

## 8η Σειρά Ασκήσεων

### Ασκηση 17.

Αναπτύξτε ένα πρόγραμμα σε γλώσσα Pascal με το οποίο θα ταξινομούνται με την μέθοδο της συγχώνευσης (merge sort) μια σειρά από το πολύ 15 εγγραφές (records) που περιλαμβάνουν προσωπικά στοιχεία φοιτητών (επώνυμο, όνομα και αριθμό μητρώου). Θεωρείστε ότι τα ονόματα δεν είναι μεγαλύτερα από 20 χαρακτήρες. Χρησιμοποιείτε τις ακόλουθες δηλώσεις:

**const**

size = 15; {μέγιστος αριθμός εγγραφών}

**type**

string20 = **Array** [1..20] of char;

student = **record**

Epitheto, Onoma : string20;

AM : integer

**end**;

lista = **Array** [1..size] of student;

Μπορείτε, αν θέλετε, να χρησιμοποιήσετε τον τύπο string της HP Pascal και τις αντίστοιχες συναρτήσεις (κεφ 10.1 των σημειώσεων).

Χρησιμοποιείτε το ακόλουθα δεδομένα:

|                    |              |       |
|--------------------|--------------|-------|
| ΚΩΣΤΟΠΟΥΛΟΣ        | ΝΙΚΟΛΑΟΣ     | 98901 |
| ΔΙΑΜΑΝΤΙΔΗΣ        | ΔΗΜΗΤΡΙΟΣ    | 98902 |
| ΝΙΚΟΛΑΟΥ           | ΠΕΤΡΟΣ       | 98806 |
| ΠΑΠΑΔΙΑΜΑΝΤΗΣ      | ΚΩΝΣΤΑΝΤΙΝΟΣ | 98856 |
| ΚΩΣΤΟΠΟΥΛΟΣ        | ΔΗΜΗΤΡΙΟΣ    | 98847 |
| ΕΥΣΤΡΑΤΙΟΥ         | ΜΑΡΙΑ        | 98790 |
| ΔΗΜΗΤΡΙΟΥ          | ΑΛΕΞΑΝΔΡΑ    | 98791 |
| ΚΩΣΤΟΠΟΥΛΟΥ        | ΙΩΑΝΝΑ       | 98999 |
| ΔΙΑΜΑΝΤΙΔΗΣ        | ΔΗΜΗΤΡΙΟΣ    | 98903 |
| ELGHALI            | ABDULLAH     | 98907 |
| ΠΑΠΑΔΙΑΜΑΝΤΟΠΟΥΛΟΣ | ΑΛΕΞΑΝΔΡΟΣ   | 98990 |

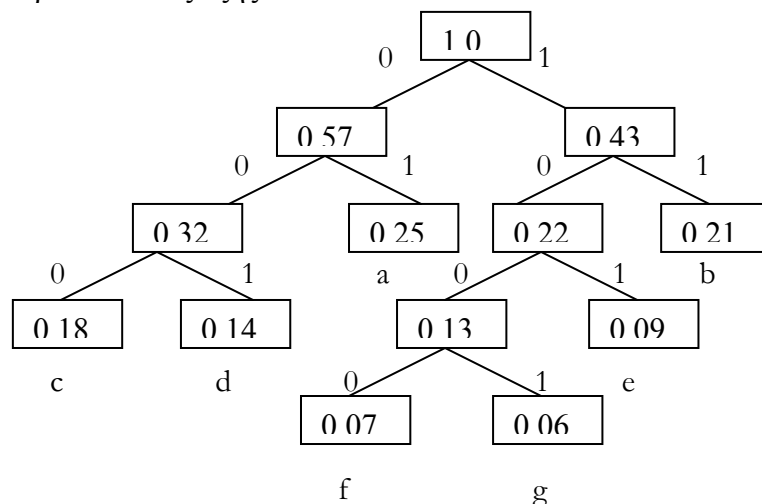
Τα δεδομένα (Επίθετο, Όνομα, Αρ. Μητρώου) θα πρέπει να διαβάζονται από ένα αρχείο κειμένου (Text file) που θα φτιάξετε με το vi. Στο αρχείο αυτό χρησιμοποιείτε μια γραμμή για κάθε φοιτητή. Το αρχείο αυτό θα πρέπει να έχει τουλάχιστον 10 ονόματα. Στη συνέχεια, θα πρέπει να γίνεται αλφαβητική ταξινόμηση των εγγραφών, με βάση το επώνυμο και, αν χρειάζεται, το όνομα και τον Αριθμό Μητρώου. Η ταξινομημένη λίστα εγγραφών θα πρέπει να εμφανίζεται στην οθόνη και να καταχωρείται σε ένα αρχείο εγγραφών (file of student). Το πρόγραμμά σας θα πρέπει τέλος να δίνει τη δυνατότητα στον χρήστη να εκτυπώνει την ταξινομημένη λίστα εγγραφών από το αρχείο εγγραφών που δημιουργήθηκε.

### Άσκηση 18.

Αναπτύξτε ένα πρόγραμμα σε γλώσσα Pascal με το οποίο μία συνδεδεμένη λίστα ακεραίων θα διασπάται σε δύο λίστες: η πρώτη θα περιλαμβάνει τους θετικούς αριθμούς της αρχικής λίστας και η δεύτερη τους αρνητικούς. Η αρχική λίστα θα εισάγεται από το πληκτρολόγιο.

### Άσκηση 19.

Μια από τις εφαρμογές της θεωρίας κωδίκων είναι και η συμπίκνωση κειμένου για οικονομία χώρου κατά την αποθήκευση. Αντί να χρησιμοποιούνται δυαδικές συμβολοσειρές ίδιου μεγέθους για κάθε χαρακτήρα (π.χ. όπως η κωδικοποίηση ASCII), στην κωδικοποίηση Huffman γίνεται χρήση της γνώσης της συχνότητας εμφάνισης κάθε γράμματος (στατιστικά) σε ένα συνηθισμένο κείμενο. Συμφέρει π.χ. ο κωδικός των χαρακτήρων a, e, h, i, n, o, s, t, <space> να είναι πιο σύντομος από τον κωδικό των j, k, q, v, x, z. Αν για παράδειγμα θέλουμε να κωδικοποιήσουμε κείμενο που αποτελείται μόνο a, b, c, d, e, f, g με γνωστές συχνότητες εμφάνισης δηλαδή: 0.25, 0.21, 0.18, 0.14, 0.09, 0.07, 0.06 αντίστοιχα, τότε η οικονομικότερη κωδικοποίηση είναι: 01, 11, 000, 001, 101, 1000, 1001 αντίστοιχα που μπορεί να αναπαρασταθεί ως εξής:



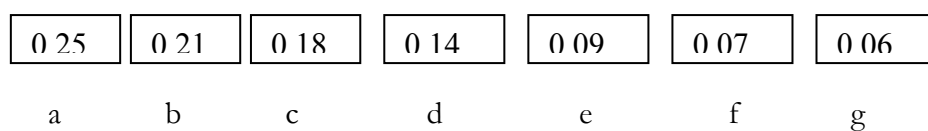
Αυτό το δέντρο Huffman κατασκευάζεται ως εξής: Αρχίζουμε με n δεντράκια ενός κόμβου ταξινομημένα με συχνότητα εμφάνισης σε φθίνουσα σειρά. Μια πρώτη προσέγγιση:

for i:=1 to n-1 do  
begin

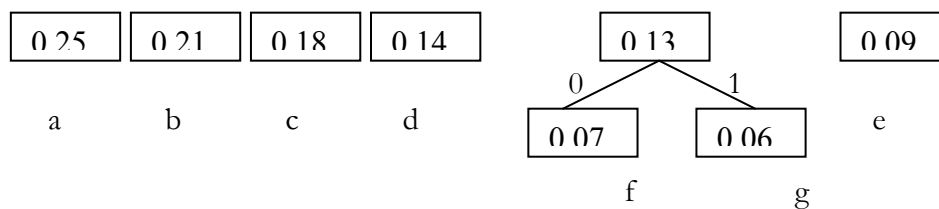
- Merge last two subtrees
- Rearrange subtrees in nonincreasing order of root – probability

end

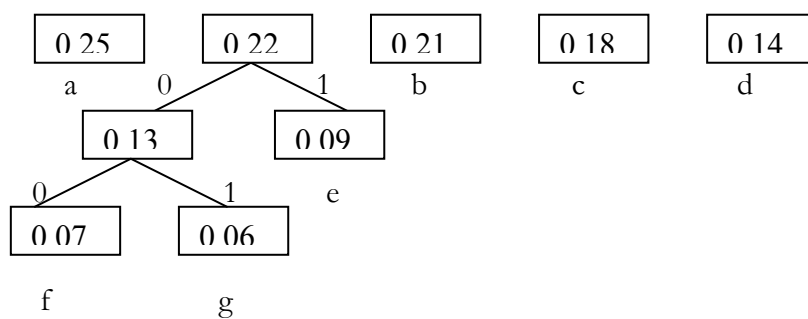
Στο παράδειγμά μας:



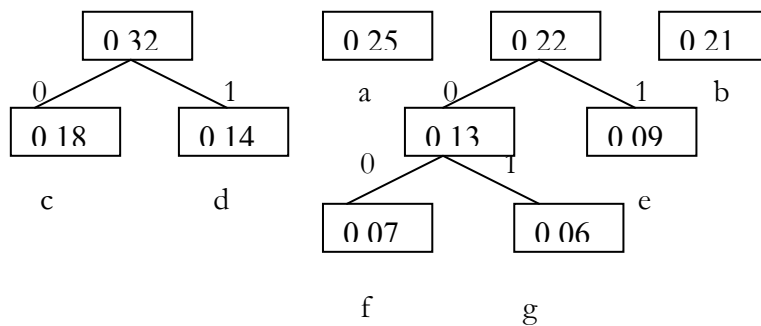
i=1



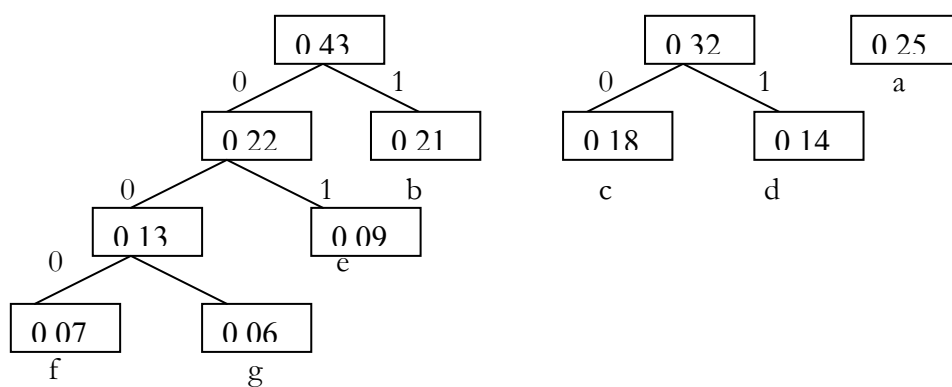
i=2



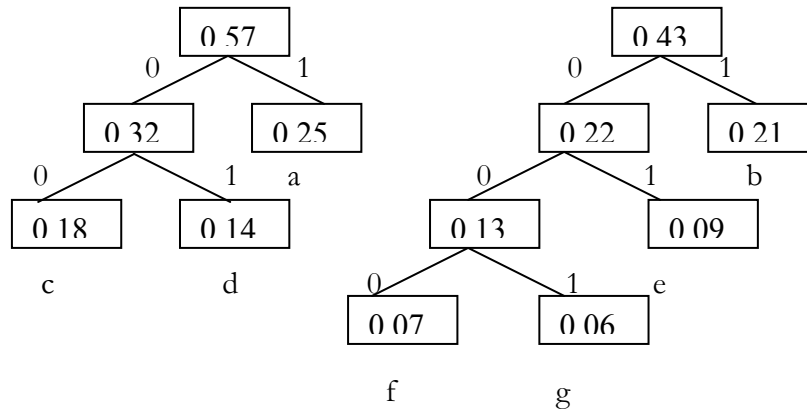
i=3



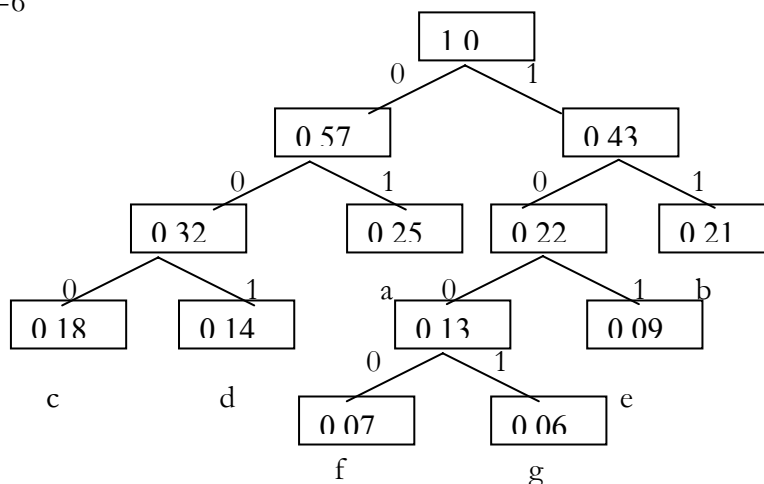
i=4



i=5



i=6



A. Σχεδιάστε ένα πρόγραμμα Pascal που υλοποιεί τον παραπάνω αλγόριθμο κατασκευής δέντρου Huffman. Χρησιμοποιείτε για σύμβολα μόνο γράμματα του Λατινικού αλφαβήτου και το κενό και για συχνότητες αυτές που δίνονται στον επόμενο πίνακα (μέσος όρος εμφάνισης σε αγγλικό κείμενο):

|      |      |      |      |         |      |      |      |      |      |      |
|------|------|------|------|---------|------|------|------|------|------|------|
| a    | b    | c    | d    | e       | f    | g    | h    | i    | j    | k    |
| .065 | .013 | .022 | .032 | .104    | .021 | .015 | .047 | .058 | .001 | .005 |
| l    | m    | n    | o    | p       | q    | r    | s    | t    | u    | v    |
| .032 | .032 | .058 | .064 | .015    | .001 | .049 | .056 | .081 | .023 | .008 |
| w    | x    | y    | z    | <space> |      |      |      |      |      |      |
| .018 | .001 | .017 | .001 | .172    |      |      |      |      |      |      |

B. Προαιρετικά: Σχεδιάστε διαλογικό πρόγραμμα Pascal που να έχει τη δυνατότητα να κωδικοποιήσει και να αποκωδικοποιήσει αγγλικό κείμενο (αγνοώντας σημεία στίξης, κλπ) χρησιμοποιώντας το προηγούμενο πρόγραμμα σας του ερωτήματος A.

*Για εξάσκηση*