

Ενότητα 1. Δομές Δεδομένων και Αλγόριθμοι

1.1 Στοίβα

Δύο τριτοετείς φοιτητές αποφάσισαν τον Δεκαπενταύγουστο να πάρουν το αυτοκίνητό τους και να περάσουν ένα τριήμερο στην Αίγινα. Θεωρούσαν ότι το οχηματαγωγό πλοίο χωρητικότητας 30 αυτοκινήτων, που εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ ήταν αρκετά μεγάλο και δεν ήταν απαραίτητη η εκ των προτέρων κράτηση θέσης αυτοκινήτου. Όταν όμως έφθασαν στο λιμάνι, πληροφορήθηκαν, ότι το πλοίο ήταν πλήρες από οχήματα και μόνο σε περίπτωση ακύρωσης θα ελευθερώνονταν θέση για το αυτοκίνητό τους. Αυτό όμως θα το μάθαιναν λίγο πριν τον απόπλου. Τελικά, για καλή τους τύχη, έγινε μία ακύρωση και επιβιβάστηκαν τελευταίοι στο πλοίο μαζί με το αυτοκίνητό τους. Τότε, συνειδητοποίησαν ότι, το πλοίο είχε μία πόρτα, την ίδια για είσοδο και έξοδο των οχημάτων και θα αποβιβάζονταν πρώτοι αποφεύγοντας την ταλαιπωρία της αναμονής.

Παρατήρησαν, μάλιστα, ότι τα οχήματα που μπήκαν πρώτα ήταν σταθμευμένα στο βάθος του πλοίου **σε μία σειρά** το ένα μετά το άλλο και τα τελευταία συμπλήρωναν **αυτή τη σειρά** μέχρι και την πόρτα και μέχρι το πλήθος των 30 αυτοκινήτων.

Δηλαδή, η κατασκευή του «γκαραζ» του πλοίου και ο τρόπος στάθμευσης των οχημάτων προσομοιάζε στη δομή της «Στοίβας».



Στοίβα (stack), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.

Η παραπάνω μέθοδος ονομάζεται **Τελευταίο Μέσα, Πρώτο Έξω ή LIFO** (=Last In First Out).

Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας στοίβας σε κατακόρυφη σειρά. Χαρακτηριστικό παράδειγμα είναι μια στοίβα από πιάτα. Παίρνουμε προς χρήση το πιάτο που τοποθετήσαμε τελευταίο.



Οι **κύριες λειτουργίες** σε μια στοίβα είναι δύο:

1. Η **ώθηση** (push) στοιχείου στην κορυφή της στοίβας.
Στη διαδικασία της ώθησης ελέγχουμε αν η στοίβα είναι γεμάτη.
Στην περίπτωση που προσπαθήσουμε να «προσθέσουμε» ένα στοιχείο σε μια ήδη γεμάτη στοίβα, έχουμε **υπερχείλιση** (overflow) της στοίβας.
2. Η **απώθηση** (pop) στοιχείου από τη στοίβα.
Στη διαδικασία της απώθησης ελέγχουμε αν υπάρχει ένα τουλάχιστον στοιχείο στη στοίβα.
Στην περίπτωση που προσπαθήσουμε να «αφαιρέσουμε» ένα στοιχείο από μία κενή στοίβα, έχουμε **υποχείλιση** (underflow) της στοίβας.

Υλοποίηση στοίβας με χρήση μονοδιάστατου πίνακα

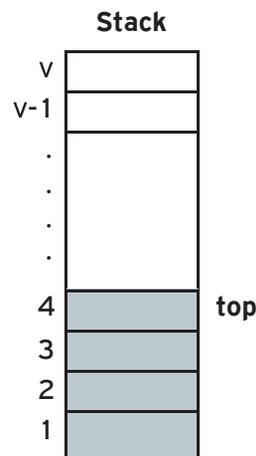
- Χρησιμοποιούμε μια βοηθητική μεταβλητή (top), που δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοίβας.
- Η **ώθηση** ενός νέου στοιχείου στη στοίβα (εισαγωγή στοιχείου στον πίνακα) γίνεται πάντα στην κορυφή της. Συγκεκριμένα, η μεταβλητή top αυξάνεται κατά ένα:

$$\text{top} \leftarrow \text{top} + 1$$

και στη συνέχεια γίνεται η ώθηση του στοιχείου.

- Η **απώθηση** ενός στοιχείου από τη στοίβα (εξαγωγή από τον πίνακα) γίνεται πάντα από την κορυφή της στοίβας. Συγκεκριμένα, εξάγεται το στοιχείο που δείχνει η μεταβλητή top και στη συνέχεια η μεταβλητή top μειώνεται κατά ένα:

$$\text{top} \leftarrow \text{top} - 1$$



Εικόνα 1. 1. Υλοποίηση Στοίβας



Σε μια κενή στοίβα/πίνακα θεωρούμε ότι η αρχική τιμή της μεταβλητής top είναι μηδέν ($\text{top} \leftarrow 0$). Η μεταβλητή top είναι η μεταβλητή που δείχνει τη θέση που τοποθετήθηκε το τελευταίο στοιχείο στη στοίβα/πίνακα (δηλ. δείχνει την κορυφή της στοίβας).

Κατά την **ώθηση** ενός στοιχείου στη στοίβα (εισαγωγή ενός στοιχείου στον πίνακα), πρώτα αυξάνεται η τιμή της μεταβλητής top κατά ένα, δηλ. $\text{top} \leftarrow \text{top} + 1$, και στη συνέχεια γίνεται η ώθηση του στοιχείου στην κορυφή της στοίβας.

Κατά την **απώθηση** ενός στοιχείου από τη στοίβα (εξαγωγή στοιχείου από τον πίνακα) μειώνεται η τιμή της μεταβλητής top κατά ένα, δηλ. $\text{top} \leftarrow \text{top} - 1$. Στην απώθηση δε διαγράφεται το στοιχείο, στην πραγματικότητα δε γίνεται καμία παρέμβαση στα περιεχόμενα του πίνακα. Απλώς ο δείκτης κορυφή δείχνει στην προηγούμενη θέση.

1.1.1 Παραδείγματα υλοποίησης στοίβας με χρήση μονοδιάστατου πίνακα

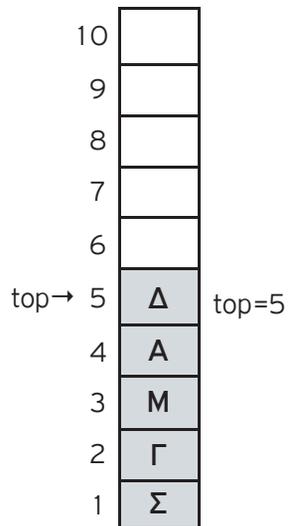


Παράδειγμα 1 – Ώθηση & Απώθηση στοιχείου σε στοίβα

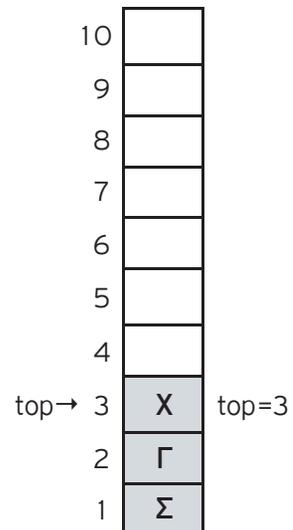
- 1) Σε μια στοίβα 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
 - i. Να προσδιορίσετε την τιμή του δείκτη top και να σχεδιάσετε την παραπάνω στοίβα.
 - ii. Αν εφαρμόσετε τις παρακάτω λειτουργίες: **Απώθηση, Απώθηση, Απώθηση, Ώθηση X, Ώθηση Δ** και **Απώθηση**, ποια είναι η νέα τιμή της top και ποια η τελική μορφή της στοίβας;
- 2) Σε μια άδεια στοίβα 10 θέσεων ωθούνται τα στοιχεία Ο, Σ, Λ, Τ, Ε. Με ποιον τρόπο πρέπει να γίνει η ώθηση και η απώθηση των στοιχείων, ώστε να έχουμε ως έξοδο τα στοιχεία Τ, Ε, Λ, Ο, Σ, με το στοιχείο Σ να βρίσκεται στην κορυφή της στοίβας;

Απάντηση

1) i)

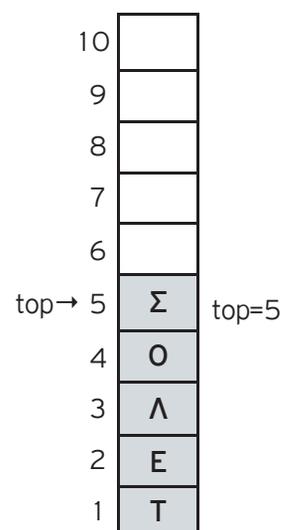
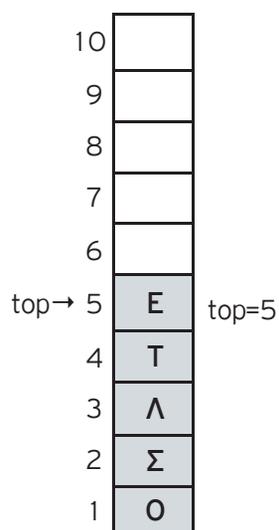


ii) Η νέα τιμή της top είναι 3 και η στοίβα γίνεται:



2) Η αρχική μορφή της στοίβας είναι:

Εκτελώντας τις λειτουργίες : **Απώθηση, Απώθηση, Απώθηση, Απώθηση, Απώθηση, Ύθνηση Τ, Ύθνηση Ε, Ύθνηση Λ, Ύθνηση Ο, Ύθνηση Σ**, τότε η τελική μορφή της στοίβας γίνεται:





Παράδειγμα 2 – Ώθηση στοιχείου στην κορυφή της στοίβας με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ, που πραγματοποιεί ώθηση στοιχείου στην κορυφή της στοίβας με χρήση μονοδιάστατου πίνακα A, 10 θέσεων.

Απάντηση

Θεωρούμε ότι η στοίβα δεν είναι κενή και η μεταβλητή top έχει μια έγκυρη τιμή.



Κώδικας σε ΓΛΩΣΣΑ [1.1]

```

1  ΓΡΑΨΕ 'Δώσε στοιχείο για να εισαχθεί στη στοίβα A:'
2  ΔΙΑΒΑΣΕ στοιχείο
3  ΑΝ top<10 ΤΟΤΕ
4      top<- top+1
5      A[top]<- στοιχείο
6  ΑΛΛΙΩΣ
7      ΓΡΑΨΕ 'Υπερχείλιση στοίβας'
8  ΤΕΛΟΣ_ΑΝ

```



Παράδειγμα 3 – Απώθηση στοιχείου από στοίβα με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που πραγματοποιεί την απώθηση στοιχείου από στοίβα με χρήση ενός μονοδιάστατου πίνακα A, 10 θέσεων

Απάντηση

Θεωρούμε ότι η στοίβα δεν είναι κενή και η μεταβλητή top έχει μια έγκυρη τιμή.



Κώδικας σε ΓΛΩΣΣΑ [1.2]

```

1  ΑΝ top>=1 ΤΟΤΕ
2      ΓΡΑΨΕ A[top]
3      top<- top-1
4  ΑΛΛΙΩΣ
5      ΓΡΑΨΕ 'Υποχείλιση στοίβας'
6  ΤΕΛΟΣ_ΑΝ

```



Παράδειγμα 4 – Επιβίβαση & Αποβίβαση αυτοκινήτων σε πλοίο

Ένα οχηματαγωγό πλοίο, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Στο λιμάνι του Πειραιά προσέρχονται τα οχήματα για αναχώρηση. Τα οχήματα που επιβιβάζονται πρώτα είναι αυτά που θα αποβιβαστούν τελευταία.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:
 1. Επιβίβαση 2. Αποβίβαση 3. Έξοδος
2. Στην περίπτωση που επιλεγεί η **Επιβίβαση**, να ζητείται εισαγωγή του αριθμού κυκλοφορίας καθενός από τα οχήματα που προσέρχονται και ο αριθμός κυκλοφορίας του να καταχωρείται στη στοίβα ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «**Υπάρχει όχημα για επιβίβαση; (N/O)**». Αν ο χρήστης απαντήσει N (=ΝΑΙ), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει O (=ΟΧΙ), τότε να σταματά η διαδικασία επιβίβασης και το πρόγραμμα να επιστρέφει στο μενού Επιλογής.
3. Στην περίπτωση που επιλεγεί η **Αποβίβαση**, να τυπώνει τον αριθμό κυκλοφορίας όλων των οχημάτων με τη σειρά που αποβιβάζονται από το πλοίο στην ΑΙΓΙΝΑ.
4. Στο τέλος να τυπώνει το πλήθος των οχημάτων που επιβιβάστηκαν στο λιμάνι του ΠΕΙΡΑΙΑ.

Απάντηση



Παρακάτω υλοποιείται το πρόγραμμα του παραδείγματος επιβίβασης/αποβίβασης σε οχηματαγωγό πλοίο, όπου η «πόρτα» για την είσοδο και την έξοδο των οχημάτων στο πλοίο, είναι ίδια.

Ανάλυση της λύσης

Σύμφωνα με την εκφώνηση της άσκησης πρέπει:

- ο Να υλοποιηθεί μια δομή δεδομένων LIFO (στοίβα). Η υλοποίηση της στοίβας θα πραγματοποιηθεί με μονοδιάστατο πίνακα, όπου οι θέσεις του πίνακα θα είναι τόσες όσες και το μέγιστο πλήθος επιβιβαζόμενων οχημάτων (250). Επομένως για την υλοποίηση της στοίβας θα χρησιμοποιηθεί ένας μονοδιάστατος πίνακας **n** 250 θέσεων, π[250].
- ο Θα πρέπει να δηλωθεί η βοηθητική μεταβλητή (δείκτης) «**τοπ**», που δείχνει το στοιχείο που έχει τοποθετηθεί τελευταίο στη στοίβα. Η αρχική τιμή της μεταβλητής «**τοπ**» πρέπει να είναι 0 για να δείχνει στη «θέση» 0 του πίνακα, επισημαίνοντας κατά αυτόν τον τρόπο ότι το πλοίο/στοίβα είναι κενό. Η μεταβλητή «**τοπ**» αυξάνεται κάθε φορά κατά 1 ($\text{τοπ} \leftarrow \text{τοπ} + 1$), και δείχνει την κενή θέση, την οποία καταλαμβάνει κάθε νέο επιβιβαζόμενο όχημα στο πλοίο.
- ο Να δηλωθούν οι μεταβλητές: **επ1** (επιλογή μενού), **πλ1** (μετρητής επιβιβαζόμενων οχημάτων, με αρχική τιμή 0), **πλ2** (μετρητής αποβιβαζόμενων οχημάτων, με αρχική τιμή 0), **επ2** (μεταβλητή δήλωσης οχήματος για επιβίβαση), **αρ** (αριθμός κυκλοφορίας οχήματος, μέσω του οποίου καταχωρίζεται-δηλώνεται η επιβίβασή του), π[250].

- ο Να εμφανίζεται ένα μενού με τις επιλογές: **1.** «Επιβίβαση», **2.** «Αποβίβαση», **3.** «Έξοδος» (χρήση δομής επανάληψης «**ΜΕΧΡΙΣ_ΟΤΟΥ...**», μεταβλητή επιλογών «**επ1**»)
 - ο Στην περίπτωση της **επιβίβασης** («**ΑΝ επ1 = 1 ΤΟΤΕ**»), θα πρέπει το πρόγραμμα να ελέγχει αν υπάρχει όχημα για επιβίβαση (μεταβλητή «**επ2**»). Αν υπάρχει όχημα για επιβίβαση, θα πρέπει παράλληλα να γίνεται έλεγχος αν υπάρχει κενή θέση στο πλοίο («**τοπ < 250**»). Αν υπάρχει κενή θέση, αυξάνεται κατά ένα η τιμή της μεταβλητής «**τοπ**» («**τοπ<-τοπ+1**») και στη συνέχεια γίνεται καταχώριση του αριθμού κυκλοφορίας του οχήματος σε αυτή την κενή θέση της στοιβας/πίνακα («**π[τοπ]<-αρ**»). Αμέσως μετά την τυχόν καταχώριση του αυτοκινήτου, η τιμή του μετρητή καταχώρισης επιβιβαζόμενων οχημάτων αυξάνεται κατά 1 («**πλ1<-πλ1 + 1**»). Σε αντίθετη περίπτωση (γεμάτο πλοίο/υπερχείλιση στοιβας), να μη γίνεται καμιά καταχώριση και να εμφανίζεται το μήνυμα («**ΓΡΑΨΕ 'Το πλοίο γέμισε και δε χωρά άλλα αυτοκίνητα'**»).
 - ο Στην περίπτωση της **αποβίβασης** («**ΑΛΛΙΩΣ_ΑΝ επ1=2 ΤΟΤΕ**») θα πρέπει να ελέγχεται αν υπάρχει αυτοκίνητο για αποβίβαση, μέσω της μεταβλητής «**τοπ**» («**ΟΣΟ τοπ>=1 ΕΠΑΝΑΛΑΒΕ**»). Αν υπάρχει όχημα προς αποβίβαση, τυπώνεται ο αριθμός κυκλοφορίας του αποβιβαζόμενου οχήματος με κατάλληλο μήνυμα («**ΓΡΑΨΕ 'Αποβιβάζεται το όχημα: ', π[τοπ]**»). Κατόπιν μειώνεται η τιμή του δείκτη «**τοπ**» κατά 1 («**τοπ<-τοπ-1**»), για να δείχνει στην προηγούμενη θέση της στοιβας/πίνακα, θέση που περιέχει τον αριθμό κυκλοφορίας του επόμενου οχήματος προς αποβίβαση. Παράλληλα, αυξάνουμε την τιμή του μετρητή καταχώρισης αποβιβαζόμενων οχημάτων («**πλ2 <- πλ2 + 1**»). Σε αντίθετη περίπτωση, αν ο δείκτης «**τοπ**» από την αρχή δείχνει 0 (υποχείλιση, δεν υπάρχουν αυτοκίνητα για αποβίβαση ή έχουν αποβιβασθεί όλα τα αυτοκίνητα), θα εμφανίζεται κατάλληλο μήνυμα με το σύνολο των αποβιβασθέντων οχημάτων («**ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: ', πλ2**»).
- ο Να σημειωθεί ότι αν δεν αποβιβασθούν οχήματα ή για κάθε νέα επιλογή επιβίβασης ή αποβίβασης (νέα εκτέλεση του μενού), η μεταβλητή πλ2 έχει την αρχική της τιμή, που είναι 0.
- ο Τέλος, με την εκτύπωση του πλήθους επιβιβασθέντων οχημάτων στον Πειραιά («**ΓΡΑΨΕ 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ: ', πλ1**»), γίνεται και τυπικός έλεγχος μεταξύ των οχημάτων που έχουν επιβιβαστεί στο πλοίο κι έχουν αποβιβαστεί από αυτό.

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.



Κώδικας σε ΓΛΩΣΣΑ [1.3]

ΠΡΟΓΡΑΜΜΑ πλοίο

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ1, πλ2

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]

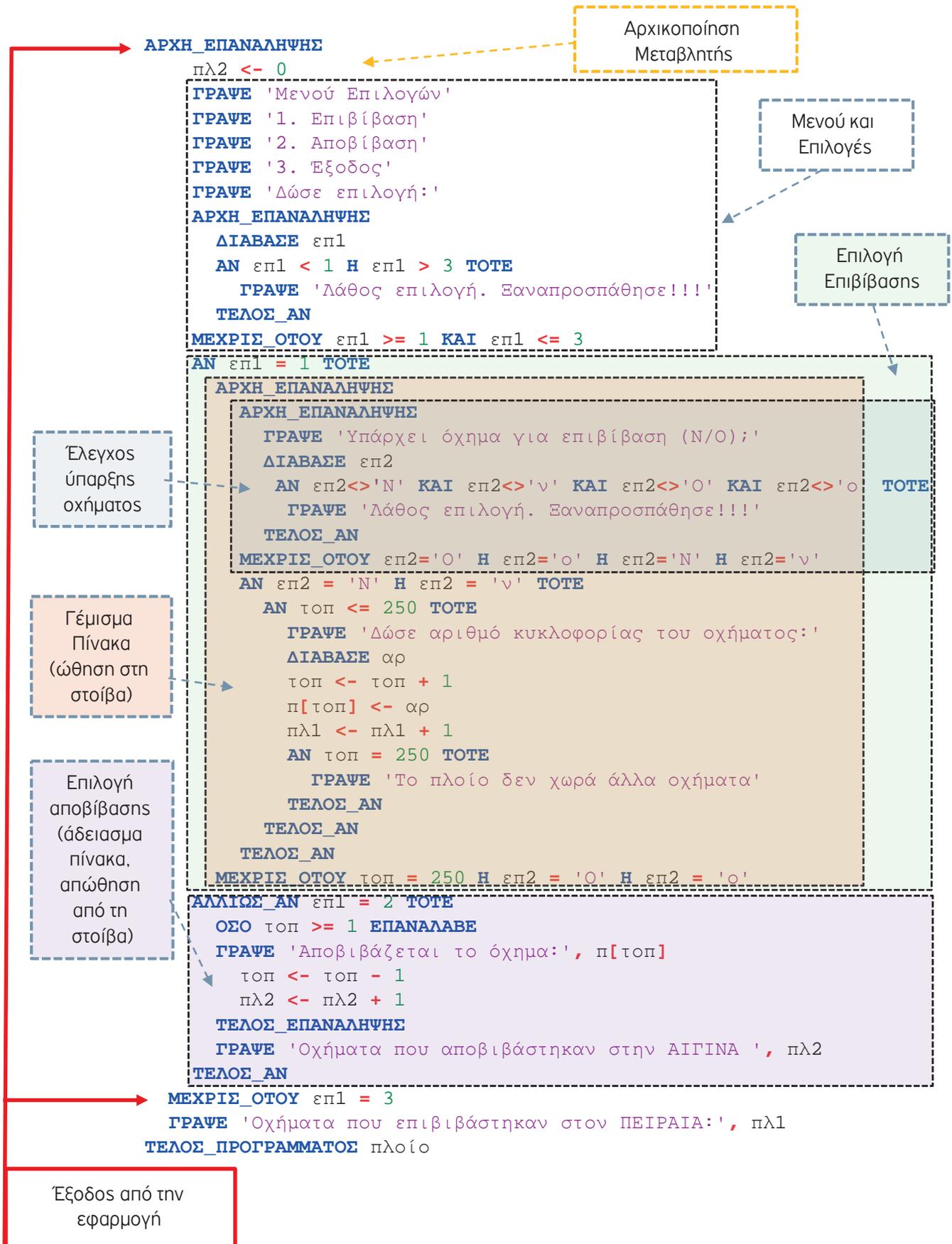
ΑΡΧΗ

τοπ <- 0

πλ1 <- 0

Δήλωση Μεταβλητών

Αρχικοποίηση Μεταβλητών



```

1  ΠΡΟΓΡΑΜΜΑ πλοίο
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ1, πλ2
4      ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
5  ΑΡΧΗ
6      τοπ <- 0
7      πλ1 <- 0
8  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
9      πλ2 <- 0
10     ΓΡΑΨΕ 'Μενού Επιλογών'
11     ΓΡΑΨΕ '1. Επιβίβαση'
12     ΓΡΑΨΕ '2. Αποβίβαση'
13     ΓΡΑΨΕ '3. Έξοδος'
14     ΓΡΑΨΕ 'Δώσε επιλογή:'
15     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
16         ΔΙΑΒΑΣΕ επ1
17         ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
18             ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
19             ΤΕΛΟΣ_ΑΝ
20     ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
21     ΑΝ επ1 = 1 ΤΟΤΕ
22         ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
23             ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
24                 ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (N/O);'
25                 ΔΙΑΒΑΣΕ επ2
26                 ΑΝ επ2<>'N' ΚΑΙ επ2<>'n' ΚΑΙ επ2<>'O' ΚΑΙ επ2<>'o' ΤΟΤΕ
27                     ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
28                     ΤΕΛΟΣ_ΑΝ
29                 ΜΕΧΡΙΣ_ΟΤΟΥ επ2='O' Η επ2='o' Η επ2='N' Η επ2='n'
30                 ΑΝ επ2 = 'N' Η επ2 = 'n' ΤΟΤΕ
31                     ΑΝ τοπ <= 250 ΤΟΤΕ
32                         ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
33                         ΔΙΑΒΑΣΕ αρ
34                         τοπ <- τοπ + 1
35                         π[τοπ] <- αρ
36                         πλ1 <- πλ1 + 1
37                     ΑΝ τοπ = 250 ΤΟΤΕ
38                         ΓΡΑΨΕ 'Το πλοίο δεν χωρά άλλα οχήματα'
39                     ΤΕΛΟΣ_ΑΝ
40                 ΤΕΛΟΣ_ΑΝ

```

```

41         ΤΕΛΟΣ_ΑΝ
42     ΜΕΧΡΙΣ_ΟΤΟΥ τοπ = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
43     ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
44         ΟΣΟ τοπ >= 1 ΕΠΑΝΑΛΑΒΕ
45             ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[τοπ]
46             τοπ <- τοπ - 1
47             πλ2 <- πλ2 + 1
48         ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
49         ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ ', πλ2
50     ΤΕΛΟΣ_ΑΝ
51     ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
52     ΓΡΑΨΕ 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ:', πλ1
53 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο

```

1.1.2 Ερωτήσεις - Ασκήσεις

Ε. 1: Δίνεται η επόμενη ακολουθία αριθμών: 4, 8, 2, 5, 9, 13.

- Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών σε στοίβα;
- Σχεδιάστε τη στοίβα μετά την τοποθέτηση των αριθμών.
- Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο των αριθμών από τη στοίβα;
- Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στη στοίβα για να εξαχθεί ο αριθμός 5;

Ε. 2: Σε μια στοίβα έχουν τοποθετηθεί κατά σειρά οι αριθμοί : 24, 7, 11, 13, 65, 39, 5.

- Να σχεδιάσετε την παραπάνω δομή.
- Ποια θα είναι η τιμή του δείκτη της παραπάνω στοίβας;
- Αν θέλετε να τοποθετήσετε τον αριθμό 25 στην στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;
- Ποια θα είναι η τιμή του δείκτη μετά την λειτουργία αυτή;
- Αν θέλετε να εξαγάγετε τον αριθμό 65 από τη στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;
- Ποια θα είναι η τιμή του δείκτη μετά τη λειτουργία αυτή;

Ε. 3: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση μιας στοίβας μπορεί να χρησιμοποιηθεί ένας πίνακας.	<input type="checkbox"/>	<input type="checkbox"/>
2	Στη στοίβα το στοιχείο που μπαίνει πρώτο βγαίνει πρώτο.	<input type="checkbox"/>	<input type="checkbox"/>
3	Στην υλοποίηση της στοίβας χρειάζονται δύο μεταβλητές-δείκτες για την υλοποίηση των δύο βασικών λειτουργιών που εκτελούνται σε αυτή.	<input type="checkbox"/>	<input type="checkbox"/>
4	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια άδεια στοίβα.	<input type="checkbox"/>	<input type="checkbox"/>
5	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια γεμάτη στοίβα.	<input type="checkbox"/>	<input type="checkbox"/>
6	Η ώθηση στοιχείου γίνεται στην κορυφή της στοίβας.	<input type="checkbox"/>	<input type="checkbox"/>
7	Στη δομή της στοίβας απαιτούνται δύο δείκτες, ο εμπρός και ο πίσω.	<input type="checkbox"/>	<input type="checkbox"/>
8	Υπερχείλιση έχουμε όταν εισάγουμε ένα στοιχείο σε μια ήδη γεμάτη στοίβα.	<input type="checkbox"/>	<input type="checkbox"/>
9	Η μέθοδος LIFO περιγράφει τη διαδικασία εκείνη κατά την οποία το στοιχείο που τοποθετείται τελευταίο εξάγεται πρώτο	<input type="checkbox"/>	<input type="checkbox"/>
10	Κάθε στοιχείο που εισάγεται πρώτο σε μια στοίβα είναι αυτό που εξάγεται πρώτο.	<input type="checkbox"/>	<input type="checkbox"/>

1.2 Ουρά

Κατά την ημέρα αναχώρησης τους από την Αίγινα, οι δύο συμφοιτητές αποφάσισαν σκόπιμα να καθυστερήσουν, προκειμένου να μπουν τελευταίοι (ώστε να βγουν πρώτοι και χωρίς ταλαιπωρία στον Πειραιά) και ταυτόχρονα, να απολαύσουν για περισσότερο χρόνο τις τελευταίες στιγμές των διακοπών τους στο νησί.

Δυστυχώς, όμως, ατύχησαν! Το πλοίο της επιστροφής τους ήταν διαφορετικό. Είχε μία είσοδο στο πίσω μέρος προς επιβίβαση των οχημάτων και μία έξοδο στο μπροστινό μέρος προς αποβίβαση. Έτσι, το αυτοκίνητο που έμπαινε πρώτο, προχωρούσε ευθεία προς την άλλη μεριά του πλοίου και έπαιρνε θέση προς αποβίβαση προς την πόρτα εξόδου. Κάθε αυτοκίνητο που έμπαινε στο πλοίο πάκκαρε υποχρεωτικά πίσω από τα άλλα, που είχαν μπει νωρίτερα, στην ίδια σειρά. Δηλαδή, το αυτοκίνητο που επιβιβαζόταν πρώτο στην Αίγινα θα αποβιβαζόταν πρώτο και στον Πειραιά. Με τον τρόπο αυτό, η επιβίβαση και η αποβίβαση των οχημάτων γινόταν πιο γρήγορα. Η κατασκευή του «γκαράζ» του πλοίου και ο τρόπος στάθμευσης των οχημάτων (τρόπος οργάνωσης των δεδομένων) προσομοιάζε στη δομή της «Ουράς». Δηλαδή, τα δεδομένα που μπαίνουν πρώτα σε μια ουρά είναι αυτά που βγαίνουν και πρώτα.



Ουρά (Queue), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.

Η παραπάνω μέθοδος ονομάζεται **Πρώτο Μέσα, Πρώτο Έξω ή FIFO** (=First In First Out).

Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας ουράς σε οριζόντια σειρά. Ο πελάτης μιας τράπεζας που μπαίνει πρώτος σε μια ουρά για να εξυπηρετηθεί, είναι αυτός που εξυπηρετείται και πρώτος, με την έξοδό του από την ουρά αναμονής.

Άλλα παραδείγματα είναι η ουρά στα λεωφορεία ή η ουρά στα ταμεία, όπου ο πρώτος που στέκεται στην ουρά εξυπηρετείται και πρώτος.

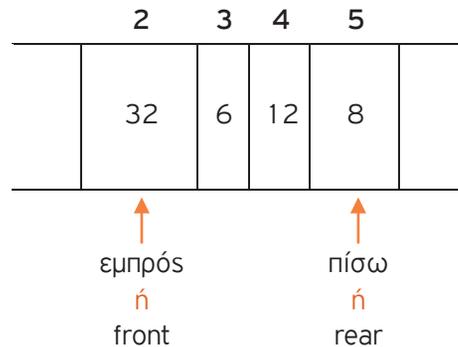


Οι **κύριες λειτουργίες** που εκτελούνται σε μια ουρά είναι δύο:

1. Η **εισαγωγή** (enqueue) στοιχείου στο πίσω άκρο της ουράς.
2. Η **εξαγωγή** (dequeue) στοιχείου από το εμπρός άκρο της ουράς.

Υλοποίηση ουράς με χρήση μονοδιάστατου πίνακα

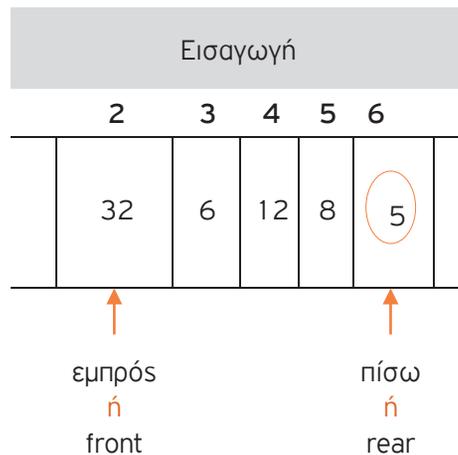
- Χρησιμοποιούμε δύο μεταβλητές, την front (ή εμπρός) που δείχνει τη θέση του 1^{ου} στοιχείου της ουράς και την rear (ή πίσω) που δείχνει τη θέση του τελευταίου στοιχείου. Ως αρχικές τιμές των μεταβλητών rear και front θεωρούμε το μηδέν.



- Η **εισαγωγή** ενός νέου στοιχείου γίνεται από το πίσω άκρο της ουράς και η τιμή της μεταβλητής rear αλλάζει ως εξής:

$$\mathbf{rear \leftarrow rear + 1}$$

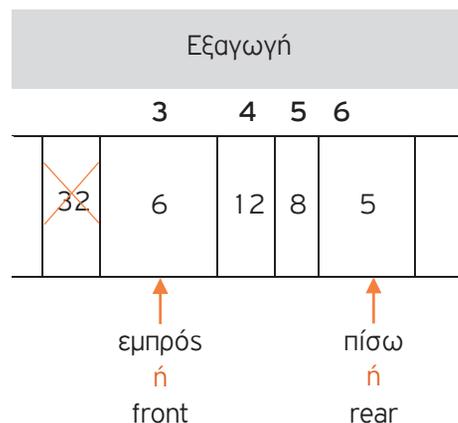
Κατά την εισαγωγή, πρώτα αυξάνουμε τον δείκτη rear κατά ένα και μετά εισάγουμε το στοιχείο στον πίνακα.



- Η **εξαγωγή** ενός στοιχείου γίνεται από το εμπρός άκρο της ουράς και η τιμή της μεταβλητής front αλλάζει ως εξής:

$$\mathbf{front \leftarrow front + 1}$$

Κατά την εξαγωγή ενός στοιχείου, αυξάνεται ο δείκτης front κατά ένα (δείχνει στην επόμενη θέση του πίνακα) χωρίς στην πραγματικότητα να γίνεται καμία παρέμβαση στα περιεχόμενα του πίνακα (χωρίς να διαγράφεται κάποιο στοιχείο).





Παράδειγμα 2 – Εισαγωγή στοιχείου σε ουρά με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που υλοποιεί την εισαγωγή στοιχείου σε ουρά, με χρήση μονοδιάστατου πίνακα A, 10 θέσεων.

Απάντηση



Κώδικας σε ΓΛΩΣΣΑ [1.4]

```

1  ΓΡΑΨΕ 'Δώσε στοιχείο για εισαγωγή στην ουρά A:'
2  ΔΙΑΒΑΣΕ στοιχείο
3  ΑΝ rear = 10 ΤΟΤΕ
4      ΓΡΑΨΕ 'Γεμάτη ουρά'
5  ΑΛΛΙΩΣ_ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
6      front<- 1
7      rear<-1
8      A[rear]<-στοιχείο
9  ΑΛΛΙΩΣ
10     rear <- rear + 1
11     A[rear]<-στοιχείο
12 ΤΕΛΟΣ_ΑΝ

```



Παράδειγμα 3 – Εξαγωγή στοιχείου από ουρά με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που υλοποιεί την εξαγωγή στοιχείου από ουρά, με χρήση μονοδιάστατου πίνακα A, 10 θέσεων.

Απάντηση



Κώδικας σε ΓΛΩΣΣΑ [1.5]

```

1  ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
2      ΓΡΑΨΕ 'Άδεια ουρά'
3  ΑΛΛΙΩΣ_ΑΝ (front = rear ) ΤΟΤΕ
4      ΓΡΑΨΕ 'Εξάγεται το στοιχείο:',A[front]
5      front <- 0
6      rear <- 0
7  ΑΛΛΙΩΣ
8      ΓΡΑΨΕ 'Εξάγεται το στοιχείο:',A[front]
9      front <- front + 1
10 ΤΕΛΟΣ_ΑΝ

```



Παράδειγμα 4 – Επιβίβαση και Αποβίβαση αυτοκινήτων σε πλοίο

Ένα οχηματαγωγό πλοίο με δύο διαφορετικές πόρτες, μία για την είσοδο και μία για την έξοδο των οχημάτων, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Τα οχήματα που **επιβιβάζονται πρώτα είναι και αυτά που θα αποβιβαστούν πρώτα**. Στο λιμάνι του Πειραιά προσέρχονται τα αυτοκίνητα για αναχώρηση.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:
 1. Επιβίβαση
 2. Αποβίβαση
 3. Έξοδος
2. Στην περίπτωση που επιλεγεί η **Επιβίβαση** το πρόγραμμα θα διαβάζει τον αριθμό κυκλοφορίας καθενός από τα οχήματα που επιβιβάζονται στο πλοίο και θα τον καταχωρίζει στην ουρά ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «**Υπάρχει όχημα για επιβίβαση; (N/O)**». Αν ο χρήστης απαντήσει N (=NAI), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει O (=OXI), τότε να σταματά η διαδικασία επιβίβασης και να επιστρέφει το πρόγραμμα στο μενού Επιλογής.
3. Στην περίπτωση που επιλεγεί η **Αποβίβαση** το πρόγραμμα θα εξάγει από την ουρά και θα εμφανίζει όλα τα αυτοκίνητα που αποβιβάστηκαν στην ΑΙΓΙΝΑ.

Απάντηση



Παρακάτω υλοποιείται το πρόγραμμα του παραδείγματος επιβίβασης/αποβίβασης σε οχηματαγωγό πλοίο, όπου η «πόρτα» για την είσοδο των οχημάτων στο πλοίο είναι διαφορετική από την «πόρτα» εξόδου των οχημάτων από αυτό.

Ανάλυση της λύσης

Σύμφωνα με την εκφώνηση της άσκησης πρέπει:

- Να υλοποιηθεί μια δομή δεδομένων FIFO (ουρά). Η υλοποίηση της ουράς θα πραγματοποιηθεί με μονοδιάστατο πίνακα 250 θέσεων ($n[250]$) όσο και το μέγιστο πλήθος επιβιβαζόμενων οχημάτων (250).
- Να δηλωθούν δύο βοηθητικές μεταβλητές (δείκτες) που δείχνουν την αρχή («**αρχ**») και το τέλος («**τέλος**») της ουράς. Οι αρχικές τιμές των δεικτών «**αρχ**» και «**τέλος**» θα είναι μηδέν.
- Στην περίπτωση της **επιβίβασης**, πρώτα θα αυξάνεται ο δείκτης «τέλος» κατά 1, για να δείχνει στην επόμενη κενή θέση της ουράς (τέλος<-τέλος+1). Σε αυτή την κενή θέση θα καταχωρίζεται ο αριθμός κυκλοφορίας («**αρχ**») του επιβιβαζόμενου οχήματος ($n[\text{τέλος}]<-αρχ$).
- Στην περίπτωση της **αποβίβασης**, πρώτα θα αποβιβάζεται το όχημα και μετά ο δείκτης «**αρχ**» θα αυξάνεται κατά 1 ($αρχ<-αρχ+1$), έτσι ώστε να δείχνει στη θέση που αντιστοιχίζεται στο επόμενο όχημα προς αποβίβαση.
- Να δηλωθούν και οι μεταβλητές: **επ1** (επιλογή μενού), **πλ** (μετρητής αποβιβαζόμενων οχημάτων), **επ2** (μεταβλητή δήλωσης οχήματος για επιβίβαση), **αφ** (αριθμός κυκλοφορίας οχήματος, μέσω του οποίου καταχωρίζεται-δηλώνεται η επιβίβασή του), **n[250]** (πίνακας 250 θέσεων, που φαίνεται το μέγιστο πλήθος των οχημάτων που μπορούν να χωρέσουν στο πλοίο).

- Να εμφανίζεται ένα μενού επιλογών με τις επιλογές: **1.** «Επιβίβαση», **2.** «Αποβίβαση», **3.** «Εξοδος» (δομή επανάληψης «**ΜΕΧΡΙΣ_ΟΤΟΥ...**», με τη βοήθεια της μεταβλητής «**επ1**» η οποία θα παίρνει τις τιμές 1 ή 2 ή 3).
- Στην περίπτωση της **επιβίβασης** («**ΑΝ επ1 = 1 ΤΟΤΕ**»):
 - ο Αρχικά, γίνεται έλεγχος αν το πλοίο είναι γεμάτο («**ΑΝ τελος=250 ΤΟΤΕ**»). Στην περίπτωση αυτή εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'**»)
 - ο Στη συνέχεια, θα πρέπει να ελεγχθεί αν υπάρχει όχημα για επιβίβαση. Ο έλεγχος πραγματοποιείται μέσω της μεταβλητής «**επ2**» («**ΔΙΑΒΑΣΕ επ2, ΑΝ επ2 <>'N'...**»).
 - ο Αν υπάρχει όχημα προς επιβίβαση, θα πρέπει να ελέγχεται αν είναι κενό το πλοίο/ουρά. Αν είναι κενό («**ΑΝ (αρχ=0 ΚΑΙ τελος=0) ΤΟΤΕ**»), αυξάνουμε τους δείκτες «**αρχ**» και «**τέλος**» για να δείχνουν στην πρώτη κενή θέση του πλοίου/ουράς («**αρχ<- 1, τελος<- 1**») και μετά γίνεται καταχώριση του αριθμού κυκλοφορίας του επιβιβαζόμενου οχήματος στη θέση αυτή (πρώτη θέση του πλοίου/ουράς) («**π[τελος]<- αρχ**»).

Στην περίπτωση που το πλοίο/ουρά δεν είναι κενό, αυξάνεται πρώτα η τιμή του πίσω δείκτη της ουράς («**τελος<-τελος+ 1**»), για να δείχνει σε κενή θέση κι έπειτα καταχωρίζεται σε αυτή ο αριθμός κυκλοφορίας του επιβιβαζόμενου οχήματος. Η διαδικασία αυτή επαναλαμβάνεται όσο υπάρχουν οχήματα προς επιβίβαση και το πλοίο δεν έχει γεμίσει («**ΜΕΧΡΙΣ_ΟΤΟΥ τελος =250 Η επ2='Ο' Η επ2='ο'**»).

Στην περίπτωση που διακοπεί η επαναληπτική διαδικασία της επιβίβασης λόγω «γεμίματος» του πλοίου («**ΑΝ τελος=250 ΤΟΤΕ**»), εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'**»).
- Στην περίπτωση της **αποβίβασης** («**ΑΛΛΙΩΣ_ΑΝ επ1 =2 ΤΟΤΕ**»):
 - ο Θα πρέπει πρώτα να ελεγχθεί αν το πλοίο/ουρά είναι κενό («**ΑΝ (αρχ=0 ΚΑΙ τελος=0) ΤΟΤΕ**») και στην περίπτωση αυτή να εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ 'Το πλοίο είναι άδειο'**»)
 - ο Έπειτα, θα πρέπει να γίνει έλεγχος της περίπτωσης που το πλοίο/ουρά έχει ένα μόνο όχημα («**ΑΛΛΙΩΣ_ΑΝ αρχ=τελος ΤΟΤΕ**»). Στην περίπτωση αυτή πρώτα αποβιβάζεται το όχημα από το πλοίο («εξάγεται» από την ουρά, «**ΓΡΑΨΕ 'Αποβιβάζεται το μοναδικό όχημα:', π[αρχ]**») και μετά μηδενίζουμε τους δείκτες («**αρχ<-0, τέλος<-0»), αφού το πλοίο/ουρά είναι πλέον κενό.**

Σημειώνεται ότι για λόγους παρακολούθησης των τιμών του πίνακα, στη θέση του πίνακα που κατείχε το αποβιβαζόμενο όχημα καταχωρίζεται ο κενός χαρακτήρας («**π[αρχ]<- ' '**»).

 - ο Αν υπάρχουν περισσότερα από ένα οχήματα για αποβίβαση, μέσα σε μια επαναληπτική διαδικασία πρώτα εμφανίζεται το όχημα που αποβιβάζεται («**ΓΡΑΨΕ 'Αποβιβάζεται το όχημα: ', π[αρχ]**») κι έπειτα ο μπροστινός δείκτης «**αρχ**» αυξάνεται κατά 1 («**αρχ<-αρχ+ 1**») και δείχνει στη θέση του επόμενου οχήματος προς αποβίβαση. Η επαναληπτική διαδικασία αποβίβασης ολοκληρώνεται, όταν πλέον δεν υπάρχει κανένα όχημα για αποβίβαση, δηλαδή όταν ο δείκτης «**αρχ**» γίνει μεγαλύτερος από τον πίσω δείκτη «**τέλος**» της ουράς («**ΜΕΧΡΙΣ_ΟΤΟΥ (αρχ>τελος)**»).
 - ο Τέλος, με την ολοκλήρωση της αποβίβασης όλων των οχημάτων, εκτυπώνεται με κατάλληλο μήνυμα το πλήθος αποβιβασθέντων οχημάτων στην ΑΙΓΙΝΑ («**ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: ', πλ**»).

- ο Σε κάθε νέα εκκίνηση αποβίβασης οχημάτων, ο μετρητής «πλ» πρέπει να μηδενίζεται, για να είναι σωστό το σύνολο των αποβιβασθέντων οχημάτων της τρέχουσας αποβίβασης. Π.χ. εκτελείται το πρόγραμμα και επιβιβάζονται 15 οχήματα, τα οποία στη συνέχεια αποβιβάζονται. Στη συνέχεια, επιλέγεται, από το μενού, νέα επιβίβαση 30 οχημάτων, τα οποία στη συνέχεια αποβιβάζονται. Αν ο μετρητής «πλ» δεν μηδενίζεται σε κάθε νέα εκκίνηση αποβίβασης, στην πρώτη αποβίβαση θα είχε την τιμή «15» και στη δεύτερη την τιμή «45» και όχι την τιμή «30» που είναι η σωστή. Δηλαδή, στη δεύτερη αποβίβαση οχημάτων το πρόγραμμα θα τύπωνε λανθασμένα «Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: 45» και όχι το σωστό «Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: 30» («**ΓΡΑΨΕ** `Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: `, πλ»).
Επίσης, αν επιλεχθεί νέα επιβίβαση μετά την ολοκλήρωση κάθε αποβίβασης, θα πρέπει η ουρά να γεμίζει από την αρχή. Κάτι τέτοιο επιτυγχάνεται με τον μηδενισμό των τιμών των δύο δεικτών, καθώς παίρνουν την ίδια τιμή («αρχ <- 0», «τελος <- 0»).

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.



Κώδικας σε ΓΛΩΣΣΑ [1.6]

ΠΡΟΓΡΑΜΜΑ πλοίο2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: αρχ, τελος, επ1, πλ

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]

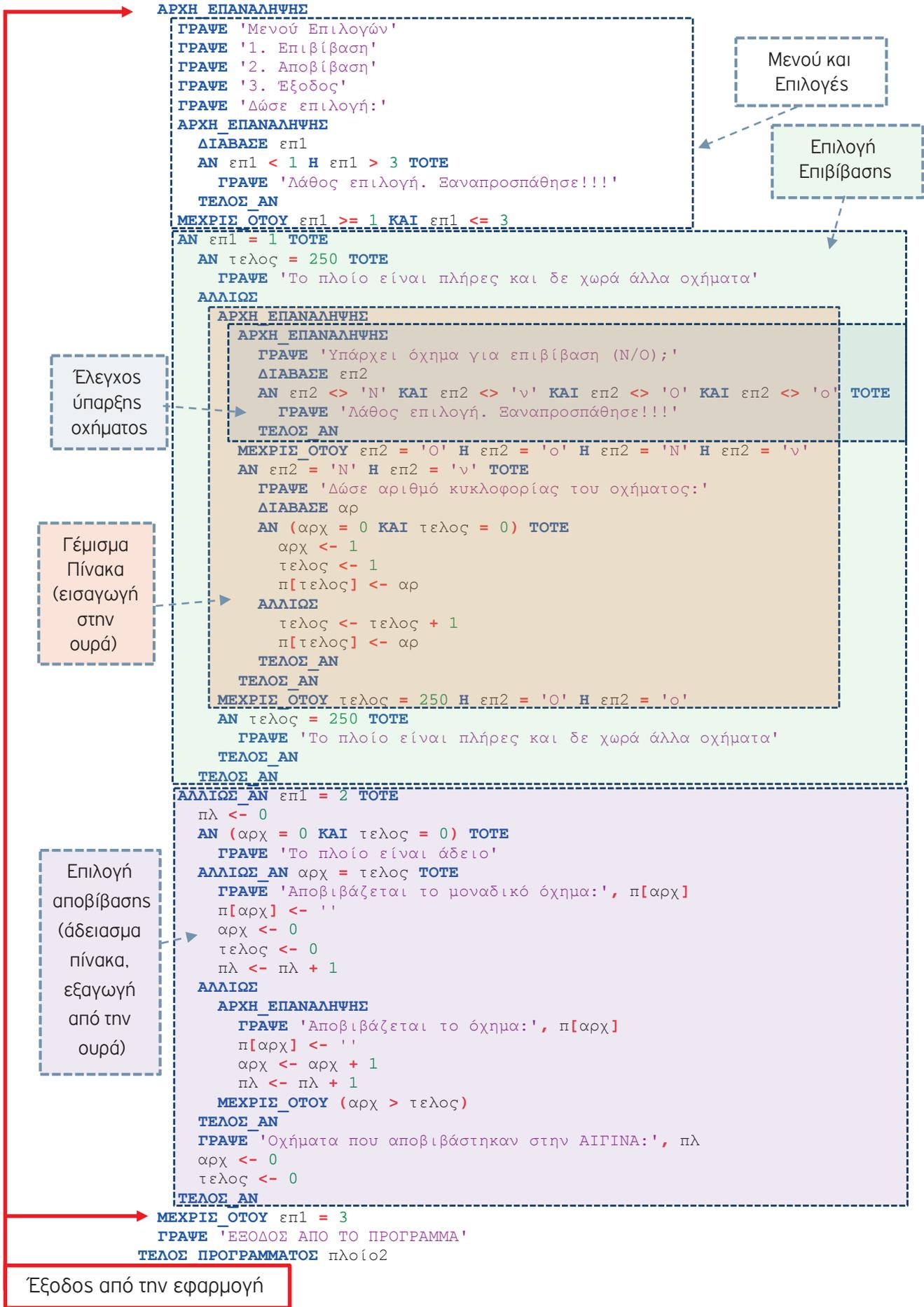
ΑΡΧΗ

αρχ <- 0

τελος <- 0

Δήλωση Μεταβλητών

Αρχικοποίηση
Μεταβλητών



```

1  ΠΡΟΓΡΑΜΜΑ πλοίο2
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: αρχ, τελος, επ1, πλ
4      ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
5  ΑΡΧΗ
6      αρχ <- 0
7      τελος <- 0
8  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
9      ΓΡΑΨΕ 'Μενού Επιλογών'
10     ΓΡΑΨΕ '1. Επιβίβαση'
11     ΓΡΑΨΕ '2. Αποβίβαση'
12     ΓΡΑΨΕ '3. Έξοδος'
13     ΓΡΑΨΕ 'Δώσε επιλογή:'
14  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
15     ΔΙΑΒΑΣΕ επ1
16     ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
17         ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
18     ΤΕΛΟΣ_ΑΝ
19  ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
20  ΑΝ επ1 = 1 ΤΟΤΕ
21     ΑΝ τελος = 250 ΤΟΤΕ
22         ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
23     ΑΛΛΙΩΣ
24     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
25         ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
26             ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (N/O);'
27             ΔΙΑΒΑΣΕ επ2
28             ΑΝ επ2<>'N' ΚΑΙ επ2<>'n' ΚΑΙ επ2<>'O' ΚΑΙ επ2<>'o' ΤΟΤΕ
29                 ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
30             ΤΕΛΟΣ_ΑΝ
31         ΜΕΧΡΙΣ_ΟΤΟΥ επ2='O' Η επ2='o' Η επ2='N' Η επ2='n'
32         ΑΝ επ2 = 'N' Η επ2 = 'n' ΤΟΤΕ
33             ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
34             ΔΙΑΒΑΣΕ αρ
35             ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
36                 αρχ <- 1
37                 τελος <- 1
38                 π[τελος] <- αρ
39             ΑΛΛΙΩΣ
40                 τελος <- τελος + 1

```

```

41         π[τελος] <- αρ
42         ΤΕΛΟΣ_ΑΝ
43         ΤΕΛΟΣ_ΑΝ
44         ΜΕΧΡΙΣ_ΟΤΟΥ τελος = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
45         ΑΝ τελος = 250 ΤΟΤΕ
46             ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
47         ΤΕΛΟΣ_ΑΝ
48         ΤΕΛΟΣ_ΑΝ
49     ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
50         πλ <- 0
51         ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
52             ΓΡΑΨΕ 'Το πλοίο είναι άδειο'
53         ΑΛΛΙΩΣ_ΑΝ αρχ = τελος ΤΟΤΕ
54             ΓΡΑΨΕ 'Αποβιβάζεται το μοναδικό όχημα:', π[αρχ]
55             π[αρχ] <- ''
56             αρχ <- 0
57             τελος <- 0
58             πλ <- πλ + 1
59         ΑΛΛΙΩΣ
60             ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
61                 ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[αρχ]
62                 π[αρχ] <- ''
63                 αρχ <- αρχ + 1
64                 πλ <- πλ + 1
65             ΜΕΧΡΙΣ_ΟΤΟΥ (αρχ > τελος)
66             ΤΕΛΟΣ_ΑΝ
67             ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ:', πλ
68             αρχ <- 0
69             τελος <- 0
70         ΤΕΛΟΣ_ΑΝ
71     ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
72     ΓΡΑΨΕ 'ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ'
73 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο2

```

1.2.2 Ερωτήσεις - Ασκήσεις

E. 1: Να δώσετε παραδείγματα ουράς από την καθημερινή ζωή.

E. 2: Δίνεται η επόμενη ακολουθία αριθμών : 4, 8, 2, 5, 9, 13.

1. Ποια λειτουργία θα χρησιμοποιήσετε για την τοποθέτηση των αριθμών σε ουρά;
2. Να σχεδιάσετε την ουρά έπειτα από την τοποθέτηση των αριθμών.
3. Ποια λειτουργία θα χρησιμοποιήσετε για την εξαγωγή των αριθμών από την ουρά;
4. Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στην ουρά για να εξαχθεί ο αριθμός 5;

E. 3:

1. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: X, A, B, A, P στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
 - i. Να προσδιορίσετε τις τιμές των δεικτών rear και front της παραπάνω ουράς και να τη σχεδιάσετε.
 - ii. Αν εφαρμόσουμε τις ακόλουθες λειτουργίες: **Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή X, Εισαγωγή Δ** και **Εξαγωγή** ποιες είναι τις τιμές των δεικτών rear και front της παραπάνω ουράς και ποια η τελική μορφή της ουράς;
2. Σε μια κενή ουρά 10 θέσεων εισάγουμε τα στοιχεία K, Φ, I, A,P. Με ποιον τρόπο πρέπει να «εισαχθούν» και να «εξαχθούν» τα στοιχεία, ώστε να έχουμε ως έξοδο τα δεδομένα A, P, X, H.

E. 4: Σε μια τράπεζα χρησιμοποιείται αυτόματο ηλεκτρονικό μηχάνημα που το χειρίζονται οι πελάτες, οι ταμίες και ο διευθυντής της τράπεζας. Κάθε ένας από τους χειριστές του μηχανήματος έχει δικαιώματα χρήσης συγκεκριμένων πλήκτρων του πληκτρολογίου.

Ο πελάτης το «**Π**», οι ταμίες το «**1**» ή το «**2**» ή το «**3**» ή το «**4**» αναλόγως της θέσης του ταμείου που εργάζονται και ο διευθυντής το πλήκτρο «**Δ**». Κατά την είσοδό του, ο κάθε πελάτης πατάει το πλήκτρο «**Π**» και εκτυπώνεται ένα χαρτί, στο οποίο αναγράφεται το νούμερο που έχει στην ουρά από την αρχή της ημέρας.

Η τράπεζα έχει 4 ταμεία, όπου όταν ο ταμίας εξυπηρετεί έναν πελάτη, πατάει το νούμερο του ταμείου του, «**1**» ή «**2**» ή «**3**» ή «**4**».

Ο διευθυντής της τράπεζας, πατώντας το κουμπί «**Δ**», σταματά τη διαδικασία εξυπηρέτησης των πελατών και μπορεί να δει το σύνολο των πελατών που έχουν ήδη εξυπηρετηθεί από το κάθε ταμείο. Ο μέγιστος αριθμός πελατών που μπορεί να εξυπηρετήσει η τράπεζα είναι 1.000 πελάτες.

Λαμβάνοντας υπόψη ότι ο πρώτος πελάτης εξυπηρετείται πρώτος και ο τελευταίος εξυπηρετείται τελευταίος, να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, όπου:

1. Να υπάρχει μενού επιλογής:
Π. Πελάτης Τ. Ταμίας Δ. Διευθυντής
 2. Στην περίπτωση που επιλεχθεί από το μενού το **Π. Πελάτης**, το πρόγραμμα εκτυπώνει το νούμερο που έχει στην ουρά (από την αρχή της ημέρας).
 3. Στην περίπτωση που επιλεχθεί από το μενού το **Τ. Ταμίας**, ο/η αρμόδιος/-α υπάλληλος επιλέγει το νούμερο του ταμείου που του/της αντιστοιχεί: «**1**» ή «**2**» ή «**3**» ή «**4**» και ο πελάτης διαγράφεται από την ουρά.
 4. Στην περίπτωση που επιλεχθεί από το μενού το **Δ. Διευθυντής**, σταματά η διαδικασία εξυπηρέτησης και το πρόγραμμα τυπώνει το νούμερο του ταμείου που εξυπηρέτησε τους περισσότερους πελάτες.
- Στο πρόγραμμα να γίνεται έλεγχος των δεδομένων εισόδου.

E. 5: Μια αεροπορική εταιρεία εκτελεί το δρομολόγιο Αθήνα – Θεσσαλονίκη κατά την περίοδο του Σεπτεμβρίου. Λόγω της Δ.Ε.Θ. υπάρχει αυξημένη ζήτηση και η εταιρεία διατηρεί λίστα αναμονής για τους επιβάτες που δεν πρόλαβαν να κλείσουν εισιτήριο, ώστε αν προκύψει κάποια ακύρωση, να ενημερώσει τον πρώτο στη σειρά πελάτη που εισήχθη στη λίστα αναμονής προκειμένου να κλείσει εισιτήριο. Η λίστα αναμονής δεν μπορεί να περιλαμβάνει περισσότερα από 10 ονόματα.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υπάρχει μενού επιλογής: **1. ΕΓΓΡΑΦΗ 2. ΑΚΥΡΩΣΗ 3. ΤΕΛΟΣ**.
2. Αν ο χρήστης επιλέξει την τιμή «**1.ΕΓΓΡΑΦΗ**», τότε θα ζητείται το όνομα του χρήστη και θα καταχωρίζεται στη λίστα αναμονής, εφόσον η λίστα αναμονής δεν έχει γεμίσει. Διαφορετικά, θα εμφανίζεται το μήνυμα: «Η λίστα αναμονής είναι πλήρης».
3. Αν ο χρήστης επιλέξει την τιμή «**2.ΑΚΥΡΩΣΗ**», τότε κάποιος από τους επιβάτες της πτήσης έχει ακυρώσει την κράτησή του, συνεπώς, το πρόγραμμα θα πρέπει να εμφανίσει το όνομα του ατόμου που είναι το πρώτο διαθέσιμο στη λίστα αναμονής. Αν δεν υπάρχουν άτομα στη λίστα αναμονής, εμφανίζεται το μήνυμα «Η λίστα αναμονής είναι άδεια».
4. Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι ο χρήστης να επιλέξει την τιμή «**3.ΤΕΛΟΣ**». Το πρόγραμμα εμφανίζει το πλήθος των ατόμων που κατάφεραν να κάνουν κράτηση μέσα από την λίστα αναμονής, καθώς και το μέγιστο πλήθος των ατόμων που περίμεναν στην ουρά αναμονής.

Στο πρόγραμμα να γίνεται έλεγχος εγκυρότητας των τιμών που πληκτρολογούνται.

E. 6: Σε ένα ταχυδρομικό κατάστημα, οι πελάτες εξυπηρετούνται με βάση τη σειρά άφιξής τους σε αυτό. Το ταχυδρομικό κατάστημα έχει ένα ταμείο και ο μέσος χρόνος εξυπηρέτησης κάθε πελάτη είναι 3 λεπτά. Η ουρά αναμονής στο κατάστημα δεν μπορεί να ξεπερνά τα 30 άτομα.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να δέχεται σαν είσοδο από τον χρήστη μία εκ των δύο τιμών εισαγωγής: «**1.ΕΙΣΑΓΩΓΗ**» ή «**2.ΕΠΟΜΕΝΟΣ**» (με έλεγχο εγκυρότητας).

2. Αν δοθεί η τιμή «**1.ΕΙΣΑΓΩΓΗ**», τότε το πρόγραμμα να διαβάξει το ονοματεπώνυμο του πελάτη και αμέσως μετά να εμφανίζει το πλήθος των ατόμων που περιμένουν πριν από αυτόν, εκτός αν η ουρά αναμονής είναι γεμάτη, οπότε να εμφανίζει το μήνυμα «Το κατάστημα γέμισε. Παρακαλούμε ελάτε άλλη φορά».
3. Αν δοθεί η τιμή «**2.ΕΠΟΜΕΝΟΣ**», τότε το πρόγραμμα να εμφανίζει το ονοματεπώνυμο του πελάτη προς εξυπηρέτηση.
4. Η παραπάνω διαδικασία να επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλοι οι πελάτες.
5. Στο τέλος το πρόγραμμα να εμφανίζει το πλήθος των ατόμων που εξυπηρετήθηκαν, καθώς και τον μέσο χρόνο αναμονής των πελατών.

E. 7: Ένας εκτυπωτής χρησιμοποιεί μια ουρά εκτύπωσης για να τοποθετεί σε αυτήν τα αρχεία που έχουν σταλεί προς εκτύπωση με τη σειρά που αυτά στάλθηκαν. Κάθε φορά εκτυπώνει το αρχείο που βρίσκεται στην αρχή της ουράς εκτύπωσης, το οποίο και εξάγει. Λόγω της περιορισμένης μνήμης του εκτυπωτή, θεωρούμε ότι στην ουρά μπορούν να εισαχθούν το πολύ 15 αρχεία.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να διαβάξει επαναληπτικά, με έλεγχο εγκυρότητας, το γράμμα "N" που καθορίζει την έλευση νέου αρχείου ή το γράμμα "E" που δηλώνει την προσπάθεια εκτύπωσης ενός αρχείου.
2. Κατά την έλευση ενός αρχείου, διαβάξει το όνομά του και εξετάζει αν υπάρχει ο διαθέσιμος χώρος στην ουρά και το αρχείο καταχωρίζεται σε αυτήν με τη διαδικασία της εισαγωγής. Στην περίπτωση που δεν υπάρχει χώρος, εμφανίζεται το μήνυμα «Η ουρά γέμισε. Δε μπορεί να εκτυπωθεί το αρχείο».
3. Όταν ο χρήστης δώσει το γράμμα "E", εξετάζει αν υπάρχουν αρχεία προς εκτύπωση και στην περίπτωση αυτή εξάγεται το κατάλληλο αρχείο εμφανίζοντας τη λέξη «Εκτύπωση» ακολουθούμενη από το όνομα του αρχείου που τυπώνεται.
4. Η επαναληπτική διαδικασία ολοκληρώνεται, όταν εκτυπωθούν όλα τα αρχεία που έχουν τοποθετηθεί στην ουρά.
5. Μετά το τέλος της διαδικασίας, το πρόγραμμα εμφανίζει τον συνολικό αριθμό των αρχείων που εκτυπώθηκαν.

Ε. 8: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση της ουράς μπορεί να χρησιμοποιηθεί πίνακας.	<input type="checkbox"/>	<input type="checkbox"/>
2	Κατά την εισαγωγή ενός στοιχείου σε ουρά, αυτό τοποθετείται στο μπροστινό άκρο της.	<input type="checkbox"/>	<input type="checkbox"/>
3	Σε μια ουρά κάθε στοιχείο της εξάγεται από το μπροστινό άκρο της.	<input type="checkbox"/>	<input type="checkbox"/>
4	Η απώθηση είναι μια από τις λειτουργίες της ουράς.	<input type="checkbox"/>	<input type="checkbox"/>
5	Η εισαγωγή και η εξαγωγή είναι οι δύο βασικές λειτουργίες της ουράς.	<input type="checkbox"/>	<input type="checkbox"/>
6	Στην ουρά το στοιχείο που μπαίνει πρώτο βγαίνει και πρώτο.	<input type="checkbox"/>	<input type="checkbox"/>
7	Η υλοποίηση της ουράς χρησιμοποιεί μία μεταβλητή-δείκτη για την εκτέλεση των δύο βασικών λειτουργιών της.	<input type="checkbox"/>	<input type="checkbox"/>
8	Η λειτουργία της εξαγωγής μπορεί να εκτελεστεί σε μια γεμάτη ουρά.	<input type="checkbox"/>	<input type="checkbox"/>