγίστων τιμών των δύο πινάκων.
        out3=[A(A>halfadd);B(B>halfadd)]; % Η έξοδος out3 θα περιέχει τα στοιχεία
και των δύο πινάκων A και B (αν υπάρχουν) που είναι μεγαλύτερα από το halffadd.
    else % Στην περίπτωση που οι πίνακες δεν είναι των ίδιων διαστάσεων οι έξοδοι παίρνουν
τους κενούς πίνακες.
        out1=[];
        out2=[];
        out3=[];
    end
else % Στην περίπτωση που οι πίνακες εισόδου είναι των ίδιων διαστάσεων, αλλά όχι
πραγματικοί οι έξοιδοι λαμβάνουν τους κενούς πίνακες.
    out1=[];
    out2=[];
    out3=[];
end
```

..και η κλήση της συνάρτησης γίνεται από το Command Window του Matlab, ως εξής:

>> [X1,X2,X3] = ProjectB2(A,B)

Αν στο Command Window δεν έχουμε ορίσει από πρίν τα A, B δεν θα μας βγάλει αποτέλεσμα η εκτέλεση αυτής της εντολής. Γι'αυτό όταν την εκτελέσουμε θα πρέπει ως εισόδους να βάλουμε πίνακες που έχουμε ορίσει από πρίν στο Command Window, ή απλά να βάλουμε νέους πίνακες, πχ.:

>> [X1,X2,X3] = ProjectB2([2,5,7,22;9,0,7,78], [2,5,7,22;9,0,7,123])

Όπου: A=[2,5,7,22;9,0,7,78] και B=[2,5,7,22;9,0,7,123]

Στην περίπτωση που ως εισόδους θέσουμε μη πραγματικούς πίνακες ή πίνακες που δεν έχουν τις ίδιες διαστάσεις, οι έξοδοι θα αποκτήσουν ως αποτελέσματα τους κενούς πίνακες.

(B3) Η συνάρτηση γράφεται στο Editor του Matlab:

```
function [out] = SecMinInd(A)
SortedA=unique(A); % Δημιουργία μεταβλητής SortedA εσωτερικά της συνάρτησης, η οποία
περιέχει τα στοιχεία του πίνακα A σε άυξουσα σειρά χωρίς επαναλλήψεις.
out=SortedA(2); % Λήψη του δεύτερου στοιχείου του πίνακα SortedA (κατά αρίθμηση MatLab)
το οποίο, εφόσον ο πίνακας είναι ταξινομημένος κατά αύξουσα σειρά, πρέπει να είναι το δεύτερο
μικρότερο στοιχείο του πίνακα.
end
```

..και η κλήση της συνάρτησης γίνεται από το Command Window του Matlab, ως εξής:

```
[out] = SecMinInd(A)
```

Αν στο Command Window δεν έχουμε ορίσει από πρίν τον πίνακα A δεν θα μας βγάλει αποτέλεσμα η εκτέλεση αυτής της εντολής. Γι'αυτό όταν την εκτελέσουμε θα πρέπει ως είσοδο να βάλουμε πίνακα που έχουμε ορίσει από πρίν στο Command Window, ή απλά να τοποθετήσουμε έναν νέο πίνακα πχ.:

>>[out] = SecMinInd([2 3 89 23;34 2 90 8; 2 1 78 34;23 90 3 12; 90 6 56 32; 45 12 90 76])

Όπου: A=[2 3 89 23;34 2 90 8; 2 1 78 34;23 90 3 12; 90 6 56 32; 45 12 90 76]