

ΕΡΓΑΣΙΑ 1^η

ΜΕΛΗ:

ΣΩΤΗΡΗΣ ΠΑΝΟΥΡΓΙΑΣ 3200128

ΘΕΟΦΑΝΗΣ ΠΑΠΑΔΟΠΟΥΛΟΣ 3200135

Μερος Α

Η κλάση `StringDoubleEndedQueueImpl` υλοποιεί τη διεπαφή `StringDoubleEndedQueue`. Αυτή η διεπαφή ορίζει τις βασικές λειτουργίες ενός διπλά συνδεδεμένου ουράς (double-ended queue) που περιέχει στοιχεία τύπου `String`. Ας αναλύσουμε τον κώδικα και τις λειτουργίες που προσφέρει:

1. Πεδία κλάσης
 - ☐ `Node first` : Πρωτος κομβος ουρας
 - ☐ `Node last` : Τελευταιος κομβος ουρας
 - ☐ `Int size` : Μεγεθος Ουρας
2. Κλαση `Node`: αναπαριστα έναν `String` κομβο της ουρας και συνδεεται με τον προηγουμενο και τον επομενο κομβο
3. Μεθοδος `isEmpty()` : επιστρεφει `true` αν η ουρα είναι κενη και αν όχι `false`
4. Μεθοδος `AddFirst AddLast` : Προσθετουν έναν κομβο στην αρχη και στο τελος αντιστοιχα
5. Μεθοδος `removeFirst removeLast` : αφαιρουν και επιστρεφουν το πρωτο και το τελευταιο στοιχειο της ουρας αντιστοιχα
6. Μεθοδος `getFirst getLast`: επιστρεφουν το πρωτο και το τελευταιο στοιχειο της ουρας αντιστοιχα
7. Μεθοδος `PrintQueue`: εκτυπωνει τα στοιχεια της ουρας
8. Μεθοδος `size` : επιστεφει το μεγαθος της ουρας

Μερος Β

Στο 2^ο μέρος καλούμαστε να χρησιμοποιήσουμε την παραπάνω υλοποίηση για να μετατρέψουμε μια αλγεβρική παρασταση από μεταθεματική σε ενθεματική μορφή.

Αρχικά διαβαζουμε την μεταθεματική μορφή της παραστασης ελέγχουμε αν αυτό που διαβάσαμε είναι null και με την μεθοδο `convPrefixToInfix` που δεχεται ένα string και επιστρέφει μια ουρα μήκους 1 με τον string με την ενθεματική μορφή.

Μεσα στην συναρτηση ελέγχουμε αν η παρασταση είναι σε σωστή μορφή αν είναι τότε προχωράμε στην μετατροπή περνοντας από όλα τα στοιχεία της παραστασης αν το συμβολο είναι ψηφιο τότε το κανουμε `addLast` , ενώ αν είναι συμβολο(+-*/)τότε κανουμε `removeLast` 2 στοιχεία και φτιαχνουμε το αποτελεσμα και το κανουμε `addLast` στην ουρα. Τελος ελέγχουμε αν η ουρα είναι μήκους 1 αν δεν είναι τότε η παρασταση θα είναι σε λάθος μορφή και επιστεφουμε το καταλληλο μηνυμα αλλιως επιστεφουμε την ουρα και με το `PrintQueue` την εμφανιζουμε σε μεταθεματική μορφή

Μερος Γ

Στο 3^ο μέρος καλούμαστε να χρησιμοποιήσουμε την παραπάνω υλοποίηση για να ελέγξουμε μια ακολουθία DNA αν είναι Watson. Στην αρχή διαβαζουμε την ακολουθία μέχρι να δωθεί ακολουθία σε σωστή μορφή, με βάση την συνάρτηση `isValidDNASequence` η οποία ελέγχει αν η ακολουθία είναι null, κενό και αν κάθε χαρακτήρας είναι εγκυρός και τότε δημιουργούμε μια ουρά προσθέτοντας τα στοιχεία της ακολουθίας μέσα με την συνάρτηση `CreateQ`.

Μετά για να γίνει ο έλεγχος καλούμε την συνάρτηση `isWatson`

Η οποία ελέγχει αν είναι σωστά αυτό γίνεται ελεγχοντας το 1^ο και το τελευταίο στοιχείο της λίστας αν είναι σωστά με βάση τους νόμους δηλαδή αν το 1^ο στοιχείο είναι A το τελευταίο να είναι T ώστε η συμπλήρωση τους να είναι βγαζει την ακολουθία αναποδα. Αν όλα είναι με βάση τους κανόνες που ελέγχουμε κάνουμε `return true` αλλιώς `return false`