ΥΛΟΠΟΙΗΣΗ ΔΙΕΠΑΦΗΣ Α : Αρχικά δηλώσαμε τέσσερις ιδιωτικές μεταβλητές(τις μεταβλητές firstNode και lastNode από private access τις τροποποιήσαμε σε protected ώστε να έχουμε πρόσβαση στα δεδομένα τους από την κλάση DNAPalindrome του ερωτήματος Β).Αυτές ειναι η listName(για το όνομα της λίστας),οι firstNode-lastNode(για τον πρώτο και τελευταίο κόμβο της λίστας) και η sizeQueue(για το μέγεθος της λίστας).Στη συνέχεια, δημιουργήσαμε δύο κατασκευαστές για κάθε υλοποίηση.Έναν default constructor και έναν one-argument constructor.Ο πρώτος καλεί τον δεύτερο με προκαθορισμένο όρισμα το List για το όνομα της λίστας.Ο δεύτερος παίρνει ως όρισμα το όνομα της λίστας,αρχικοποιεί τον πρώτο, τον τελευταίο κόμβο σε null και το μέγεθος της λίστας.Ακολουθούν εννιά μέθοδοι,οι τέσσερις από αυτές ειναι για εισαγωγή και διαγραφή στοιχείου.Οι addFirst(char),removeFirst(),addLast(char) και removeLast() για εισαγωγή στην άρχη,διαγραφή του πρώτου στοιχείου,εισαγωγή στο τέλος και διαγραφή του τελευταίου στοιχείου αντίστοιχα.Μετά από κάθε λειτουργία ενημερώνεται το μέγεθος της λίστας και αν πραγματοποιήθηκε διαγραφή επιστρέφεται το στοιχείου που διαγράφηκε.'Επειτα είναι οι getFirst() και getLast() οι οποίες επιστρέφουν το πρώτο και το τελευταίο στοιχείο αντίστοιχα,χωρίς να διαγραφεί.Τέλος,εχουμε τις printQueue(PrintStream), size(), isEmpty().Η πρώτη τυπώνει τα στοιχεία της λίστας,η δεύτερη επιστρέφει το μέγεθος της και η τρίτη ελέγχει αν η λίστα είναι άδεια επιστρέφοντας true ή false.Η υλοποίηση ολοκληρώνεται με την δημιουργία της main.Στη main δημιουργούμε ένα στιγμιότυπο της διεπαφής Α με το οποίο χειριζόμαστε τη λίστα.Πραγματοποίουμε μερικές εισαγωγές στοιχείων και επείτα μερικές διαγραφές ελέγχοντας για exceptions.

ΥΛΟΠΟΙΗΣΗ ΔΙΕΠΑΦΗΣ Γ : Αρχικά δηλώσαμε τέσσερις ιδιωτικές μεταβλητές.Αυτές ειναι η listName(για το όνομα της λίστας),οι head-tail(για τον πρώτο και δεύτερο κόμβο της λίστας) και η sizeQueue(για το μέγεθος της λίστας).Στη συνέχεια, δημιουργήσαμε δύο κατασκευαστές για κάθε υλοποίηση.Έναν default constructor και έναν one-argument constructor.Ο πρώτος καλεί τον δεύτερο με προκαθορισμένο όρισμα το List για το όνομα της λίστας.Ο δεύτερος παίρνει ως όρισμα το όνομα της λίστας,αρχικοποιεί τον πρώτο, τον τελευταίο κόμβο σε null και το μέγεθος της λίστας.Ακολουθούν έξι μέθοδοι,οι δύο από αυτές ειναι για εισαγωγή και διαγραφή στοιχείου.Οι put(char),get() για εισαγωγή στοιχείου και διαγραφή στοιχείου αντίστοιχα.Μετά από κάθε λειτουργία ενημερώνεται το μέγεθος της λίστας και αν πραγματοποιήθηκε διαγραφή επιστρέφεται το στοιχείου που διαγράφηκε.Έπειτα είναι οι peek(), printQueue(PrintStream), size(), isEmpty().Η πρώτη επιστρέφει το πρώτο στοιχείο χωρίς να γίνει διαγραφή αυτού,η δεύτερη τυπώνει τα στοιχεία της λίστας,η τρίτη επιστρέφει το μέγεθος της και η τελευταία ελέγχει αν η λίστα είναι άδεια επιστρέφοντας true ή false.Η υλοποίηση ολοκληρώνεται με την δημιουργία της main.Στη main δημιουργούμε ένα στιγμιότυπο της διεπαφής Γ με το οποίο χειριζόμαστε τη λίστα.Πραγματοποίουμε μερικές εισαγωγές στοιχείων και επείτα μερικές διαγραφές ελέγχοντας για exceptions.

ΥΛΟΠΟΙΗΣΗ ΔΙΕΠΑΦΗΣ Δ : Δημιουργούμε ένα αντικείμενο ουράς FIFO και ένα αντικείμενο ουράς με διπλά άκρα, από την υλοποίηση της Γ διεπαφής και της Α αντίστοιχα.Στην ουρά FIFO θα αναπαριστάται το περιεχόμενο της δομής όπως και στο ερώτημα Γ.Δηλαδή θα γίνεται προσθήκη χαρακτήρων(από την ουρά της δομής) και αφαίρεση χαρακτήρων(από την κεφαλή της δομής),χωρίς τυχόν επιπλέον λειτουργίες.Αντίθετα στην ουρά με τα διπλά άκρα για την εισαγωγή στοιχείου(εκτός του πρώτου) θα γίνεται έλεγχος αν το insertItem είναι λεξικογραφικά μεγαλύτερο από τον χαρακτήρα που βρίσκεται στον πρώτο κόμβο.Αν είναι τότε θα γίνεται διαγραφή όλων των στοιχείων από τον τελευταίο κόμβο εως και τον κόμβο που περιέχει τον μικρότερο χαρακτήρα.Ύστερα θα πραγματοποιείται η εισαγωγή του νέου χαρακτήρα.Ενώ αν το insertItem είναι μικρότερο από το στοιχείο του πρώτου κόμβου θα γίνεται κανονικά η εισαγωγή του στην αρχή της λίστας.Με αυτόν τον τρόπο

καταφέρνουμε να κρατάμε τον λεξικογραφικά μικρότερο χαρακτήρα είτε στον πρώτο είτε στον τελευταίο κόμβο της λίστας.Έτσι όταν γίνεται κλήση της συναρτησης Min() σιγκρίνουμε τον πρώτο με το τελευταίο στοιχείο της ουράς με τα διπλά άκρα και το μικρότερο αυτών είναι και το μικρότερο στοιχείο της λίστας.Όταν ζητείται η διαγραφή του παλαιότερου στοιχείου της FIFO τότε θα διαγράφεται το στοιχείο που βρίσκεται στην αρχή της ουράς και θα ελέγχουμε αν το στοιχείο αυτό βρίσκεται στο τέλος της ουράς με τα διπλά άκρα.Αν ναι τότε θα γίνεται διαγραφή αυτού διαφορετικά θα συνεχιστεί η εκτέλεση του προγράμματος.

--- ΜΕΡΟΣ Β ---

Γνωρίζουμε από την εκφώνηση ότι για να είναι μια ακολουθία χαρακτήρων Watson-Crick complemented palindrome πρέπει πρώτα από όλα να είναι έγκυρη.Αυτό σημαίνει ότι οι χαρακτήρες από τους οποίους αποτελείται η ακολουθία πρέπει να είναι είτε A είτε C είτε G είτε T.Επομένως, αρχικά ζητάμε την εισαγωγή ενός string από τον χρήστη και στη συνέχεια ελέγχουμε αν είναι έγκυρο.Αν δεν είναι έγκυρο τότε το πρόγραμμα τερματίζει και τυπώνεται ενα σχετικό μήνυμα στην οθόνη.Σε περίπτωση που το string είναι έγκυρο τότε πρέπει να ελέγξουμε αν είναι Watson-Crick complemented palindrome(έγκυρο είναι και το κένο " " string).Παρατηρούμε ότι αν ένα string είναι έγκυρο για να είναι Watson-Crick complemented palindrome χρειάζεται να έχει και ζυγό μέγεθος.Αν δεν είναι ζυγού μεγέθους τότε δεν είναι Watson-Crick complemented palindrome.Σε αύτη την περίπτωση το πρόγραμμα τερματίζει και τυπώνεται ένα σχετικό μήνυμα στην οθόνη.Αν,λοιπόν, το string είναι έγκυρο και έχει ζυγό μέγεθος τότε τυπώνονται στην οθόνη τα συμπληρωματικά ζευγάρια των νουκλεοτιδίων του string.Τέλος,ελέγχουμε αν το πλήθος των συμπληρωματικών ζευγαριών ισούται με το μέγεθος του string διαιρούμενο με το 2 και αν ισχύει τυπώνεται ένα μήνυμα στην οθόνη που ενημερώνει το χρήστη ότι το string είναι Watson-Crick complemented palindrome.