Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

HW1 - Trie (ατομική εργασία)

Ημερομηνία παράδοσης: Τετάρτη 26 Φεβρουαρίου

Σε αυτή την εργασία καλείστε να γράψετε ένα πρόγραμμα σε Java το οποίο δημιουργεί ένα ασυμπίεστο Trie για να αποθηκευτούν λέξεις από το αγγλικό αλφάβητο. Το Trie που θα δημιουργήσετε θα το γεμίσετε με λέξεις από ένα αρχείο-λεξικό που περιέχει λέξεις του αγγλικού αλφαβήτου. Θα σας δοθεί κώδικας ο οποίος διαβάζει το αρχείο και επιστρέφει το περιεχόμενο του με τη μορφή ενός πίνακα από java.lang.String.

Αφού δημιουργήσετε το Trie το πρόγραμμα έχει τις παρακάτω διαφορετικές επιλογές:

A. εκτύπωση του αριθμού των λέξεων που περιέχει το Trie
B. αναζήτηση λέξης στο Trie
C. εκτύπωση του Trie στο stdout κατά pre-order
D. εκτύπωση του Trie σε αρχείο και σε μορφή που να διαβάζεται και να μετασχηματίζεται από το πρόγραμμα graphviz/dot.
E. εξαγωγή των λέξεων του Trie που απέχουν N χαρακτήρες από μία δοθείσα λέξη.
F. εξαγωγή των λέξεων του Trie που έχουν ως ρίζα μία δοθείσα λέξη.

Βοηθητικός κώδικας

Σας παρέχεται ο κώδικας της κλάσης **ce326.hw1.Utilities** η οποία περιέχει στατικές μεθόδους για την υλοποίηση των παρακάτω λειτουργιών.

Διάβασμα αρχείου λέξη προς λέξη

Η στατική μέθοδος **readFile** διαβάζει ένα αρχείο λέξη-λέξη και επιστρέφει τις λέξεις του διάβασε σε ένα πίνακα από **java.lang.String**. Ο πίνακας δεν είναι απαραίτητο να περιέχει τόσες λέξεις όσο είναι το μέγεθος του. Μπορεί να περιέχει λιγότερες λέξεις από το μέγεθος του και σε αυτή την περίπτωση οι τελευταίες θέσεις του πίνακα δεν δείχνουν σε αντικείμενα τύπου **java.lang.String**, αλλά σε **null**.

Εκτύπωση ενός java.lang.String σε αρχείο

Η κλάση **ce325.hw1.Utilities** περιέχει τη στατική μέθοδο **writeFile** για την εκτύπωση ενός String σε αρχείο του οποίο το path λαμβάνεται επιπλέον όρισμα. Η συνάρτηση επιστρέφει true εάν η εγγραφή έγινε με επιτυχία διαφορετικά false.

Μετασχηματισμός ενός αρχείου dot σε αρχείο εικόνας τύπου png

Η κλάση **ce325.hw1.Utilities** περιέχει τη στατική μέθοδο dot2png για το μετασχηματισμό ενός αρχείου dot σε png (με τη βοήθεια του προγράμματος graphviz/dot)

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

Η κλάση ce326.hw1.TrieNode

Η κλάση **ce326.hw1.TrieNode** συμβολίζει τον κόμβο του Trie, ο οποίος έχει κατ' ελάχιστον τα εξής πεδία (μπορείτε να προσθέσετε και δικά σας):

`class TrieNode {`
`  TrieNode children [];`
`  boolean isTerminal;`
`}`
	`children: πίνακας από 26 δείκτες (όσα και τα γράμματα του αλφαβήτου) προς τα παιδιά του κόμβου.`
	
	`isTerminal: Σημειώνεται κατά πόσο ο κόμβος αποτελεί το τέλος μιας λέξης ή όχι.`

Οι κατασκευαστές και οι μέθοδοι της κλάσης TrieNode αποτελούν δική σας επιλογή.

Η κλάση ce326.hw1.Trie

Πρόκειται για την κλάση του **Trie**, η οποία κατ' ελάχιστο πρέπει να έχει τα εξής πεδία:

`class Trie {`
 	
`  TrieNode root;`
 	
`  public static final String alphabet = "abcdefghijklmnopqrstuvwxyz";`
 	
`}`
	`root: η ρίζα του Trie`
	
	`alphabet: σταθερά που αντιπροσωπεύει το αλφάβητο του Trie.`

και τις public κατασκευαστές και μεθόδους :

Μέθοδος	Περιγραφή
`Trie(String []words)`	Κατασκευαστής της κλάσης. Λαμβάνει ως όρισμα ένα πίνακα από λέξεις, τις οποίες θα καταχωρήσει στο Trie. Ο πίνακας μπορεί να περιέχει εγγραφές null στις τελευταίες θέσεις του.
`boolean add(String word)`	Αναζητά τη λέξη word στη δομή και εάν δεν υπάρχει την προσθέτει. Επιστρέφει true σε περίπτωση επιτυχούς εισαγωγής διαφορετικά false.
`boolean contains(String word)`	Επιστρέφει true εάν υπάρχει η λέξη στη δομή διαφορετικά false.
`int size()`	Επιστρέφει τον αριθμό των καταχωρημένων λέξεων στο Trie.
`String[] differByOne(String word)`	Επιστρέφει ένα πίνακα από λέξεις που έχουν το ίδιο μήκος με τη δοθείσα word και διαφέρουν κατά ένα γράμμα. Η μέθοδος δεν εξετάζει εάν η λέξη υπάρχει ή όχι στο Trie.
`String[] differBy(String word, int max)`	Επιστρέφει ένα πίνακα από λέξεις που έχουν το ίδιο μήκος με τη δοθείσα word και διαφέρουν από ένα έως και **max** χαρακτήρες. Η μέθοδος δεν εξετάζει εάν η λέξη υπάρχει ή όχι στο Trie. **Σημείωση:** Εάν δεν μπορείτε να υλοποιήσετε τη γενική μέθοδο differBy υλοποιήστε την πιο ειδική differByOne.
`String toString()`	Επιστρέφει ένα String που αντιστοιχεί στην pre-order διαπέραση του Trie. Κάθε κόμβος του Trie αντιστοιχεί σε ένα πεζό γράμμα του αγγλικού αλφαβήτου. Λεπτομέρειες τη μορφή εκτύπωσης κατά τη διαπέραση pre-order μπορείτε να βρείτε στο Παράρτημα 3.
`String toDotString()`	Επιστρέφει ένα String το οποίο περιγράφει το Trie σε μορφή αναγνώσιμη από το πρόγραμμα graphviz/dot. Λεπτομέρειες για τη

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

	μορφή του Strring μπορείτε να βρείτε στο Παράρτημα 2.
`String[] wordsOfprefix(String prefix)`	Επιστρέφει ένα πίνακα από λέξεις που έχουν ως κοινό πρόθεμα το String prefix.

Σημείωση: Όλες οι αναζητήσεις πρέπει να γίνονται πάνω στη δομή Trie και ΟΧΙ χρησιμοποιώντας βοηθητικές δομές δεδομένων. Για παράδειγμα η εύρεση των λέξεων που απέχουν έως Ν χαρακτήρες από τη δοθείσα δεν είναι επιτρεπτό να γίνει εξάγωντας όλες τις λέξεις ίδιου μήκους με τη δοθείσα σε μία λίστα ή πίνακα και αναζήτηση των λέξεων που ενδιαφέρουν πάνω στη λίστα ή των πίνακα.

Η κλάση ce326.hw1.HW1

Η κλάση σας δίνεται έτοιμη και δεν πρέπει να την αλλάξετε. Περιέχει τη στατική μέθοδο main από την οποία εκκινεί το πρόγραμμα σας. Το πρόγραμμα εκκινεί λαμβάνοντας δύο έως τέσσερα ορίσματα από τη γραμμή εντολών. Τα δύο πρώτα ορίσματα είναι σε όλες τις περιπτώσεις τα εξής:

- **1ο όρισμα:** το όνομα του αρχείου λεξικού από το οποίο θα διαβαστούν οι λέξεις που θα αποτελέσουν το περιεχόμενο του Trie και
- **2ο όρισμα:** μία από τις λέξεις print_preorder, print_dot, distant_words, prefixed_words που αποτελούν τις εντολές λειτουργίας του προγράμματος.

Το πρόγραμμα αφού καλέσει τη στατική συνάρτηση **ce325.hw1.Utilities.readFile** με παράμετρο το πρώτο όρισμα της γραμμής εντολών, χρησιμοποιεί τον πίνακα από Strings που επιστρέφει η συνάρτηση, προκειμένου να δημιουργήσει ένα ασυμπίεστο Trie όπου κάθε κόμβος έχει 26 δείκτες προς τα παιδιά του.

Οι λειτουργίες της συνάρτησης main είναι οι εξής:

- **size:** Δεν λαμβάνει επιπλέον ορίσματα. Εκτυπώνει των αριθμό των λέξεων που περιέχει το Trie.
- **contains_word:** Λαμβάνει ως **3ο όρισμα** μία λέξη την οποία και αναζητά μέσα στο Trie. Εάν τη βρει εκτυπώνει "`W found`", διαφορετικά εκτυπώνει "`W not found`", όπου W η λέξη που δόθηκε ως 3ο όρισμα.
- **print_preorder:** Δεν λαμβάνει επιπλέον ορίσματα. Εκτυπώνεται στο stdout χρησιμοποιώντας τη μέθοδο toString.
- **print_dot:** Δεν λαμβάνει επιπλέον ορίσματα. Με τη βοήθεια της μεθόδου toDotString της κλάσης Trie, λαμβάνεται ένα String το οποίο περιγράφει το Trie σε μορφή αναγνώσιμη από το πρόγραμμα graphviz/dot. Με τη βοήθεια της μεθόδου Utilities.writeFile το πρόγραμμα γράφεται σε ένα αρχείο με κατάληξη .**dot** , το οποίο στη συνέχεια μετασχηματίζεται σε εικόνα PNG μέσω της μεθόδου Utilities.dot2png, προκειμένου να μπορείτε να εξετάσετε τη δομή του Trie βλέποντας την παραγόμενη εικόνα.

  Σημείωση: Η μέθοδος **dot2png** λαμβάνει ως όρισμα το όνομα του αρχείου χωρίς την κατάληξη **.dot** στο τέλος και παράγει ένα αρχείο PNG με το ίδιο όνομα και κατάληξη **.png** (δείτε το σχετικό κώδικα της μεθόδου).

  Λεπτομέρειες για την μορφή της εικόνας που παράγεται από το πρόγραμμα dot μπορείτε να βρείτε στο Παράρτημα 1. Λεπτομέρειες για τη μορφή του αρχείου dot μπορείτε να βρείτε στο Παράρτημα 2.

- **distant_words:** Λαμβάνει δύο επιπλέον ορίσματα:

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

    - **3ο όρισμα:** έναν θετικό ακέραιο **N** που περιγράφει τη μέγιστη απόσταση των αναζητούμενων λέξεων από τη δοθείσα και
    - **4ο όρισμα:** τη δοθείσα λέξη με βάση την οποία θα γίνει η αναζήτηση.

  Εντοπίζει τις λέξεις του Trie που διαφέρουν από ~~0~~1 έως και N γράμματα από την αρχική λέξη. Εκτυπώνει στην οθόνη τις λέξεις που εντόπισε με λεξικογραφική σειρά χωριζόμενες μεταξύ τους με τον χαρακτήρα αλλαγής γραμμής.

- **prefixed_words:** λαμβάνει ως **3ο όρισμα** μία λέξη η οποία αποτελεί το πρόθεμα των λέξεων που θα αναζητηθούν στο Trie. Εκτυπώνονται με λεξικογραφική σειρά οι λέξεις με το συγκεκριμένο πρόθεμα που περιέχονται στο Trie. Εάν το αλφαριθμητικό prefix υπάρχει ως αυτόνομη λέξη στο λεξικό, εκτυπώνεται και η συγκεκριμένη λέξη.

Διευκρίνιση για την αναζήτηση λέξεων στο Trie με απόσταση N χαρακτήρες

Ας υποθέσουμε ότι έχετε ένα λεξικό με τις λέξεις **aaa, aac, aab, aca, baa, bca** και αναζητούμε για την λέξη **aab** τις λέξεις του λεξικού που απέχουν ένα χαρακτήρα. Οι λέξεις αυτές είναι οι **aaa, aac** ~~**και aab**~~. Η λέξη **aca** διαφέρει κατά δύο χαρακτήρες (στις θέσεις 1 και 2), η λέξη **baa** διαφέρει κατά δύο χαρακτήρες (στις θέσεις 0 και 2) και η λέξη **bca** διαφέρει κατά 3 χαρακτήρες (σε όλες τις θέσεις υπάρχουν διαφορές).

Οδηγίες Αποστολής

Δημιουργία password στο autolab

Το βήμα αυτό θα το κάνετε μόνο εάν δεν έχετε συνδεθεί ξανά στο autolab ή έχετε ξεχάσει το password σας. Το ίδιο username/password ισχύει για όλες τις εργασίες.

- Πηγαίνετε στη διεύθυνση https://autolab.e-ce.uth.gr/auth/users/sign_in (απαιτείτα VPN για να συνδεθείτε από το σπίτι) και επιλετε "Forgot your password?"
- Στη σελίδα που εμφανίζεται βάζετε το e-mail σας στο Πανεπιστήμιο θεσσαλίας (υποχρεωτικά με κατάληξη `@uth.gr`) και πατάτε "SEND ME REQUEST PASSWORD INSTRUCTIONS.
- Λαμβάνετε ένα e-mail με θέμα "Reset password instructions" στον προσωπικό σας λογαριασμό που περιέχει ένα σύνδεσμο. Πατάτε το σύνδεσμο και βάζετε ένα συνθηματικό της αρεσκείας σας. Αποθηκεύστε αυτό το συνθηματικό για όλες τις εργασίες.
- Στη σελίδα που εμφανίζεται στη συνέχεια, συμπληρώστε στο πεδίο nickname το ΑΕΜ σας και πατήστε υποβολή.
- Συνεχίστε στην υποβολή της 1ης εργασίας.

Υποβολή της 1ης εργασίας

Δημιουργήστε ένα κατάλογο (directory) με όνομα hw1submit και μέσα εκεί τοποθετήστε τα αρχεία **`Trie.java`** και **`TrieNode.java`** (αυστηρά με αυτά τα ονόματα). Συμπιέστε τον κατάλογο αυτό σε μορφή **zip**. Παράγεται ένα αρχείο με όνομα **`hw1submit.zip`** το οποίο θα υποβάλλετε στο autolab.

Το αρχείο **hw1submit.zip** το υποβάλλετε στο autolab ως εξής:

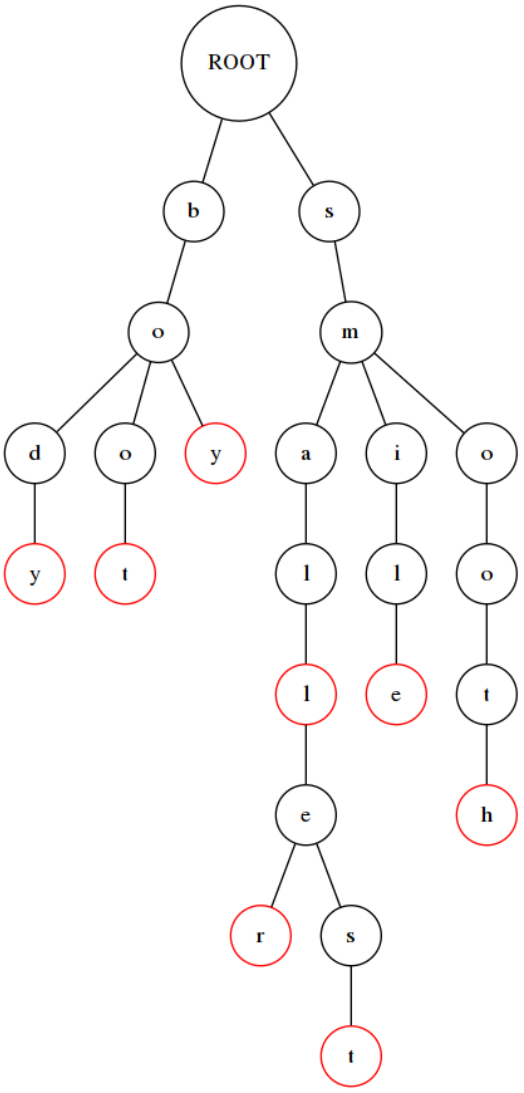
- Κάνετε login στο autolab και επιλέγετε το μάθημα "**CE325_2019-2020 (S20)**" και συγκεκριμένα το assignment **HW1**.
- Επιλέγετε το checkbox "I affirm that I have complied ...", πατάτε το κουμπί SUBMIT και επιλέγετε το αρχείο **hw1submit.zip**. Περιμένετε λίγο και βλέπετε το score σας.

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

Παράρτημα 1 - Γραφική απεικόνιση του Trie

Παρακάτω δίνεται η γραφική αναπαράσταση της δομής λεξικού για το παρακάτω σύνολο λέξεων: **body, boot, boy, small, smaller, smallest, smile, smooth,.** Με κόκκινο χρώμα σημειώνονται οι τερματικοί κόμβοι της δομής. Παρατηρήστε τα εξής:

- Παρατηρήστε ότι οι λέξεις **body, boot** και **boy** έχουν το κοινό μονοπάτι **bo**. Επίσης η λέξη **small** αποτελεί τμήμα των λέξεων **smaller** και **smallest**.
- Με κόκκινο χρώμα εμφανίζονται οι τερματικοί κόμβοι δηλαδή οι κόμβοι που αντιστοιχούν στο τέλος μιας λέξης.



Παράρτημα 2 - Το αρχείο dot που παράγει το παραπάνω σχήμα

Προκειμένου να απεικονίσετε το συγκεκριμένο Trie θα χρειαστεί να φτιάξετε ένα αρχείο txt κατάλληλο για ανάγνωση από το πρόγραμμα graphviz/dot (βάλτε κατάληξη .dot στο αρχείο που θα φτιάχνετε για να το ξεχωρίζετε από άλλα αρχεία). Ένα ενδεικτικό αρχείο για το παραπάνω δέντρο είναι το εξής:

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

```
graph Trie {
	142257191 [label="ROOT" ,shape=circle, color=black]
	142257191 -- 1044036744
	1044036744 [label="b" ,shape=circle, color=black]
	1044036744 -- 999966131
	999966131 [label="o" ,shape=circle, color=black]
	999966131 -- 1989780873
	1989780873 [label="d" ,shape=circle, color=black]
	1989780873 -- 1480010240
	1480010240 [label="y" ,shape=circle, color=red]
	999966131 -- 81628611
	81628611 [label="o" ,shape=circle, color=black]
	81628611 -- 1828972342
	1828972342 [label="t" ,shape=circle, color=red]
	999966131 -- 1452126962
	1452126962 [label="y" ,shape=circle, color=red]
	142257191 -- 931919113
	931919113 [label="s" ,shape=circle, color=black]
	931919113 -- 1607521710
	1607521710 [label="m" ,shape=circle, color=black]
	1607521710 -- 764977973
	764977973 [label="a" ,shape=circle, color=black]
	764977973 -- 381259350
	381259350 [label="l" ,shape=circle, color=black]
	381259350 -- 2129789493
	2129789493 [label="l" ,shape=circle, color=red]
	2129789493 -- 668386784
	668386784 [label="e" ,shape=circle, color=black]
	668386784 -- 1329552164
	1329552164 [label="r" ,shape=circle, color=red]
	668386784 -- 363771819
	363771819 [label="s" ,shape=circle, color=black]
	363771819 -- 2065951873
	2065951873 [label="t" ,shape=circle, color=red]
	1607521710 -- 1791741888
	1791741888 [label="i" ,shape=circle, color=black]
	1791741888 -- 1595428806
	1595428806 [label="l" ,shape=circle, color=black]
	1595428806 -- 1072408673
	1072408673 [label="e" ,shape=circle, color=red]
	1607521710 -- 1531448569
	1531448569 [label="o" ,shape=circle, color=black]
	1531448569 -- 1867083167
	1867083167 [label="o" ,shape=circle, color=black]
	1867083167 -- 1915910607
	1915910607 [label="t" ,shape=circle, color=black]
	1915910607 -- 284720968
	284720968 [label="h" ,shape=circle, color=red]
}
```

Επεξήγηση του αρχείου:

1. Το αρχείο πρέπει να ξεκινά με το αλφαριθμητικό **graph Trie** και να ακολουθούν άγκιστρα μέσα στα οποία περιέχεται η πληροφορία του δέντρου-γράφου που θα σχεδιαστεί από το πρόγραμμα. Στο τέλος κλείνουν τα άγκιστρα.
2. Κάθε κόμβος πρέπει να έχει ένα μοναδικό αναγνωριστικό. Αυτό μπορεί να είναι οτιδήποτε (αριθμός, αλφαριθμητικό κλπ). Επειδή έχουμε κόμβους που απεικονίζουν το ίδιο γράμμα σε

Σημείωση: Ομοιότητα με εργασίες συναδέλφων σας σημαίνει τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων ατόμων ή ομάδων.

διαφορετικά σημεία του γράφου δεν μπορεί να χρησιμοποιηθεί το γράμμα που αντιστοιχεί στον κόμβο ως μοναδικό αναγνωριστικό. Κατά συνέπεια, ως μοναδικό αναγνωριστικό επιλέχθηκε ο αριθμός που επιστρέφει η συνάρτηση hashCode(). Η συνάρτηση υπάρχει σε όλα τα αντικείμενα και επιστρέφει ένα μοναδικό αριθμό για κάθε αντικείμενο μέσα σε ένα πρόγραμμα.

3. Η δήλωση `1044036744 [label="b" ,shape=circle, color=black]` σηματοδοτεί τη δήλωση ενός κόμβου με μοναδικό αναγνωριστικό **1044036744**, ο οποίος θα είναι στρογγυλός, θα έχει μαύρο χρώμα και μέσα θα περιέχει το κείμενο **"b"**.
4. Η δήλωση `1867083167 -- 1915910607` σηματοδοτεί ότι ο κόμβος `1867083167` συνδέεται με τον κόμβο `1915910607`. Η δήλωση σύνδεσης ενός κόμβου μπορεί να προηγείται της δήλωσης του κόμβου (η σειρά με την οποία εμφανίζονται οι κόμβοι και οι συνδέσεις τους δεν επηρεάζει το αποτέλεσμα).

Για να εκτελέσετε το πρόγραμμα dot από την γραμμή εντολών ώστε από το παραπάνω αρχείο να παραχθεί μία εικόνα png αρκεί να γράψετε στο τερματικό του υπολογιστή σας.

```
dot -Tpng file.dot -o file.png
```

`dot` `-Tpng` `file.dot` `-o file.png`

Όπου dot το πρόγραμμα που θα εκτελεστεί, -Tpng το format στο οποίο θα βγει η εικόνα (υπάρχουν και άλλες επιλογές), file.dot το αρχείο κειμένου της παραπάνω μορφής με κατάληξη dot και -o file.png το αρχείο εικόνας μορφής PNG που θα παραχθεί. Δοκιμάστε την εντολή στον υπολογιστή σας.

Η παραπάνω λειτουργία μετατροπής ενός αρχείου κειμένου με κατάληξη dot σε εικόνα PNG υλοποιείται από τη στατική συνάρτηση **dot2png** της κλάσης **Utilities**. Προσοχή: Η συνάρτηση **dot2png** λαμβάνει ως όρισμα το όνομα του αρχείου χωρίς την κατάληξη dot στο τέλος. Και παράγει ένα αρχείο PNG με το ίδιο όνομα και κατάληξη **.png** (δείτε τον σχετικό κώδικα).

Παράρτημα 3 - Pre-Order διαπέραση του Trie

Η pre-order διαπέρεση του Trie πρέπει να διέπεται από τους εξής κανόνες:

1. Εκτυπώνει όλο το δέντρο σε μία γραμμή, δηλαδή δεν παρεμβάλλεται χαρακτήρας αλλαγής γραμμής ανάμεσα στην εκτύπωση δύο διαδοχικών κόμβων και εκτυπώνεται χαρακτήρας αλλαγής γραμμής μετά από την εκτύπωση του τελευταίου κόμβου.
2. Πριν από κάθε κόμβο εκτυπώνεται ένας κενός χαρακτήρας.
3. Κάθε κόμβος πρέπει να απεικονίζεται με το πεζό γράμμα στο οποίο αντιστοιχεί. Επιπλέον, οι τερματικοί κόμβοι απεικονίζονται από τον χαρακτήρα '!' αμέσως μετά τον πεζό χαρακτήρα (χωρίς κενό ενδιάμεσα).

Η pre-order διαπέραση για το παραπάνω παράδειγμα Trie είναι το εξής:

```
b o d y! o t! y! s m a l l! e r! s t! i l e! o o t h!
```