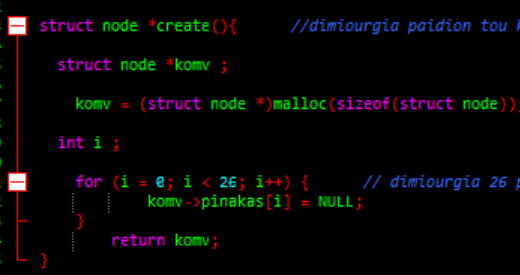
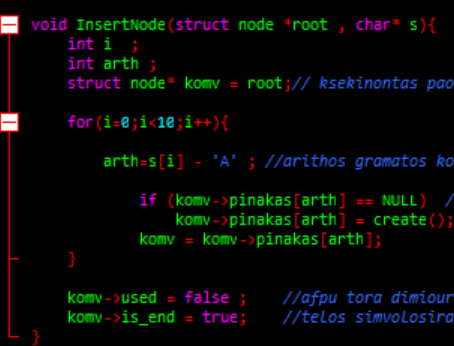
Συναρτήσεις

Συνάρτηση : create

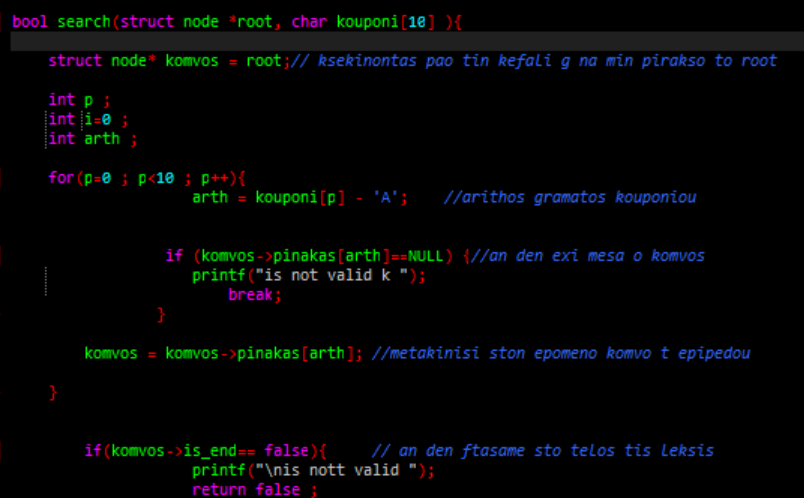
* Δεν φέρνει κάτι για να χρησιμοποιήσει
* Επιστροφή : pointer την διεύθυνση του κόμβου όπου θα δημιουργήσει τα παιδιά

* Με δυναμική δέσμευση δημιουργεί χώρο , τύπου struct node για τον κόμβο
* Στην συνέχεια εντός του for δημιουργία / αρχικοποιεί τα 26 παιδία (όσα τα γράμματα του αλφάβητου ) με ΝULL (γιατί ακόμα δεν έχουμε επιλέξει ποια πρέπει να χρησιμοποιηθούν )
* Στο τέλος επιστρέφει την διεύθυνση του χώρου που μόλις έχω δημιουργήσει



Συνάρτηση : InsertNode

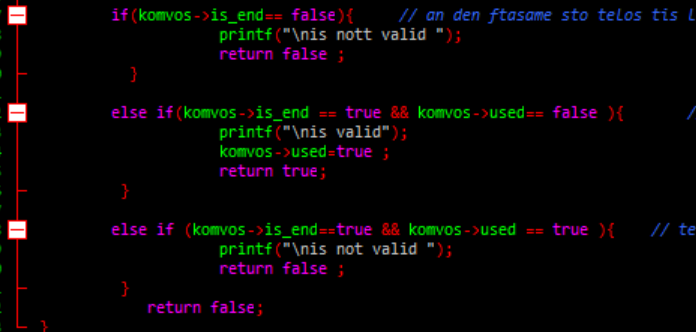
* .Φέρνει το δέντρο μέσα ( root , από την κορυφή ) και την συμβολοσειρά που θέλει να εισάγει στο δέντρο
* .Δεν επιστρέφει κάτι η συνάρτηση
* για να μην επηρεάσω την κάθε διαδρομή / δέντρου με το ίσον αντιγράφω τα χαρακτηριστικά της δημιουργίας του (τους κόμβους )
* στο for κάθε φορά εντάσσω στο δέντρο τον χαρακτήρα που έχει σειρά
* πως;
* με το arth κρατάω τον αριθμό που του αναλογεί .Αν σε εκείνο το αριθμό το παιδί είναι ακόμη null (δλδ δεν έχει παιδιά ) τότε καλώ την συνάρτηση create για να μου δημιουργήσει τα παιδιά του συγκεκριμένου κόμβου
* και στην συνέχει προχωρώ στον κόμβο που δημιούργησα ή προϋπήρχε (komv=komv->pinakas[arth];)
* όταν τελειώσω με την εισαγωγή όλων των γραμμάτων τότε ενημερώνω ότι έφτασα στο τελευταίο γράμμα και ότι ακόμα δεν έχει χρησιμοποιηθεί .



Συνάρτηση : Search

* . Φέρνει το δέντρο με τις πληροφορίες και το κουπόνι που έχουμε διαβάσει .
* . Επιστρέφει true or false (το χρησιμοποιούμε κυρίως για να σταματά την συνάρτηση )

* + Στόχος συνάρτησης : να ψάξει αν υπάρχει το κουπόνι και αν έχει ξανά χρησιμοποιηθεί .
  + Για να μην επηρεάσω την κορυφή της διαδρομής / δέντρου αντιγράφω τα χαρακτηριστικά σε άλλο κόμβους)
  + Δημιουργώ ένα for που αναζητά με την σειρά κάθε γράμμα του κουπονιού αν υπάρχει στο δέντρο .
  + στο κάθε επίπεδο αναζητά αν ο κόμβος που αντιστοιχεί στο γράμμα είναι δημιουργημένος και αν όχι τότε λέει οτι δεν υπάρχει το κουπόνι και βγαίνει από την συνάρτηση.

Στην συνέχεια διενεργώ τους πιο κάτω ελέγχους και στο τέλος κάθε ελέγχου βγαίνω από την συνάρτηση

1. αν δεν έχουμε φτάσει μέχρι το τέλος της διαδρομής τότε ο κόμβος δεν είναι διαθέσιμος .

2. αν είμαστε στο τέλος της διαδρομής και δεν έχει χρησιμοποιηθεί τότε ο κόμβος είναι διαθέσιμος .

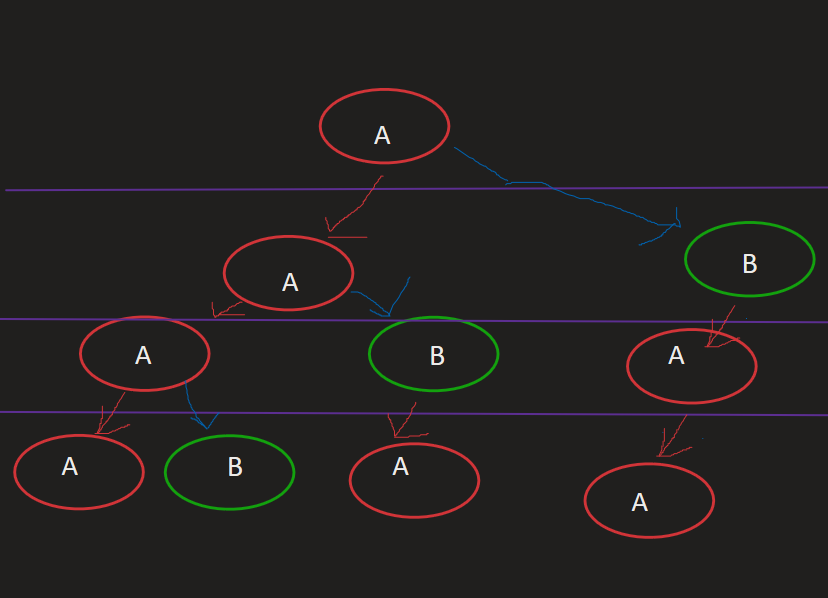
3. αν είμαστε στο τέλος της διαδρομής και ο κόμβος έχει χρησιμοποιηθεί τότε ο κόμβος δεν είναι διαθέσιμος .

Διαδικασία

1.ΑΑΑΑ

* υπάρχει ο κόμβος Α ; όχι δημιουργώ Α
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά πάω αριστερά
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά πάω αριστερά
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά

2.ΑΑΑΒ

* υπάρχει ο κόμβος Α ; ναι χρησιμοποιώ Α
* υπάρχει παιδί Α ; ναι ,χρησιμοποιώ Α , πάω αριστερά για το επόμενο
* υπάρχει παιδί Α ; ναι ,χρησιμοποιώ Α , πάω δεξιά για το επόμενο
* υπάρχει παιδί Β ; όχι δημιουργώ Β δεξιά

3. ΑΑΒΑ

* υπάρχει ο κόμβος Α ; ναι χρησιμοποιώ Α
* υπάρχει παιδί Α ; ναι ,χρησιμοποιώ Α , πάω δεξιά για το επόμενο
* υπάρχει παιδί Β ; όχι δημιουργώ Β δεξιά , πάω αριστερά για το επόμενο
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά

4. ΑΒΑΑ

* υπάρχει ο κόμβος Α ; ναι χρησιμοποιώ Α
* υπάρχει παιδί Β ; όχι δημιουργώ Β δεξιά , πάω αριστερά για το επόμενο
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά πάω αριστερά
* υπάρχει παιδί Α ; όχι δημιουργώ Α αριστερά