Προβλήματα με χρήση στοίβας και ουράς

1. Για μια δεδομένη στοίβα 100 θέσεων με ακέραιους αριθμούς να κατασκευάσετε τα παρακάτω υποπρογράμματα:
	1. Συνάρτηση, που να δέχεται τη στοίβα και τον δείκτη top και να επιστρέφει μια λογική τιμή για το αν αυτή περιέχει ή όχι στοιχεία.
	2. Διαδικασία, η οποία δέχεται τη στοίβα, τον δείκτη top και μία ακέραια θετική τιμή. Αν η τιμή είναι μικρότερη από τον δείκτη, θα γίνεται απώθηση όλων των στοιχείων μέχρι το top να φτάσει σε αυτή την τιμή.
	3. Διαδικασία, η οποία θα δέχεται τη στοίβα, τον δείκτη top και μία ακέραια θετική τιμή Ν. Θα γίνεται επαναληπτικά ώθηση στοιχείων Ν φορές ή όσες φορές είναι δυνατό να γίνει. Αν η στοίβα είναι γεμάτη, θα εμφανίζεται σχετικό μήνυμα.
2. Να γράψετε πρόγραμμα, το οποίο θα ελέγχει αν μία λέξη είναι καρκινική, δηλαδή αν μπορεί να διαβαστεί το ίδιο και ανάποδα. Το πρόγραμμα θα λειτουργεί ως εξής:
	1. Θα διαβάζει μία λέξη με το πολύ δέκα γράμματα και θα τα τοποθετεί με σειρά στη στοίβα ΑΡΧΙΚΗ. Η διαδικασία θα τερματίζεται όταν η στοίβα γεμίσει ή όταν δοθεί για χαρακτήρας το κενό.
	2. Στη συνέχεια, με τη διαδικασία της απώθησης θα τοποθετεί όλους τους χαρακτήρες έναν προς έναν στον πίνακα ΑΝΑΠΟΔΑ.
	3. Τέλος, θα συγκρίνει τις δύο δομές και θα εμφανίζει μήνυμα για το αν το περιεχόμενό τους είναι ίδιο.
3. Να γράψετε πρόγραμμα που θα προσομοιώνει τη λειτουργία της καταγραφής ιστορικού σε ένα πρόγραμμα περιήγησης στο διαδίκτυο, ως εξής:
	1. Θα αρχικοποιεί μια άδεια στοίβα, η οποία θα μπορεί να περιέχει το πολύ 1000 διευθύνσεις.
	2. Θα διαβάζει επαναληπτικά επιλογές του χρήστη, οι οποίες μπορούν να είναι: Ν (νέα σελίδα), Π (πίσω), Χ (έξοδος).
	3. Όταν δοθεί Ν, θα δίνεται από το πληκτρολόγιο η σελίδα που θέλει να επισκεφτεί ο χρήστης και θα καλείται η διαδικασία ΝΕΑ\_ΣΕΛ, που περιγράφεται παρακάτω.
	4. Όταν δοθεί Π, θα καλείται η συνάρτηση ΠΙΣΩ και το αποτέλεσμά της θα εκχωρείται στον δείκτη top. Αν υπάρχει προηγούμενη σελίδα, αυτή θα εμφανίζεται. Αν όχι, θα εμφανίζεται μήνυμα πως δεν υπάρχει. Η λειτουργία της συνάρτησης περιγράφεται αναλυτικά παρακάτω.
	5. Όταν δοθεί Χ η επαναληπτική διαδικασία θα τερματίζεται και θα εμφανίζεται ο αριθμός ιστοσελίδων που επισκέφθηκε ο χρήστης.
	6. Η διαδικασία ΝΕΑ\_ΣΕΛ θα δέχεται τη στοίβα, τη θέση του δείκτη top και την τελευταία διεύθυνση που διαβάστηκε. Σε περίπτωση που μπορεί να γίνει ώθηση θα τοποθετεί τη νέα διεύθυνση στη σωστή θέση, διαφορετικά θα εμφανίζει μήνυμα αδυναμίας πραγματοποίησης της λειτουργίας.
	7. Η συνάρτηση ΠΙΣΩ θα δέχεται τη στοίβα και τη θέση του δείκτη top και σε περίπτωση που υπάρχει στοιχείο για απώθηση θα μειώνει τη θέση του δείκτη κατά ένα και θα επιστρέφει την τιμή του. Διαφορετικά δε θα πραγματοποιεί αλλαγή.
4. Να γράψετε πρόγραμμα το οποίο:
	1. Θα δημιουργεί μια στοίβα η οποία θα μπορεί να δεχτεί το πολύ 20 λέξεις.
	2. Θα εμφανίζει επαναληπτικά τις επιλογές: Ν (νέα λέξη), Α (αναίρεση), Ε (έξοδος).
	3. Αν επιλεγεί Ν, θα δίνεται από το πληκτρολόγιο μια λέξη και θα καλείται η διαδικασία ΩΘΗΣΗ, που περιγράφεται παρακάτω.
	4. Αν επιλεγεί Α, θα καλείται η διαδικασία ΑΝΑΙΡΕΣΗ, που περιγράφεται παρακάτω.
	5. Αν επιλεγεί Ε, θα εμφανίζεται η πρόταση λέξη προς λέξη, ξεκινώντας από την αρχή της στοίβας. Στη συνέχεια θα εμφανίζονται οι λέξεις που διαγράφηκαν μία προς μία, ξεκινώντας από αυτή που διαγράφηκε πρώτη, καθώς και ο αριθμός τους.
	6. Η διαδικασία ΩΘΗΣΗ θα δέχεται τη στοίβα και την τιμή του δείκτη top και θα τοποθετεί τη νέα λέξη, εφ’ όσον αυτό είναι εφικτό. Διαφορετικά θα εμφανίζει σχετικό μήνυμα.
	7. Η διαδικασία ΑΝΑΙΡΕΣΗ θα δέχεται τη στοίβα και τον δείκτη top και σε περίπτωση που υπάρχει λέξη για να διαγραφεί θα το κάνει, τοποθετώντας ένα κενό στη θέση της, αλλάζοντας κατάλληλα και τη θέση του δείκτη. Διαφορετικά θα εμφανίζει σχετικό μήνυμα.
5. Για μία δεδομένη ουρά 100 θέσεων με ακέραιους αριθμούς, να γράψετε τα ακόλουθα υποπρογράμματα:
	1. Συνάρτηση που να δέχεται τους δύο δείκτες της ουράς και να επιστρέφει λογική τιμή, ανάλογα με το αν αυτή περιέχει στοιχεία ή όχι.
	2. Συνάρτηση που να δέχεται τους δύο δείκτες και να επιστρέφει τον αριθμό στοιχείων που περιέχει η ουρά.
	3. Διαδικασία, η οποία θα δέχεται την ουρά, τους δύο δείκτες της και μία θετική ακέραια τιμή Ν. Σε περίπτωση που υπάρχουν Ν κενές θέσεις, θα πραγματοποιείται εισαγωγή ακέραιων αριθμών από το πληκτρολόγιο ισάριθμες φορές. Αν οι θέσεις είναι λιγότερες, θα πραγματοποιείται η ίδια διαδικασία μόνο για τις διαθέσιμες θέσεις. Σε περίπτωση που η ουρά είναι γεμάτη, θα εμφανίζεται σχετικό μήνυμα.
	4. Διαδικασία, η οποία θα δέχεται την ουρά και τους δύο δείκτες της και, εφ’ όσον διαπιστώσει ότι αυτή περιέχει στοιχεία και υπάρχουν κενές θέσεις στην αρχή, θα τα μεταφέρει στις πρώτες θέσεις.
	5. Διαδικασία η οποία θα δέχεται την ουρά και τους δύο δείκτες, θα διαβάζει έναν ακέραιο αριθμό και θα κάνει εισαγωγή του αν αυτό είναι δυνατό. Σε κάθε περίπτωση θα επιστρέφει στο κύριο πρόγραμμα μια λογική μεταβλητή με το αν πραγματοποιήθηκε ή όχι η εισαγωγή.
6. Σε ένα αεροδρόμιο, μια αεροπορική εταιρεία έχει ανοίξει τρία γκισέ για τη διαδικασία επιβίβασης (check in) των επιβατών στην επόμενη πτήση. Κάθε επιβάτης πάει εκεί για να παραδώσει τις αποσκευές του και εξυπηρετείται πρώτος αυτός που φτάνει πρώτος, από όποιο γκισέ είναι διαθέσιμο. Για την επόμενη πτήση έχουν πουληθεί συνολικά 150 εισιτήρια. Να γράψετε πρόγραμμα, το οποίο:
	1. Αρχικοποιεί μια ουρά 150 θέσεων.
	2. Διαβάζει επαναληπτικά τις επιλογές που δίνονται από το πληκτρολόγιο: C (check in), Ν (νέος στην ουρά), Τ (τέλος).
	3. Αν δοθεί η επιλογή Ν, διαβάζεται το όνομα ενός επιβάτη και τοποθετείται στην τελευταία θέση της ουράς. Αν η ουρά είναι γεμάτη, τότε εμφανίζεται σχετικό μήνυμα.
	4. Αν δοθεί η επιλογή C, ο πρώτος επιβάτης που περιμένει στην ουρά εξυπηρετείται από ένα γκισέ. Θα διαβάζεται ο αριθμός του γκισέ (1,2 ή 3) και στη συνέχεια θα εμφανίζεται το όνομα του πελάτη με μήνυμα «μπορείτε να επιβιβαστείτε στο αεροπλάνο». Σε περίπτωση που δεν υπάρχει άλλος επιβάτης στην ουρά θα εμφανίζεται μήνυμα λάθους.
	5. Μετά από κάθε τέτοια επιλογή θα εμφανίζεται μήνυμα για το πόσοι επιβάτες περιμένουν αυτή τη στιγμή στην ουρά.
	6. Στο τέλος της διαδικασίας, το οποίο θα πραγματοποιείται με την επιλογή Τ, θα εμφανίζεται μήνυμα για το πόσους επιβάτες εξυπηρέτησε το κάθε γκισέ. Επίσης, αν υπήρξαν κάποιοι που έχασαν την πτήση και πόσοι ήταν αυτοί. Ο αριθμός αυτός προκύπτει από το αν εμφανίστηκαν όλοι όσοι είχαν αγοράσει εισιτήριο.
7. Σε ένα οινοποιείο λειτουργεί αυτόματο σύστημα για την εμφιάλωση των μπουκαλιών. Το μηχάνημα τροφοδοτείται από έναν χειριστή ηλεκτρονικά και μπορεί να εξυπηρετήσει μέχρι 200 μπουκάλια ταυτόχρονα, πραγματοποιώντας όλα τα στάδια (πλύσιμο, γέμισμα, τοποθέτηση ετικέτας, σφράγισμα μπουκαλιού). Στην άλλη άκρη της γραμμής παραγωγής, ένας άλλος υπάλληλος παραλαμβάνει τα μπουκάλια που είναι έτοιμα και τα τοποθετεί σε κιβώτια. Κάθε μπουκάλι εξυπηρετείται με τη διαδικασία πρώτο μέσα – πρώτο έξω. Να γράψετε πρόγραμμα, το οποίο:
	1. Θα αρχικοποιεί την ουρά των 200 θέσεων.
	2. Θα διαβάζει επαναληπτικά μια επιλογή από τις παρακάτω: 1. Εισαγωγή 2. Εξαγωγή 3. Τέλος εμφιάλωσης.
	3. Με την επιλογή 1, που δίνει ο χειριστής στην είσοδο του συστήματος, θα καλείται η διαδικασία ΕΙΣΑΓΩΓΗ, που περιγράφεται παρακάτω.
	4. Με την επιλογή 2 θα καλείται η διαδικασία ΕΞΑΓΩΓΗ, που περιγράφεται επίσης παρακάτω.
	5. Με την επιλογή 3, η εμφιάλωση θα τερματίζεται και σε περίπτωση που έχουν απομείνει μπουκάλια μέσα στο μηχάνημα θα καλείται η διαδικασία εξαγωγή όσες φορές χρειάζεται ώστε να εμφιαλωθούν και αυτά.
	6. Στο τέλος θα εμφανίζεται ο αριθμός ολόκληρων κιβωτίων που εμφιαλώθηκαν, δεδομένου ότι κάθε κιβώτιο χωράει 6 μπουκάλια.
	7. Η διαδικασία ΕΙΣΑΓΩΓΗ θα δέχεται την ουρά και τους δύο δείκτες της και, εφ’ όσον αυτό είναι δυνατό, θα τοποθετεί σε αυτήν έναν μπουκάλι τη φορά, αφού πρώτα διαβάσει το είδος του. Σε περίπτωση που η ουρά είναι γεμάτη, θα εμφανίζεται μήνυμα αναμονής.
	8. Η διαδικασία ΕΞΑΓΩΓΗ θα δέχεται την ουρά και τους δύο δείκτες και θα εμφανίζει μήνυμα για το είδος κρασιού που περιείχε το μπουκάλι που εξάγεται. Στη συνέχεια θα τροποποιεί κατάλληλα τους δύο δείκτες. Αν δεν υπάρχει μπουκάλι προς εξαγωγή, θα εμφανίζεται μήνυμα «Ουρά εμφιάλωσης άδεια».

1. Μια ακτοπλοϊκή εταιρεία έχει εκδοτήρια στο λιμάνι του Πειραιά. Ο κόσμος περιμένει στην ουρά και κάθε φορά εξυπηρετείται ο πελάτης που περιμένει πρώτος στην ουρά. Κάθε νεοεισερχόμενος περιμένει πίσω από τον τελευταίο. Τα διαθέσιμα εισιτήρια για το επόμενο δρομολόγιο πλοίου είναι 800 και κάθε επιβάτης μπορεί να αγοράσει ένα ή περισσότερα. Τα εισιτήρια κοστίζουν 45€ το κανονικό και 30€ το μειωμένο. Να γράψετε εφαρμογή για την παραπάνω διαδικασία, η οποία:
	1. Θα αρχικοποιεί μια άδεια ουρά που να μπορεί να εξυπηρετήσει ακόμα κι αν ο κάθε πελάτης αγοράσει εισιτήριο μόνο για τον εαυτό του, καθώς και μια άδεια στοίβα, στην οποία θα καταχωρηθούν τα ονόματα των επιβατών που θα μπουν στο πλοίο.
	2. Θα διαβάζει επαναληπτικά τρεις επιλογές: Α (αγορά εισιτηρίων), Ν (νέο άτομο στην ουρά), Τ (τέλος πώλησης).
	3. Για την επιλογή Ν, θα διαβάζεται το όνομα ενός πελάτη και, αν υπάρχει χώρος στην ουρά, αυτό θα τοποθετείται στην κατάλληλη θέση. Διαφορετικά θα εμφανίζεται μήνυμα πως θα πρέπει να επιβιβαστεί σε άλλο δρομολόγιο.
	4. Για την επιλογή Α, θα δίνεται από το πληκτρολόγιο ο αριθμός των κανονικών και μειωμένων εισιτηρίων που επιθυμεί ο πελάτης. Αν οι διαθέσιμες θέσεις αρκούν, θα διαβάζονται ένα προς ένα τα ονόματα των επιβατών που αφορούν τα εισιτήρια και θα εισάγονται στην στοίβα με σειρά. Στη συνέχεια, θα εμφανίζεται το όνομα του επιβάτη με μήνυμα για το ποσό που πρέπει να πληρώσει και το όνομά του θα εξάγεται από την ουρά. Αν οι διαθέσιμες θέσεις δεν επαρκούν, θα πωλούνται με τη διαδικασία που περιγράφηκε παραπάνω μόνο τα εισιτήρια που είναι διαθέσιμα.
	5. Η επαναληπτική διαδικασία θα τερματίζεται αν επιλεγεί Τ ή αν εξαντληθούν τα εισιτήρια.
	6. Μετά το τέλος της διαδικασίας θα εμφανίζεται με σειρά η λίστα των επιβατών, ξεκινώντας από το τελευταίο όνομα που δόθηκε.
	7. Θα εμφανίζεται επίσης μήνυμα με τον αριθμό εισιτηρίων που πουλήθηκε, το ποσοστό πληρότητας του δρομολογίου και τα έσοδα που είχε από τα εισιτήρια των επιβατών.