

29.1. Η ΔΟΜΗ ΔΕΔΟΜΕΝΩΝ «ΟΥΡΑ»

1. Να περιγράψετε τη δομή δεδομένων «Ουρά» χρησιμοποιώντας ένα παράδειγμα από την καθημερινή ζωή.

Απάντηση:

Για να κατανοήσουμε τη λειτουργία μιας ουράς δεδομένων, θα χρησιμοποιήσουμε ως παράδειγμα μία ουρά ανθρώπων που περιμένουν να εξυπηρετηθούν. Κάθε άτομο που εισέρχεται τοποθετείται στο τέλος της ουράς, ενώ το άτομο που θα εξυπηρετηθεί είναι αυτό που βρίσκεται στην αρχή της ουράς. Επίσης, το άτομο που στάθηκε πρώτο στην ουρά, θα εξυπηρετηθεί και πρώτο. Τα ίδια ισχύουν σε μια ουρά δεδομένων: κάθε νέο δεδομένο εισάγεται στο πίσω μέρος της ουράς και, αντίστοιχα, εξάγεται το δεδομένο που βρίσκεται στην αρχή της ουράς.

Παρατήρηση

Άλλα παραδείγματα είναι: η ουρά στα λεωφορεία, η ουρά στα ταμεία, η αναμονή των αυτοκινήτων σε ένα φωτεινό σηματοδότη κ.λπ.

2. Να δώσετε τον ορισμό της δομής δεδομένων «Ουρά».

Απάντηση:

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

3. Ποια μέθοδο επεξεργασίας δεδομένων χρησιμοποιεί η δομή δεδομένων «Ουρά»;

Απάντηση:

Τη μέθοδο Πρώτο Μέσα, Πρώτο Έξω, ή αλλιώς FIFO, από τα αρχικά των λέξεων «First In First Out». Κατά τη μέθοδο αυτήν το δεδομένο που εισάγεται πρώτο, είναι αυτό

που εξάγεται πρώτο. Αντίθετα, το δεδομένο που βρίσκεται στο τέλος της ουράς είναι και αυτό που εξάγεται τελευταίο.

4. Ποιες είναι οι κύριες λειτουργίες στη δομή δεδομένων «Ουρά»;

Απάντηση:

Στη δομή δεδομένων ουρά χρησιμοποιούνται δύο λειτουργίες:

- η εισαγωγή (enqueue) ενός στοιχείου στο πίσω άκρο της ουράς και
- η εξαγωγή (dequeue) ενός στοιχείου από το μπροστινό άκρο της ουράς.

5. Να δώσετε ένα παράδειγμα στο οποίο να φαίνεται η λειτουργία της δομής δεδομένων «Ουρά».

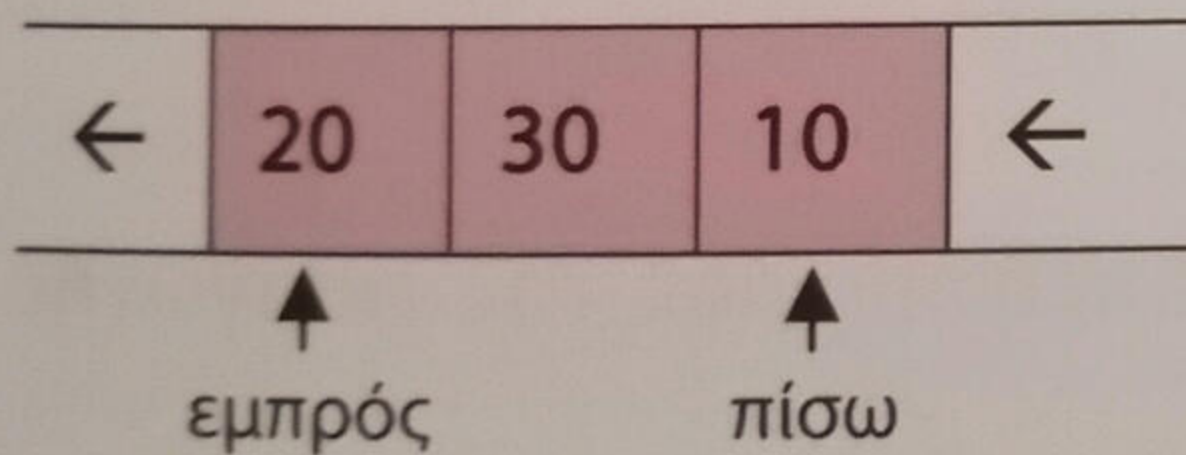
Απάντηση:



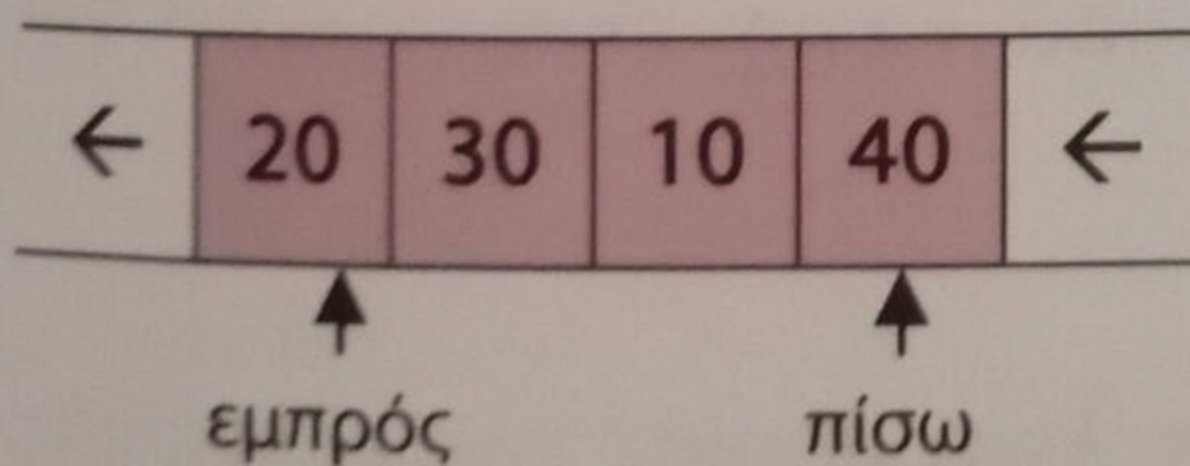
Για να κατανοήσουμε τη διάταξη των στοιχείων της δομής δεδομένων «Ουρά», μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων σε μια οριζόντια σειρά.

Μια ουρά τη σχεδιάζουμε οριζόντια και, για να τη διαχειριστούμε, χρησιμοποιούμε δύο μεταβλητές-δείκτες:

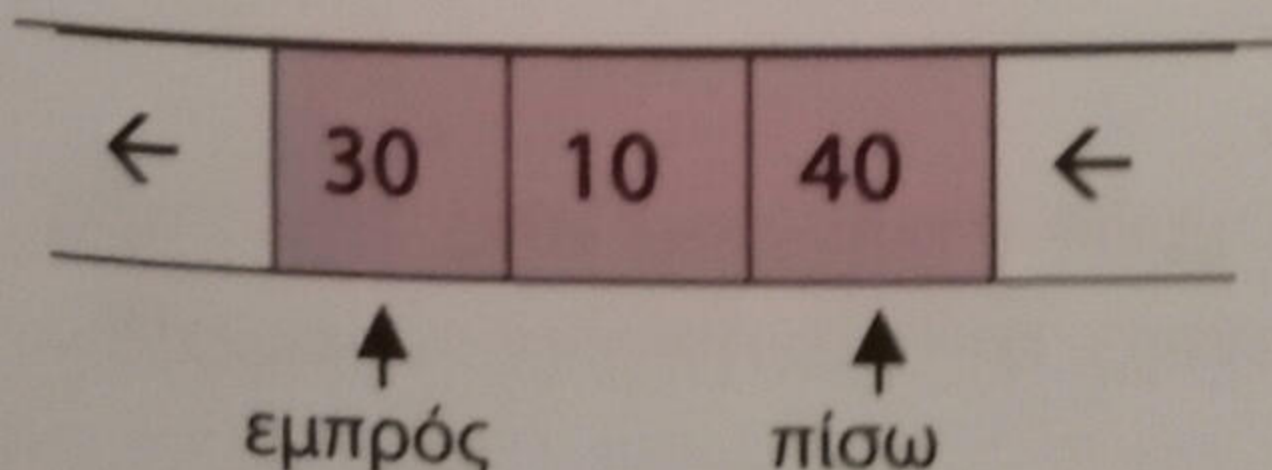
- τον δείκτη **εμπρός** (front), που δείχνει τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί, δηλαδή τη θέση του πρώτου στοιχείου της ουράς (εμπρός άκρο της ουράς),
- και τον δείκτη **πίσω** (rear), που δείχνει τη θέση του στοιχείου που μόλις εισήλθε, δηλαδή τη θέση του τελευταίου στοιχείου της ουράς (πίσω άκρο της ουράς).



Στο διπλανό σχήμα φαίνεται μία ουρά με τρία στοιχεία, με τον δείκτη «εμπρός» να δείχνει το εμπρός άκρο της ουράς, δηλαδή το στοιχείο 20, και τον δείκτη «πίσω» να δείχνει το πίσω άκρο της ουράς, δηλαδή το στοιχείο 10.



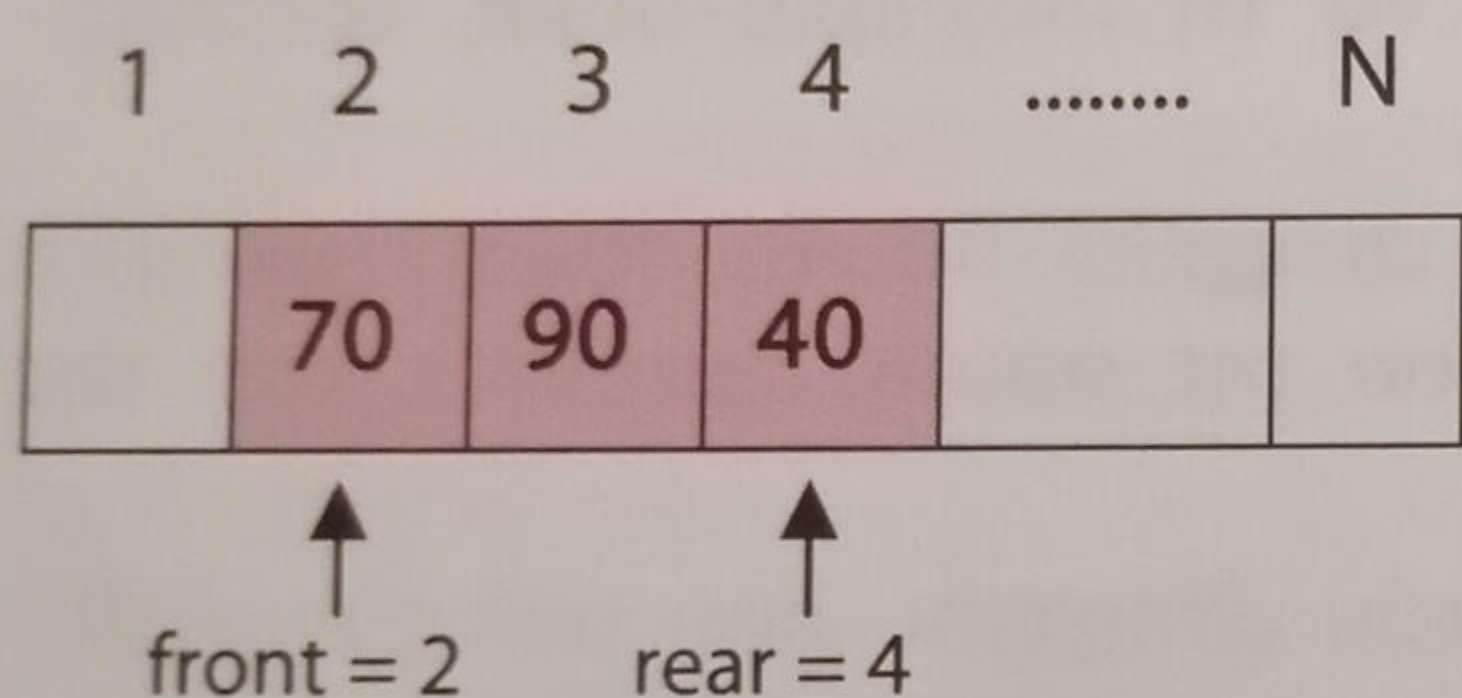
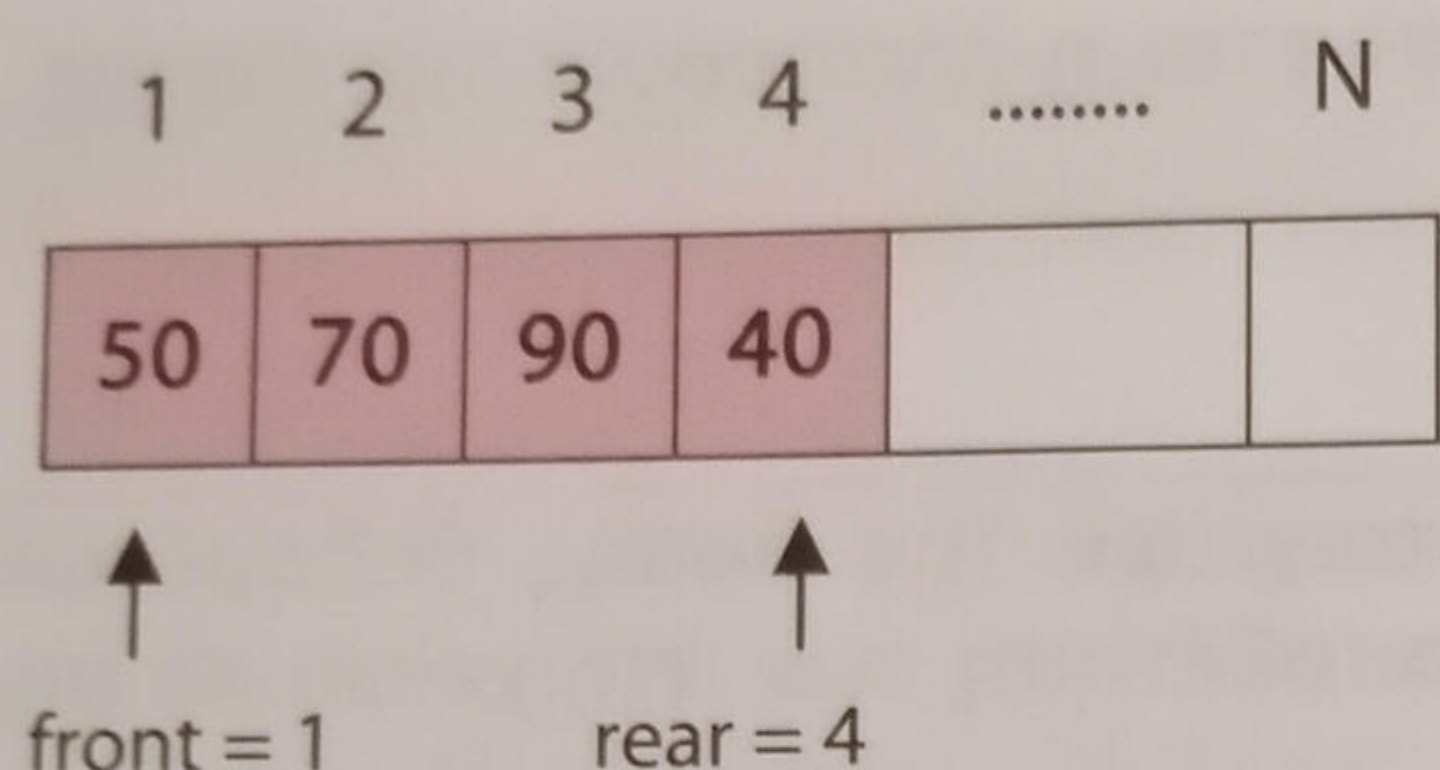
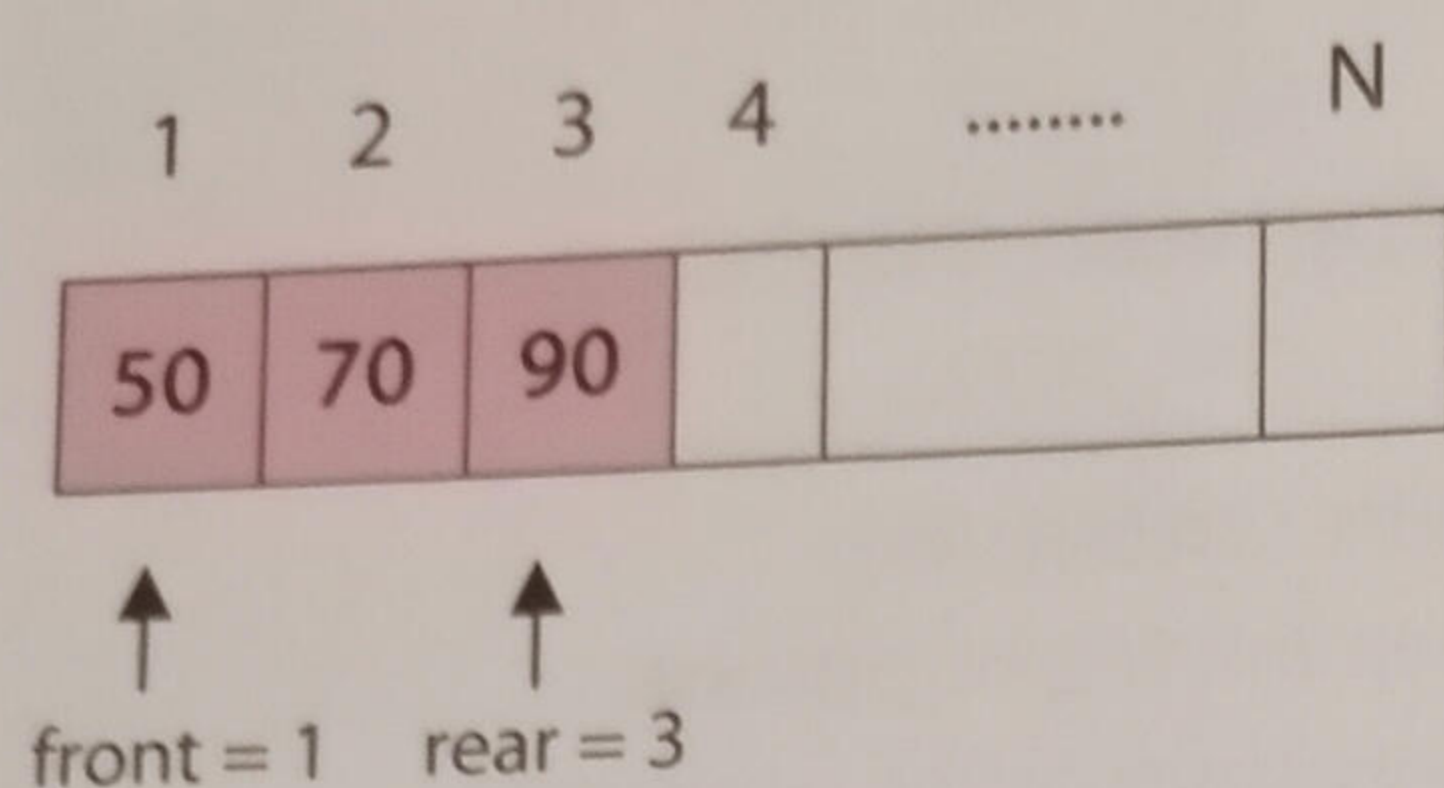
Για την τοποθέτηση ενός νέου στοιχείου στην ουρά π.χ. του αριθμού 40, εκτελούμε τη λειτουργία της εισαγωγής και το δεδομένο εισάγεται στο πίσω άκρο της ουράς. Ο δείκτης «πίσω» δείχνει τώρα το στοιχείο 40.



Για την αφαίρεση ενός στοιχείου, εκτελούμε τη λειτουργία της εξαγωγής και εξάγεται το στοιχείο που υπάρχει στην αρχή της ουράς, δηλαδή το στοιχείο 20. Ο δείκτης «εμπρός» δείχνει τώρα το στοιχείο 30.

6. Να εξηγήσετε πώς γίνεται η υλοποίηση της δομής δεδομένων «Ουρά» με τη βοήθεια ενός μονοδιάστατου πίνακα.

Απάντηση:



Τη δομή δεδομένων «Ουρά» μπορούμε να την υλοποιήσουμε με τη βοήθεια ενός μονοδιάστατου πίνακα N θέσεων και, για να τη διαχειριστούμε, θα χρησιμοποιήσουμε δύο δείκτες: τον δείκτη «εμπρός (front)», που δείχνει το πρώτο στοιχείο της ουράς, και τον δείκτη «πίσω (rear)», που δείχνει το τελευταίο στοιχείο της ουράς. Αν στην ουρά έχουν εισαχθεί τα στοιχεία 50, 70, 90, τότε η κατάσταση της ουράς και οι τιμές των δεικτών απεικονίζονται δίπλα.

Η εισαγωγή ενός στοιχείου γίνεται στο πίσω άκρο της ουράς. Για να βάλουμε ένα στοιχείο π.χ. τον αριθμό 40, εκτελούμε τη λειτουργία «εισαγωγή», κατά την οποία πρώτα αυξάνεται ο δείκτης «πίσω (rear)» κατά ένα και στη θέση που δείχνει στον πίνακα, τοποθετείται το νέο στοιχείο, δηλαδή το 40.

Η εξαγωγή ενός στοιχείου γίνεται από το εμπρός άκρο της ουράς. Για να βγάλουμε ένα στοιχείο από την ουρά, εκτελούμε τη λειτουργία «εξαγωγή», κατά την οποία εξέρχεται το στοιχείο που δείχνει ο δείκτης «εμπρός (front)», δηλαδή το 50 και στη συνέχεια ο δείκτης «εμπρός (front)» αυξάνεται κατά ένα για να δείχνει την επόμενη θέση του πίνακα στην οποία υπάρχει το ακόλουθο στοιχείο που θα εξαχθεί.

7. Τι πρέπει να προσέχουμε κατά την εκτέλεση της λειτουργίας της εισαγωγής ενός στοιχείου στην «Ουρά»;

Απάντηση:

Κατά την εκτέλεση της λειτουργίας της εισαγωγής ενός στοιχείου στην ουρά, πρέπει να εξετάζουμε αν υπάρχει διαθέσιμος χώρος στην ουρά (να μην είναι γεμάτη), για να τοποθετήσουμε το στοιχείο σε αυτήν.

8. Τι πρέπει να προσέχουμε κατά την εκτέλεση της λειτουργίας της εξαγωγής ενός στοιχείου από την «Ουρά»;

Απάντηση:

Κατά την εκτέλεση της λειτουργίας της εξαγωγής ενός στοιχείου από την ουρά, πρέπει να εξετάζουμε αν υπάρχει ένα τουλάχιστον στοιχείο σε αυτήν για να εξαχθεί, δηλαδή η ουρά να μην είναι άδεια.

- Σε μια κενή ουρά/πίνακα οι αρχικές τιμές των μεταβλητών «εμπρός/front» και «πίσω/rear» είναι μηδέν.
- Κατά την εισαγωγή ενός στοιχείου στην ουρά (εισαγωγή ενός στοιχείου στον πίνακα), πρώτα αυξάνεται η τιμή της μεταβλητής «πίσω/rear» κατά ένα, ($\text{rear} \leftarrow \text{rear} + 1$) και στη συνέχεια γίνεται η εισαγωγή του στοιχείου στη θέση του πίνακα όπου δείχνει ο δείκτης στο πίσω μέρος της ουράς.
- Κατά την εξαγωγή ενός στοιχείου από την ουρά αυξάνεται η τιμή της μεταβλητής «εμπρός/front» κατά ένα, ($\text{front} \leftarrow \text{front} + 1$) για να δείχνει την επόμενη θέση του πίνακα στην οποία υπάρχει το επόμενο στοιχείο που θα εξαχθεί. Στην πραγματικότητα δεν γίνεται καμία παρέμβαση στα περιεχόμενα του πίνακα και δεν διαγράφεται κανένα στοιχείο. Απλώς η μεταβλητή «εμπρός/front» δείχνει στην επόμενη θέση του πίνακα.
- Η δομή δεδομένων «Ουρά» θεωρείται ως στατική δομή για τη ΓΛΩΣΣΑ, επειδή υλοποιείται με πίνακα.

ΜΕΘΟΔΟΛΟΓΙΕΣ

29.1. Αλγόριθμος υλοποίησης της λειτουργίας της εισαγωγής.

Έστω ότι για την υλοποίηση μιας ουράς χρησιμοποιούμε, τον μονοδιάστατο πίνακα *table* με *N* θέσεις (το *N* είναι ένας οποιοσδήποτε ακέραιος θετικός αριθμός), τη μεταβλητή *front*, που περιέχει τη θέση του στοιχείου του πίνακα, στο οποίο υπάρχει το πρώτο στοιχείο της ουράς, και τη μεταβλητή *rear*, που περιέχει τη θέση του στοιχείου του πίνακα, στο οποίο υπάρχει το τελευταίο στοιχείο της ουράς.

ΓΑΔ

Για την εισαγωγή ενός στοιχείου *X* εξετάζουμε αν ο πίνακας/ουρά:

- είναι γεμάτος, ελέγχοντας την τιμή του δείκτη *rear* αν είναι ίση με το μέγεθος του πίνακα ($\text{AN rear} = \text{N TOTE}$),
- είναι άδειος, ελέγχοντας αν οι δύο δείκτες έχουν την τιμή μηδέν ($\text{ΑΛΛΙΩΣ_AN front} = 0 \text{ ΚΑΙ rear} = 0 \text{ TOTE}$) οπότε οι δύο δείκτες παίρνουν την τιμή 1, καθώς είναι το πρώτο στοιχείο που εισάγεται στη θέση *rear* ($\text{table[rear]} \leftarrow X$),
- έχει χώρο (ΑΛΛΙΩΣ), οπότε αυξάνεται ο πίσω δείκτης κατά ένα ($\text{rear} \leftarrow \text{rear} + 1$) και στη θέση που δείχνει τοποθετείται το νέο στοιχείο ($\text{table[rear]} \leftarrow X$).

Το τμήμα εντολών που υλοποιεί την εισαγωγή φαίνεται δίπλα.

ΔΙΑΒΑΣΕ *X* !αν ουρά γεμάτη
 AN $\text{rear} = \text{N}$ TOTE

ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΑΛΛΙΩΣ_ΑΝ $\text{front} = 0$ ΚΑΙ $\text{rear} = 0$
 TOTE !αν ουρά άδεια

$\text{front} \leftarrow 1$
 $\text{rear} \leftarrow 1$
 $\text{table[rear]} \leftarrow X$

ΑΛΛΙΩΣ ! έχει στοιχεία
 $\text{rear} \leftarrow \text{rear} + 1$! και χωράει
 $\text{table[rear]} \leftarrow X$
 ΤΕΛΟΣ_ΑΝ

Για την εισαγωγή πολλών στοιχείων μέχρι η ουρά να γεμίσει, μπορούμε να χρησιμοποιήσουμε έναν από τους παρακάτω τρόπους:

! ο front και ο rear έχουν αρχικές τιμές
 AN rear = N ΤΟΤΕ
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΑΛΛΙΩΣ
 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ! επανάληψη
 ΔΙΑΒΑΣΕ X
 AN front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
 front $\leftarrow$ 1 ! αν ουρά άδεια
 rear $\leftarrow$ 1
 table[rear] $\leftarrow$ X
 ΑΛΛΙΩΣ ! αν ουρά έχει χώρο
 rear $\leftarrow$ rear + 1
 table[rear] $\leftarrow$ X
 ΤΕΛΟΣ_ΑΝ
 ΜΕΧΡΙΣ_ΟΤΟΥ rear = N ! μέχρι να γεμίσει
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΤΕΛΟΣ_ΑΝ

! ο front και ο rear έχουν αρχικές τιμές
 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
 ΔΙΑΒΑΣΕ X
 AN front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
 front $\leftarrow$ 1
 rear $\leftarrow$ 1
 table[rear] $\leftarrow$ X
 ΑΛΛΙΩΣ_ΑΝ rear < N ΤΟΤΕ
 rear $\leftarrow$ rear + 1
 table[rear] $\leftarrow$ X
 ΤΕΛΟΣ_ΑΝ
 AN rear = N ΤΟΤΕ
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΤΕΛΟΣ_ΑΝ
 ΜΕΧΡΙΣ_ΟΤΟΥ rear = N

Για την εισαγωγή πολλών στοιχείων μετά από ερώτηση ή μέχρι η ουρά να γεμίσει, μπορούμε να χρησιμοποιήσουμε έναν από τους παρακάτω τρόπους:

! ο front και ο rear έχουν αρχικές τιμές
 AN rear = N ΤΟΤΕ
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΑΛΛΙΩΣ
 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Θέλεις να εισάγεις στοιχείο;'
 ΔΙΑΒΑΣΕ απάντηση
 AN απάντηση = 'ΝΑΙ' ΤΟΤΕ
 ΔΙΑΒΑΣΕ X
 AN front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
 front $\leftarrow$ 1
 rear $\leftarrow$ 1
 table[rear] $\leftarrow$ X
 ΑΛΛΙΩΣ
 rear $\leftarrow$ rear + 1
 table[rear] $\leftarrow$ X
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΑΝ
 ΜΕΧΡΙΣ_ΟΤΟΥ απάντηση = 'ΟΧΙ' Ή rear = N
 AN rear = N ΤΟΤΕ
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΤΕΛΟΣ_ΑΝ
 ΕΛΟΣ_ΑΝ

! ο front και ο rear έχουν αρχικές τιμές
 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Θέλεις να εισάγεις στοιχείο;'
 ΔΙΑΒΑΣΕ απάντηση
 AN απάντηση = 'ΝΑΙ' ΤΟΤΕ
 ΔΙΑΒΑΣΕ X
 AN front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
 front $\leftarrow$ 1
 rear $\leftarrow$ 1
 table[rear] $\leftarrow$ X
 ΑΛΛΙΩΣ_ΑΝ rear < N ΤΟΤΕ
 rear $\leftarrow$ rear + 1
 table[rear] $\leftarrow$ X
 ΤΕΛΟΣ_ΑΝ
 AN rear = N ΤΟΤΕ
 ΓΡΑΨΕ 'Ουρά γεμάτη'
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΑΝ
 ΜΕΧΡΙΣ_ΟΤΟΥ απάντηση = 'ΟΧΙ' Ή rear = N

! Ο έλεγχος «AN rear = N ΤΟΤΕ» μπορεί
 ! να μπει και μετά τη ΜΕΧΡΙΣ_ΟΤΟΥ

Για την εξαγωγή ενός στοιχείου X εξετάζουμε αν ο πίνακας/ουρά:

- είναι άδειος, ελέγχοντας την τιμή των δύο δεικτών να είναι μηδέν ($front = 0$ ΚΑΙ $rear = 0$),
- έχει ένα μόνο στοιχείο, ελέγχοντας αν $front=rear$, οπότε εξάγεται το στοιχείο και οι δύο δείκτες γίνονται μηδέν γιατί αδειάζει η ουρά,
- έχει στοιχεία (ΑΛΛΙΩΣ), οπότε εξάγεται το πρώτο από μπροστά στοιχείο και αυξάνεται ο εμπρός δείκτης κατά ένα ($front \leftarrow front + 1$).

```

ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
  ΓΡΑΨΕ 'Ουρά άδεια'
ΑΛΛΙΩΣ_ ΑΝ front = rear ΤΟΤΕ
  ΓΡΑΨΕ ' Εξάγεται το', table[front]
  front ← 0
  rear ← 0
ΑΛΛΙΩΣ
  ΓΡΑΨΕ ' Εξάγεται το', table[front]
  front ← front + 1
ΤΕΛΟΣ_ΑΝ

```

Για την εξαγωγή πολλών στοιχείων, εκτελούμε μία επαναληπτική διαδικασία, μέχρι η ουρά να αδειάσει, όπου αυτό συμβαίνει όταν ο μπροστινός δείκτης $front$ γίνει μεγαλύτερος από τον πίσω δείκτη $rear$, δηλαδή ΜΕΧΡΙΣ_ΟΤΟΥ $front > rear$.

! ο front και ο rear έχουν αρχικές τιμές

```
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
```

```
  ΓΡΑΨΕ 'Ουρά άδεια'
```

```
ΑΛΛΙΩΣ_ ΑΝ front = rear ΤΟΤΕ
```

```
  ΓΡΑΨΕ ' Εξάγεται το', table[front]
```

```
  front ← 0
```

```
  rear ← 0
```

```
ΑΛΛΙΩΣ
```

```
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
```

```
    Υ ← table[front]
```

```
    ΓΡΑΨΕ ' Εξάγεται το', Υ
```

```
    front ← front + 1
```

```
  ΜΕΧΡΙΣ_ΟΤΟΥ front > rear
```

```
  front ← 0      !εφόσον άδειασε η ουρά
```

```
  rear ← 0      ! οι δείκτες γίνονται μηδέν
```

```
ΤΕΛΟΣ_ΑΝ
```

! ο front και ο rear έχουν αρχικές τιμές

```
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
```

```
  ΓΡΑΨΕ 'Ουρά άδεια'
```

```
ΑΛΛΙΩΣ_ ΑΝ front = rear ΤΟΤΕ
```

```
  ΓΡΑΨΕ ' Εξάγεται το', table[front]
```

```
  front ← 0
```

```
  rear ← 0
```

```
ΑΛΛΙΩΣ
```

```
  ΟΣΟ front <= rear ΕΠΑΝΑΛΑΒΕ
```

```
    Υ ← table[front]
```

```
    ΓΡΑΨΕ ' Εξάγεται το', Υ
```

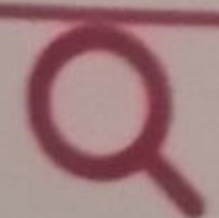
```
    front ← front + 1
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  front ← 0      !εφόσον άδειασε η ουρά
```

```
  rear ← 0      ! οι δείκτες γίνονται μηδέν
```

```
ΤΕΛΟΣ_ΑΝ
```



- Το πλήθος των στοιχείων μιας ουράς βρίσκεται από τη σχέση: $rear - front + 1$
- Από την τιμή των δύο δεικτών ($front$ και $rear$) καταλαβαίνουμε την κατάσταση της ουράς Π.χ. σε έναν πίνακα 100 θέσεων, που χρησιμοποιείται ως ουρά:
 - Αν $front=0$ και $rear=0$, σημαίνει ότι ο πίνακας δεν έχει κανένα στοιχείο μέσα, άρα η ουρά είναι άδεια.
 - Αν $front=1$ $rear=100$, θεωρούμε ότι ο πίνακας είναι γεμάτος με στοιχεία και άρα η ουρά είναι γεμάτη.
 - Αν $front = rear$, σημαίνει ότι ο πίνακας έχει ένα μόνο στοιχείο μέσα, άρα και η ουρά έχει ένα μόνο στοιχείο.

- Αν $0 < \text{rear} < 100$, σημαίνει ότι ο πίνακας έχει κάποια στοιχεία μέσα και η ουρά δεν είναι γεμάτη.
- Αν $\text{front} > 1$ και $\text{rear} = 100$, θεωρούμε ότι η ουρά είναι γεμάτη στο πίσω μέρος και έχει κενές θέσεις μπροστά.

ΠΑΡΑΔΕΙΓΜΑ

Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ που:

- Χρησιμοποιεί τον πίνακα ΟΥΡΑ[100] για την υλοποίηση ουράς 100 θέσεων.
- Εκτελεί τη λειτουργία της εισαγωγής και της εξαγωγής ως εξής:
Διαβάζει την επιλογή της λειτουργίας που θα εκτελεστεί («ΕΣ» για εισαγωγή και «ΕΞ» για εξαγωγή). Όταν πρέπει να εκτελεστεί η λειτουργία της εισαγωγής, διαβάζει έναν πραγματικό αριθμό και τον τοποθετεί στη ουρά, αν υπάρχει διαθέσιμος χώρος σε αυτήν, ενώ τη λειτουργία της εξαγωγής την εκτελεί, αν υπάρχουν στοιχεία στην ουρά. Η διαδικασία επαναλαμβάνεται, μέχρι να μη μπορεί να δεχθεί άλλους αριθμούς η ουρά, οπότε και εμφανίζει το μήνυμα «ΟΥΡΑ ΓΕΜΑΤΗ» ή επαναλαμβάνεται, μέχρι να αδειάσει η ουρά, οπότε και εμφανίζει το μήνυμα «ΑΔΕΙΑ ΟΥΡΑ».

Λύση:

ΠΡΟΓΡΑΜΜΑ Διαχείριση_ουράς
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : i, front, rear

ΠΡΑΓΜΑΤΙΚΕΣ: ΟΥΡΑ[100], X, Y

ΧΑΡΑΚΤΗΡΕΣ: λειτουργία

ΑΡΧΗ

front $\leftarrow$ 0

rear $\leftarrow$ 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ λειτουργία

ΑΝ λειτουργία = 'ΕΣ' ΤΟΤΕ

ΔΙΑΒΑΣΕ X

ΑΝ rear = 100 ΤΟΤΕ

ΓΡΑΨΕ 'Ουρά γεμάτη-Δεν μπορεί να γίνει εισαγωγή στοιχείου'

ΑΛΛΙΩΣ_ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ

front $\leftarrow$ 1

rear $\leftarrow$ 1

ΟΥΡΑ[rear] $\leftarrow$ X

ΑΛΛΙΩΣ

rear $\leftarrow$ rear + 1

ΟΥΡΑ[rear] $\leftarrow$ X

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ

ΓΡΑΨΕ 'Ουρά άδεια-Δεν μπορεί να γίνει εξαγωγή στοιχείου'

ΑΛΛΙΩΣ

ΓΡΑΨΕ ΟΥΡΑ[front]

front $\leftarrow$ front + 1

! αρχικά η ουρά είναι άδεια οπότε οι δύο δείκτες

! front και rear έχουν την τιμή 0

! αν πρέπει να εκτελεστεί η εισαγωγή

! διαβάζουμε τον αριθμό που θα μπει στην ουρά

! και εκτελούμε τη λειτουργία της εισαγωγής

! όπως στη μεθοδολογία 29.1

! αν πρέπει να εκτελεστεί η εξαγωγή

! εκτελούμε τη λειτουργία της εξαγωγής

! όπως στη μεθοδολογία 29.2


```

ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ rear = 100 'Η front > rear
ΑΝ rear = 100 ΤΟΤΕ
  ΓΡΑΨΕ 'ΟΥΡΑ ΓΕΜΑΤΗ'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'ΑΔΕΙΑ ΟΥΡΑ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

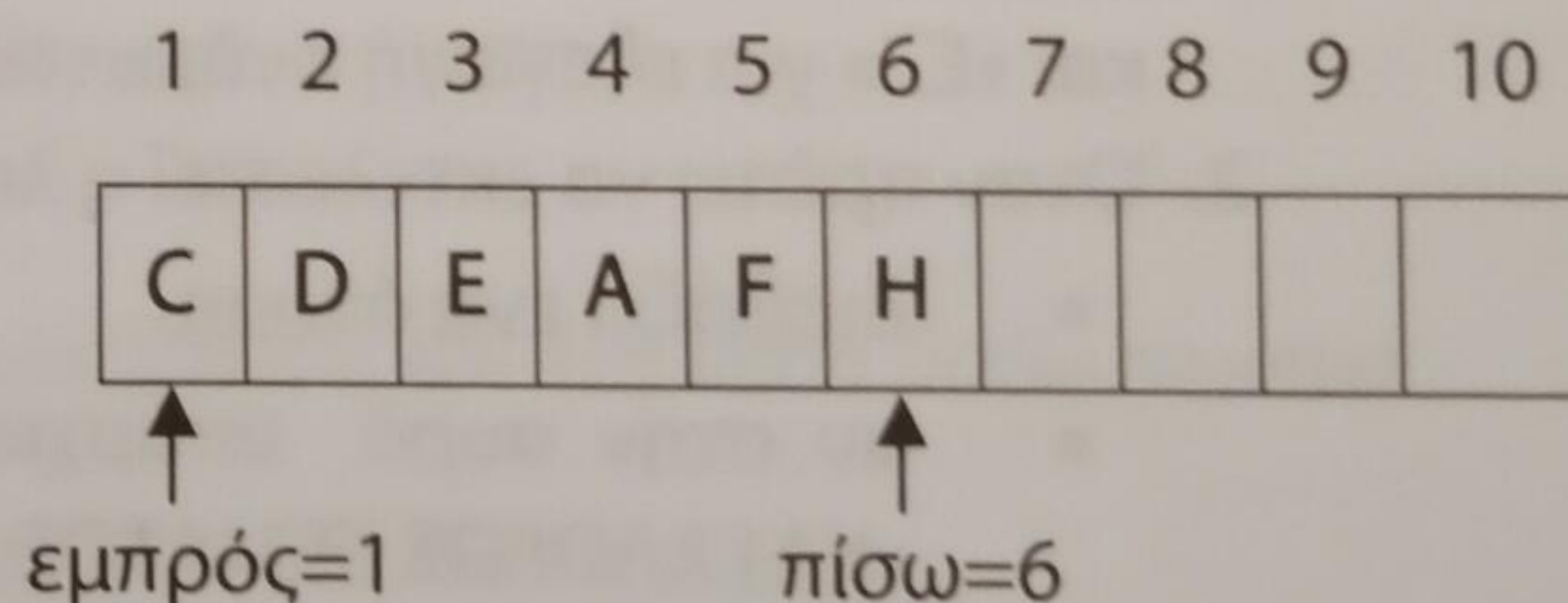
! η περίπτωση $front > rear$

ΑΣΚΗΣΕΙΣ ΛΥΜΕΝΕΣ

1. Δίνεται η εξής ακολουθία γραμμάτων: C, D, E, A, F, H.
 - α. 1. Να τοποθετήσετε τα γράμματα σε μια ουρά 10 θέσεων και να σχεδιάσετε την ουρά.
 2. Ποιες τιμές θα έχουν οι δείκτες της ουράς;
 - β. 1. Να τοποθετήσετε τα γράμματα σε μια στοίβα 8 θέσεων και να σχεδιάσετε τη στοίβα.
 2. Ποια θα είναι η τιμή του δείκτη της στοίβας;
 - γ. 1. Να σχεδιάσετε την ουρά μετά τη διαδοχική εξαγωγή δύο στοιχείων. Ποια στοιχεία θα εξαχθούν;
 2. Να σχεδιάσετε τη στοίβα, μετά την εισαγωγή του στοιχείου B.
 - δ. Πόσες φορές πρέπει να εκτελεστεί η λειτουργία της απώθησης, για να εξαχθεί το γράμμα A από τη στοίβα, και πόσες φορές πρέπει να εκτελεστεί η λειτουργία της εξαγωγής, για να εξαχθεί το γράμμα F από την ουρά;
 - ε. Αν εφαρμόσετε τις λειτουργίες εισαγωγή K, εισαγωγή L, εξαγωγή, εξαγωγή, ποιες είναι οι τιμές των δεικτών της και ποια η τελική μορφή της;

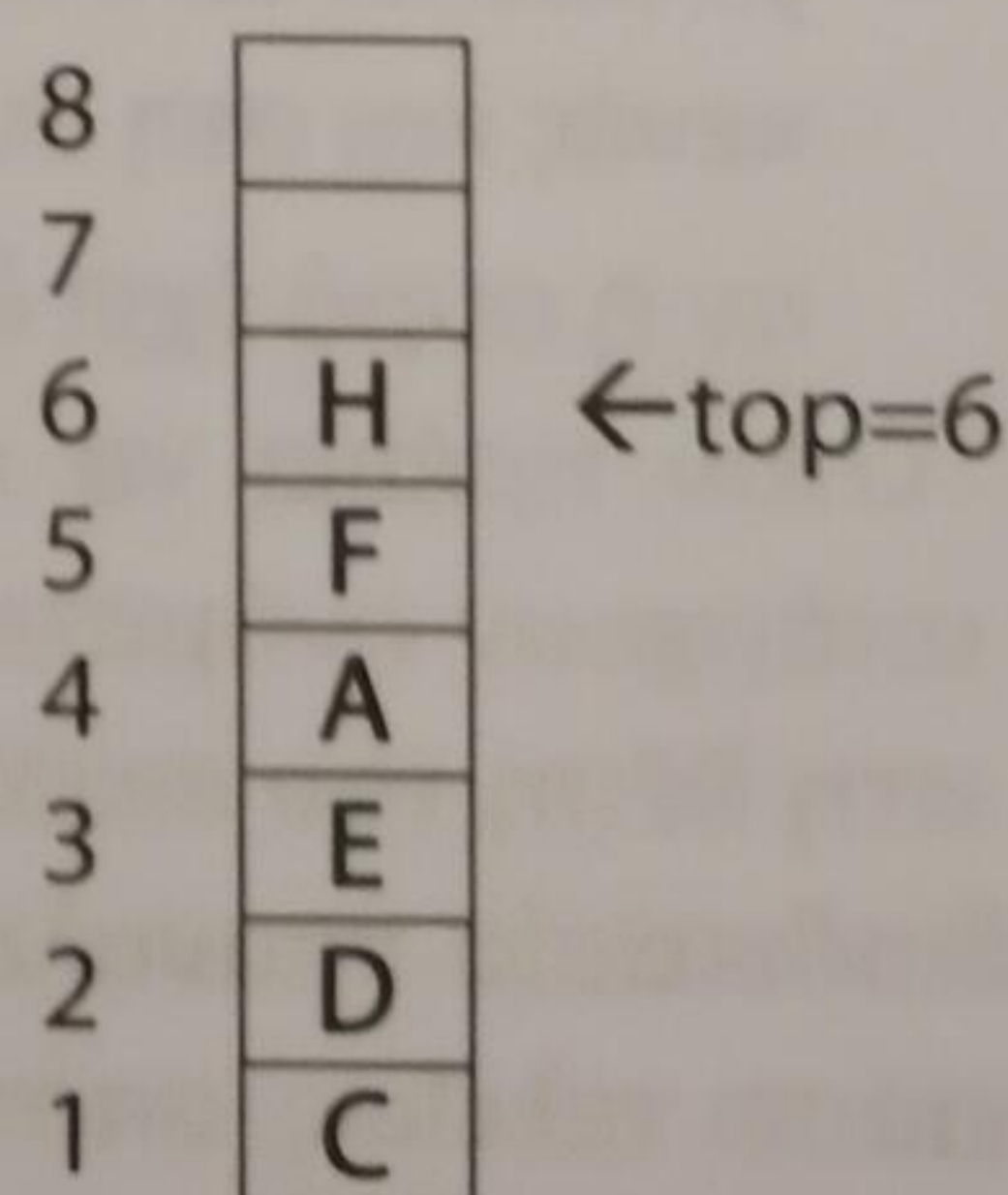
Λύση:

- α. 1. Η κατάσταση της ουράς, μετά την τοποθέτηση των στοιχείων, φαίνεται δίπλα.

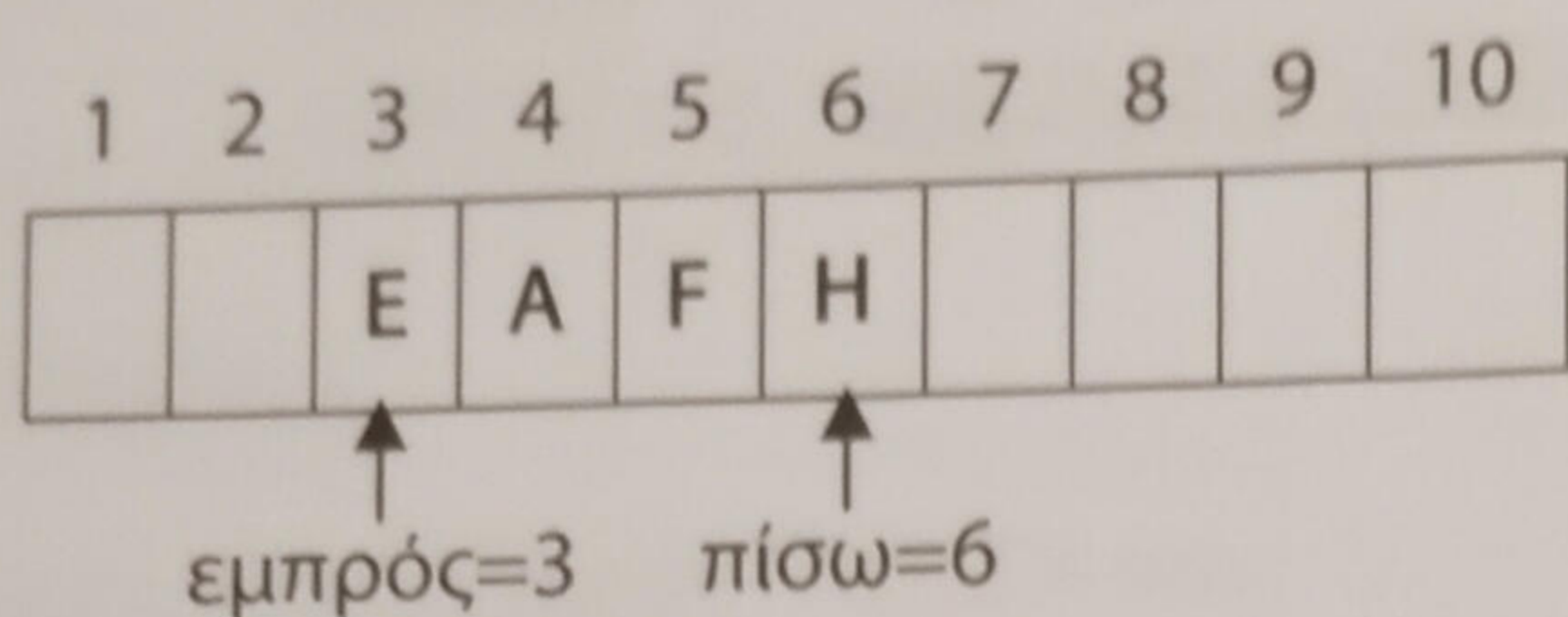


2. Ο δείκτης εμπρός (front) έχει την τιμή 1 και ο δείκτης πίσω (rear) την τιμή 6.

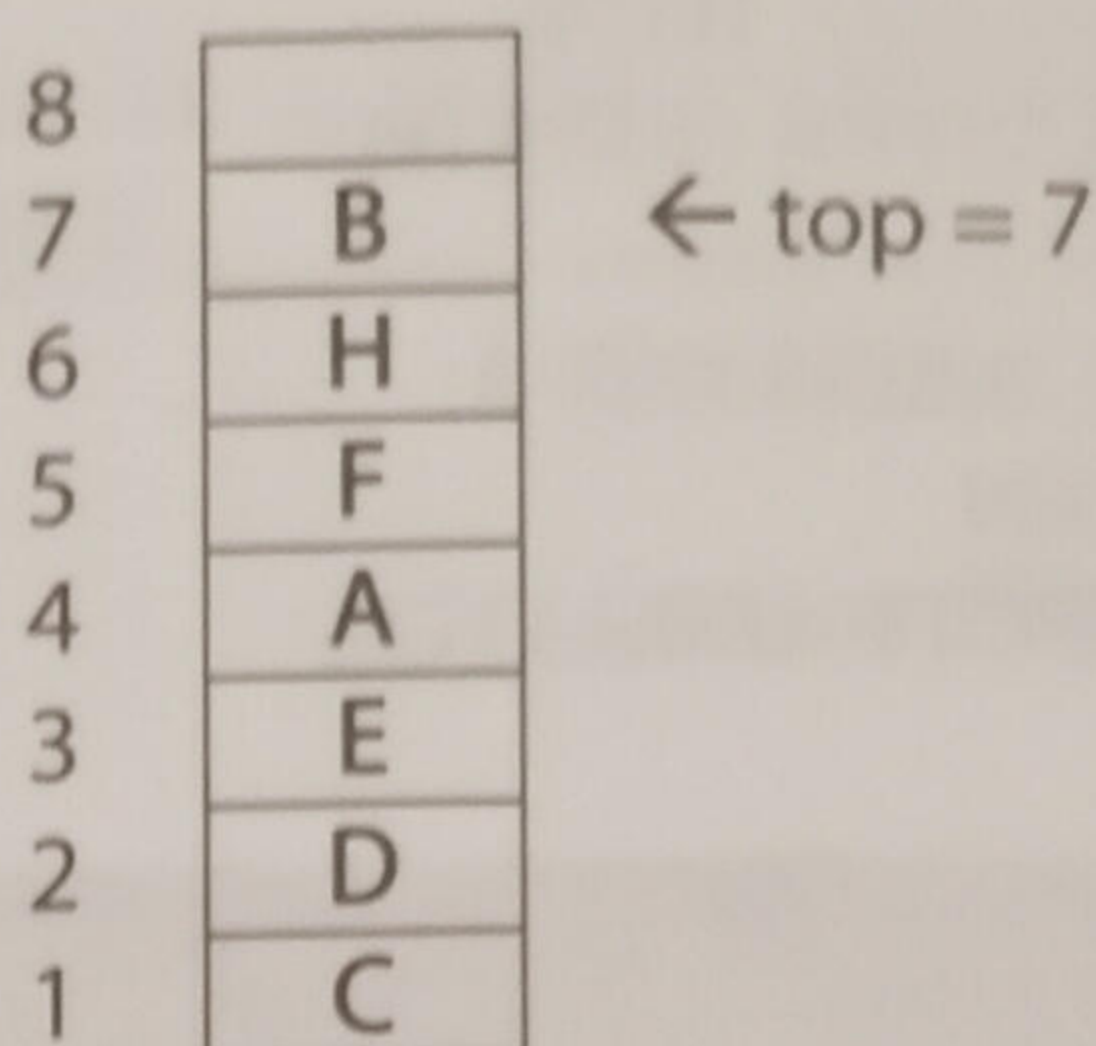
- β. 1. Η κατάσταση της στοίβας, μετά την τοποθέτηση των στοιχείων, φαίνεται δίπλα.
2. Ο δείκτης top έχει την τιμή 6 και δείχνει την κορυφή της στοίβας.



- γ. 1. Τα στοιχεία που θα εξαχθούν είναι το γράμμα C και το γράμμα D. Η ουρά, μετά την εξαγωγή, έχει την εξής μορφή:

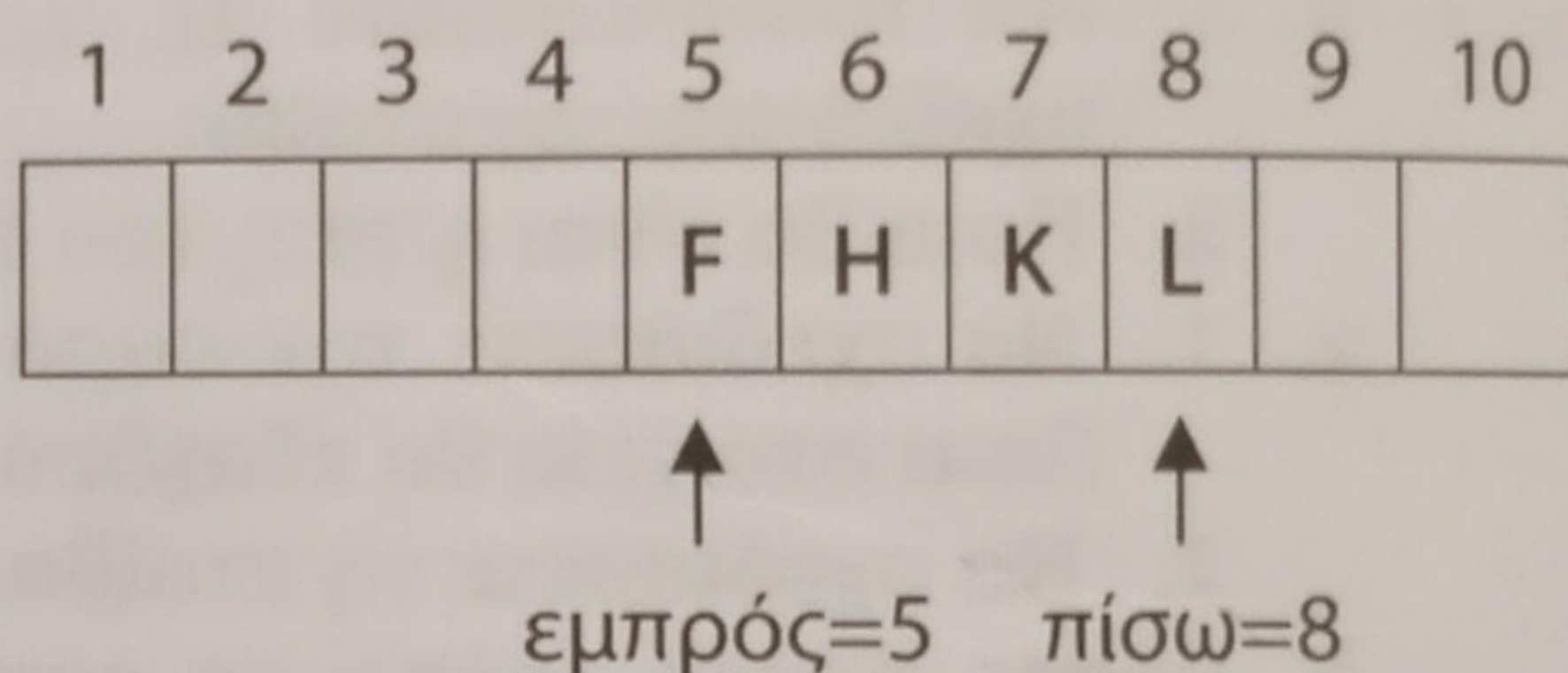


2. Η στοίβα, μετά την εισαγωγή του στοιχείου B, έχει την παρακάτω μορφή:



- δ. Για να εξέλθει το στοιχείο A από τη στοίβα, πρέπει να εκτελεστεί 4 φορές η λειτουργία της απώθησης. Για να εξέλθει το στοιχείο F από την ουρά, πρέπει να εκτελεστεί η λειτουργία της εξαγωγής 3 φορές.

- ε. Στο πίσω μέρος της ουρά θα γίνουν δύο εισαγωγές, των στοιχείων K και Λ, ενώ από εμπροστά θα γίνουν δύο εξαγωγές και θα αφαιρεθούν τα στοιχεία E και A. Η κατάσταση της ουράς, μετά την εκτέλεση των λειτουργιών, φαίνεται δίπλα.



2. Να γραφεί κύριο πρόγραμμα σε ΓΛΩΣΣΑ το οποίο διαχειρίζεται έναν κατάλογο ονομάτων με μορφή μιας δομής δεδομένων με τον παρακάτω τρόπο:

α. Χρησιμοποιεί τον πίνακα ON[200] για την υλοποίηση ουράς 200 θέσεων.

β. Εκτελεί τη λειτουργία της εισαγωγής και της εξαγωγής ονομάτων ως εξής:

1. Διαβάζει την επιλογή της λειτουργίας που θα εκτελεστεί («ΕΣ» για εισαγωγή και «ΕΞ» για εξαγωγή ονόματος).

2. Όταν πρέπει να εκτελεστεί η λειτουργία της εισαγωγής:

- διαβάζει ένα όνομα,
- αν στην ουρά υπάρχουν ήδη 200 ονόματα εμφανίζει το μήνυμα «ΚΑΤΑΛΟΓΟΣ ΓΕΜΑΤΟΣ»,
- αν η ουρά έχει διαθέσιμο χώρο μπροστά, αλλά είναι γεμάτη πίσω, μετακινεί τα στοιχεία της προς τα μπροστά τόσες θέσεις όσες είναι οι κενές και στη συνέχεια τοποθετεί το όνομα στην ουρά,
- αν η ουρά έχει διαθέσιμο χώρο πίσω τοποθετεί το όνομα στην ουρά.

3. Όταν πρέπει να εκτελεστεί η λειτουργία της εξαγωγής, εξετάζει αν υπάρχουν ονόματα στην ουρά, οπότε και αφαιρεί αυτό που πρέπει και στη θέση του εισάγει το κενό.

Η διαδικασία επαναλαμβάνεται, μέχρι ο κατάλογος να γεμίσει ή μέχρι να αδειάσει τελείως, οπότε και εμφανίζει το μήνυμα «ΑΔΕΙΟΣ ΚΑΤΑΛΟΓΟΣ».

Λύση:

```
ΑΡΧΗ
front ← 0
rear ← 0
flag ← Ψευδής
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ λειτουργία
  ΑΝ λειτουργία = 'ΕΣ' ΤΟΤΕ
    ΔΙΑΒΑΣΕ όνομα
    ΑΝ front = 1 ΚΑΙ rear = 200 ΤΟΤΕ
      ΓΡΑΨΕ 'ΚΑΤΑΛΟΓΟΣ ΓΕΜΑΤΟΣ'
      flag ← Αληθής
    ΑΛΛΙΩΣ_ΑΝ front > 1 ΚΑΙ rear = 200 ΤΟΤΕ
      υπάρχουν_ουρά ← (rear - front) + 1
      ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ υπάρχουν_ουρά
        ΟΥΡΑ[i] ← ΟΥΡΑ[front - 1 + i]
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
      front ← 1
      rear ← υπάρχουν_ουρά
      rear ← rear + 1
      ΟΥΡΑ[rear] ← όνομα
    ΑΛΛΙΩΣ_ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
      front ← 1
      rear ← 1
      ΟΥΡΑ[rear] ← όνομα
    ΑΛΛΙΩΣ
      rear ← rear + 1
      ΟΥΡΑ[rear] ← όνομα
  ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
  ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ουρά άδεια-Δεν μπορεί να γίνει εξαγωγή στοιχείου'
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ ΟΥΡΑ[front]
    ΟΥΡΑ[front] ← ''
    front ← front + 1
    ΑΝ front > rear ΤΟΤΕ
      ΓΡΑΨΕ 'ΑΔΕΙΟΣ ΚΑΤΑΛΟΓΟΣ'
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ flag = Αληθής 'Η front > rear
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

*! αρχικά η ουρά είναι άδεια οπότε οι δύο δείκτες
! front και rear έχουν την τιμή 0
! μεταβλητή για να ελέγχουμε αν η ουρά γέμισε πλήρως*

*! αν πρέπει να εκτελεστεί η εισαγωγή
! διαβάζουμε το όνομα που θα μπει στην ουρά
! έλεγχος αν η ουρά γέμισε πλήρως*

! η ουρά γέμισε πλήρως

*! αν υπάρχουν μπροστά θέσεις
! ολισθαίνουμε τα στοιχεία προς
! τα μπροστά και δημιουργείται
! χώρος στο πίσω μέρος
! μετά την ολίσθηση αλλάζουμε τους δείκτες
! ο δείκτης εμπρός έχει τιμή 1
! ο πίσω, το νέο τελευταίο στοιχείο
! αυξάνουμε τον δείκτη
! τοποθετούμε το όνομα στην ουρά*

! εκτελούμε τη λειτουργία της εισαγωγής

! αν πρέπει να εκτελεστεί η εξαγωγή

*! κάνω εξαγωγή και στη θέση βάζω το κενό
! έλεγχος αν άδειασε η ουρά*

3. Η περιήγηση επισκεπτών σε ένα σπήλαιο πραγματοποιείται σε ομάδα των 20 ατόμων τα οποία εισέρχονται σε αυτό από ένα συγκεκριμένο σημείο εισόδου και εξέρχονται από κάποιο άλλο σημείο εξόδου. Λόγω στενότητας χώρου υπάρχει ένα σύστημα εισόδου με το οποίο τα άτομα εισέρχονται στο σπήλαιο ο ένας μετά τον άλλον σχηματίζοντας μια σειρά και βαδίζοντας με αυτόν τον τρόπο σε όλη τη διαδρομή, μέχρι την έξοδο. Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ το οποίο διαχειρίζεται την περιήγηση ως εξής:
- Χρησιμοποιεί τον πίνακα ΕΠΙΣ[20] για να προσομοιώνει τη σειρά εισόδου των επισκεπτών με μία ουρά δεδομένων.
 - Διαβάζει επαναληπτικά τον κωδικό εισιτηρίου (αποτελείται από 3 γράμματα και 2 αριθμούς) του κάθε επισκέπτη της ομάδας επισκεπτών που εισέρχεται στο σπήλαιο και τον καταχωρίζει στην ουρά ΕΠΙΣ μέχρι αυτή να γεμίσει εμφανίζοντας το μήνυμα «ΣΠΗΛΑΙΟ ΠΛΗΡΕΣ».
 - Εμφανίζει:
 - τον κωδικό του εισιτηρίου του πρώτου επισκέπτη που θα βγει από το σπήλαιο,
 - τον κωδικό του εισιτηρίου του τελευταίου επισκέπτη που θα βγει από το σπήλαιο.

Λύση:

```

ΠΡΟΓΡΑΜΜΑ Σπήλαιο
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: i, front, rear
  ΧΑΡΑΚΤΗΡΕΣ: ΕΠΙΣ[20], κωδικός  ! α ερώτημα
ΑΡΧΗ
  front ← 0
  rear ← 0
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ κωδικός
    ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ  ! β ερώτημα – μεθοδολογία 29.1
      front ← 1
      rear ← 1
      ΕΠΙΣ[rear] ← κωδικός
      ΑΛΛΙΩΣ_ΑΝ rear < 20 ΤΟΤΕ  ! λειτουργία της εισαγωγής
        rear ← rear + 1
        ΕΠΙΣ[rear] ← κωδικός
      ΤΕΛΟΣ_ΑΝ
      ΑΝ rear = 20 ΤΟΤΕ
        ΓΡΑΨΕ 'ΣΠΗΛΑΙΟ ΠΛΗΡΕΣ'
        ΤΕΛΟΣ_ΑΝ
      ΜΕΧΡΙΣ_ΟΤΟΥ rear = 20
      ΓΡΑΨΕ ΕΠΙΣ[front]
      front ← front + 1
      ! γ1 ερώτημα – εμφάνιση πρώτου επισκέπτη
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΑΝ front = 20 ΤΟΤΕ  ! λειτουργία της εξαγωγής -- μεθοδολογία 29.2
        ΓΡΑΨΕ ΕΠΙΣ[front]
        ΤΕΛΟΣ_ΑΝ
        front ← front + 1
        ! γ2 ερώτημα – εμφάνιση τελευταίου επισκέπτη
      ΜΕΧΡΙΣ_ΟΤΟΥ front > rear
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
  
```


4. Μια υπάλληλος του γραφείου ΟΑΕΔ εξυπηρετεί ανέργους με σειρά προτεραιότητας. Ο κάθε άνεργος προμηθεύεται έναν αριθμό που εκδίδεται από ειδικό μηχάνημα σχηματίζοντας έτσι μία σειρά προτεραιότητας. Ο αριθμός προτεραιότητας αρχίζει από το νούμερο 101 και αυξάνεται για κάθε άνεργο που επιθυμεί να εξυπηρετηθεί από την υπάλληλο. Η έκδοση αριθμών προτεραιότητας διακόπτεται από το μηχάνημα, όταν φτάσει στον αριθμό 150. Με δεδομένο ότι στην ουρά προτεραιότητας υπάρχουν ήδη 15 άνεργοι, να γραφεί κύριο πρόγραμμα σε ΓΛΩΣΣΑ το οποίο διαχειρίζεται την εξυπηρέτηση των ανέργων ως εξής:

α. Χρησιμοποιεί τον πίνακα KIN[50] για να προσομοιώνει την ουρά εξυπηρέτησης των ανέργων, όπου στις 15 πρώτες θέσεις υπάρχουν οι αριθμοί των 15 πρώτων ανέργων.

β. Διαβάζει επαναληπτικά το γράμμα «N», που καθορίζει την είσοδο νέου ανέργου, ή το γράμμα «E», που δηλώνει ότι ένας άνεργος θα εξυπηρετηθεί, και πραγματοποιεί τις παρακάτω ενέργειες:

- κατά την είσοδο ενός νέου ανέργου, εξετάζει αν το μηχάνημα μπορεί να εκδώσει αριθμό προτεραιότητας, τον οποίο δημιουργεί και καταχωρίζει στην ουρά με τη λειτουργία της «εισαγωγής», διαφορετικά, εμφανίζει το μήνυμα «Δεν μπορείτε να λάβετε αριθμό προτεραιότητας»,
- όταν ένας άνεργος πρόκειται να εξυπηρετηθεί, εξετάζεται αν υπάρχουν τέτοιοι άνεργοι και στη συνέχεια, με τη λειτουργία της «εξαγωγής» αφαιρείται από την ουρά ο αριθμός προτεραιότητάς του τοποθετώντας στη θέση του τον αριθμό μηδέν, εμφανίζεται το μήνυμα «Εξυπηρέτηση» και δίπλα ο αριθμός προτεραιότητας.

Η επαναληπτική διαδικασία ολοκληρώνεται, όταν εξυπηρετηθούν όλοι οι άνεργοι.

γ. Εμφανίζει:

1. τον συνολικό αριθμό ανέργων που εξυπηρετήθηκαν,
2. πόσες φορές η ουρά είχε έναν μόνο άνεργο.

Λύση:

ΑΡΧΗ

πλήθος_ανέργων $\leftarrow 0$

πλήθος_ένα $\leftarrow 0$

αριθμός $\leftarrow 115$

front $\leftarrow 1$

! ο front έχει την τιμή 1, (αρχή ουράς)

rear $\leftarrow 15$

! ο rear έχει την τιμή 15, (πίσω ουράς)

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ενέργεια

ΑΝ ενέργεια = 'N' ΤΟΤΕ

! αν πρέπει να γίνει εισαγωγή νέου ανέργου

ΑΝ rear = 50 ΤΟΤΕ

ΓΡΑΨΕ 'Δεν μπορείτε να λάβετε αριθμό προτεραιότητας'

ΑΛΛΙΩΣ

! εξετάζουμε αν υπάρχει χώρος

αριθμός $\leftarrow$ αριθμός + 1

! δημιουργούμε τον αριθμό προτεραιότητας

rear $\leftarrow$ rear + 1

! αυξάνουμε τη μεταβλητή rear,

KIN[rear] $\leftarrow$ αριθμός

! και εισάγουμε τον αριθμό προτεραιότητας

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

! αν πρέπει να εξυπηρετηθεί ένας άνεργος

ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ

ΓΡΑΨΕ 'Ουρά άδεια-Δεν υπάρχουν άνεργοί'

ΑΛΛΙΩΣ

Υ ← ΚΙΝ[front]

! τον εξυπηρετούμε (τον βγάζουμε από την ουρά)

ΚΙΝ[front] ← 0

! στη θέση του βάζουμε το μηδέν

ΓΡΑΨΕ 'Εξυπηρέτηση', Υ

πλήθος_άνέργων ← πλήθος_άνέργων + 1 *! μετράμε όσους εξυπηρετήθηκαν*

ΑΝ front = rear ΤΟΤΕ *! όταν υπάρχει ένας μόνο άνεργος*

πλήθος_ένα ← πλήθος_ένα + 1 *! μετράμε πόσες φορές υπήρχε ένας άνεργος*

ΤΕΛΟΣ_ΑΝ

front ← front + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ front > rear

ΓΡΑΨΕ πλήθος_άνέργων, πλήθος_ένα

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

5. Σε ένα χυτήριο χρυσού, ο λιωμένος χρυσός αρχικά μετατρέπεται σε ράβδους χρυσού και συσκευάζεται. Στη συνέχεια φορτώνεται σε ένα όχημα το οποίο διαθέτει 50 μικρά βαγόνια όπου μέσα στο καθένα τοποθετείται μια συσκευασία ράβδων χρυσού με τη σειρά, αρχίζοντας από το πρώτο βαγόνι και τελειώνοντας στο τελευταίο. Το όχημα ξεφορτώνει τις συσκευασίες στον μυστικό προορισμό του. Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα διαχειρίζεται τη φόρτωση και την εκφόρτωση συσκευασιών ράβδων χρυσού στο όχημα μεταφοράς με τον παρακάτω τρόπο:

α. Εμφανίζει ένα μενού με τις επιλογές:

1. Φόρτωση

2. Εκφόρτωση

3. Έξοδος

β. Υλοποιεί το μενού επιλογών ως εξής:

Αν επιλεχθεί η φόρτωση, διαβάζει την απάντηση του χρήστη (ΝΑΙ ή ΟΧΙ) για το εάν υπάρχει συσκευασία για φόρτωση ή όχι, εμφανίζοντας κατάλληλο μήνυμα. Αν υπάρχει, διαβάζει το βάρος της συσκευασίας και την καταχωρεί στην ουρά ΧΡΥΣΟΣ. Η διαδικασία φόρτωση επαναλαμβάνεται μέχρι να μην υπάρχει άλλη συσκευασία για φόρτωση ή μέχρι το όχημα μεταφοράς να γεμίσει οπότε εμφανίζει το μήνυμα «Το όχημα είναι πλήρες».

Αν επιλεχθεί η εκφόρτωση και εφόσον υπάρχουν συσκευασίες φορτωμένες σε αυτό, ξεφορτώνει το όχημα εμφανίζοντας το βάρος της κάθε συσκευασίας με τη σειρά που φορτώθηκαν από το χυτήριο, το συνολικό βάρος τους, καθώς και το πλήθος αυτών. Διαφορετικά εμφανίζει το μήνυμα «Το όχημα είναι άδειο».

Αν επιλεχθεί η έξοδος το πρόγραμμα τερματίζει, διαφορετικά επαναλαμβάνει τα βήματα α και β.

Λύση:

ΠΡΟΓΡΑΜΜΑ Χυτήριο_χρυσού

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: εμπρός, πίσω, σύνολο, επιλογή, πλήθος

ΠΡΑΓΜΑΤΙΚΕΣ: ΧΡΥΣΟΣ[50], βάρος, σύνολο

ΧΑΡΑΚΤΗΡΕΣ: απάντηση

ΑΡΧΗ

εμπρός $\leftarrow$ 0

πίσω $\leftarrow$ 0

πλήθος $\leftarrow$ 0

σύνολο $\leftarrow$ 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ '1. Φόρτωση'

! εμφάνιση μενού επιλογών

ΓΡΑΨΕ '2. Εκφόρτωση'

ΓΡΑΨΕ '3. Έξοδος'

ΔΙΑΒΑΣΕ επιλογή

ΑΝ επιλογή = 1 ΤΟΤΕ

! αν επιλεγθεί η φόρτωση

ΑΝ πίσω = 50 ΤΟΤΕ

ΓΡΑΨΕ 'Το όχημα είναι πλήρες'

ΑΛΛΙΩΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

! διαβάζεται επαναληπτικά για κάθε συσκευασία

ΓΡΑΨΕ 'Θέλεις να φορτώσεις συσκευασία;'

ΔΙΑΒΑΣΕ απάντηση

ΑΝ απάντηση = 'ΝΑΙ' ΤΟΤΕ

! αν υπάρχει συσκευασία για φόρτωση

ΔΙΑΒΑΣΕ βάρος

! εκτελούμε τη λειτουργία της εισαγωγής

ΑΝ εμπρός = 0 ΚΑΙ πίσω = 0 ΤΟΤΕ

εμπρός $\leftarrow$ 1

πίσω $\leftarrow$ 1

ΧΡΥΣΟΣ[πίσω] $\leftarrow$ βάρος

ΑΛΛΙΩΣ

πίσω $\leftarrow$ πίσω + 1

ΧΡΥΣΟΣ[πίσω] $\leftarrow$ βάρος

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ απάντηση = 'ΟΧΙ' Ή πίσω = 50

ΑΝ πίσω = 50 ΤΟΤΕ

ΓΡΑΨΕ 'Το όχημα είναι πλήρες'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ_ΑΝ επιλογή = 2 ΤΟΤΕ

! αν επιλεγθεί η εκφόρτωση

ΑΝ εμπρός = 0 ΚΑΙ πίσω = 0 ΤΟΤΕ

! εκτελούμε τη λειτουργία της εξαγωγής

ΓΡΑΨΕ 'Το όχημα είναι άδειο'

ΑΛΛΙΩΣ_ΑΝ εμπρός = πίσω ΤΟΤΕ

ΓΡΑΨΕ ΧΡΥΣΟΣ[εμπρός]

πλήθος $\leftarrow$ πλήθος + 1

σύνολο $\leftarrow$ σύνολο + ΧΡΥΣΟΣ[εμπρός]

εμπρός $\leftarrow$ 0

πίσω $\leftarrow$ 0


```

ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ ΧΡΥΣΟΣ[εμπρός]
  σύνολο ← σύνολο + ΧΡΥΣΟΣ[εμπρός]
  εμπρός ← εμπρός + 1
  πλήθος ← πλήθος + 1
ΜΕΧΡΙΣ_ΟΤΟΥ front > rear
  εμπρός ← 0
  πίσω ← 0
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επιλογή = 3
  ΓΡΑΨΕ πλήθος, σύνολο
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

6. Σε μια ιδιωτική τράπεζα λειτουργούν δύο ταμεία το 1 και το 2 τα οποία μπορούν να εξυπηρετήσουν συνολικά μέχρι 500 πελάτες την ημέρα. Όταν ένας πελάτης εισέρχεται στην τράπεζα λαμβάνει έναν αριθμό σειράς και εξυπηρετείται σε ένα από τα δύο ταμεία, ενώ ο διευθυντής της τράπεζας μπορεί να διακόψει τη διαδικασία εξυπηρέτησης και να βλέπει κάθε φορά πόσοι πελάτες περιμένουν και πόσοι έχουν εξυπηρετηθεί. Να γραφεί κύριο πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

α. Δημιουργεί την κατάλληλη δομή δεδομένων για να διαχειρίζεται το σύστημα της τράπεζας.

β. Διαβάζει επαναληπτικά, το γράμμα «Π», για είσοδο πελάτη, ή το γράμμα «Τ», για εξυπηρέτηση ενός πελάτη ή το γράμμα «Δ» για τις ενέργειες του διευθυντή και πραγματοποιεί τις παρακάτω λειτουργίες:

- κατά την είσοδο ενός πελάτη, αν μπορεί η τράπεζα να τον εξυπηρετήσει, διαβάζει τον αριθμό σειράς του και τον καταχωρίζει στη δομή δεδομένων, διαφορετικά, εμφανίζει το μήνυμα «Δεν μπορούμε να σας εξυπηρετήσουμε»,
- για την εξυπηρέτηση ενός πελάτη, αν υπάρχουν πελάτες, διαβάζει τον αριθμό του ταμείου που εξυπηρετείται, εμφανίζει τον αριθμό σειράς του πελάτη και τον αφαιρεί από τη δομή δεδομένων, διαφορετικά εμφανίζει το μήνυμα «Δεν υπάρχουν πελάτες»,
- για τις ενέργειες του διευθυντή, διακόπτει την επαναληπτική διαδικασία και εμφανίζει τον αριθμό του ταμείου που εξυπηρέτησε τους περισσότερους πελάτες, το πλήθος των πελατών που περιμένουν καθώς και πόσοι έχουν εξυπηρετηθεί.

Λύση:

```

ΑΡΧΗ
  front ← 0
  rear ← 0
  πλ1 ← 0
  πλ2 ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ ενέργεια

```


ΑΝ ενέργεια = 'Π' ΤΟΤΕ *! αν πρέπει να γίνει εισαγωγή νέου πελάτη*

ΔΙΑΒΑΣΕ αριθμός *! διαβάζει τον αριθμό προτεραιότητας*

ΑΝ rear = 500 ΤΟΤΕ

ΓΡΑΨΕ 'Δεν μπορούμε να σας εξυπηρετήσουμε

ΑΛΛΙΩΣ_ ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ

front $\leftarrow$ 1

rear $\leftarrow$ 1

TP[rear] $\leftarrow$ αριθμός

ΑΛΛΙΩΣ

rear $\leftarrow$ rear + 1

TP[rear] $\leftarrow$ αριθμός

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ_ΑΝ ενέργεια = 'Τ' ΤΟΤΕ

! αν πρέπει να γίνει εξυπηρέτηση πελάτη

ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ

ΓΡΑΨΕ 'Δεν υπάρχουν πελάτες'

ΑΛΛΙΩΣ

ΔΙΑΒΑΣΕ ταμείο

ΓΡΑΨΕ 'Εξυπηρέτηση', TP[front] *! τον εξυπηρετούμε (τον βγάζουμε από την ουρά)*

ΕΠΙΛΕΞΕ ταμείο

ΠΕΡΙΠΤΩΣΗ 1

πλ1 $\leftarrow$ πλ1 + 1

ΠΕΡΙΠΤΩΣΗ 2

πλ2 $\leftarrow$ πλ2 + 1

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

front $\leftarrow$ front + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ front > rear ΤΟΤΕ

front $\leftarrow$ 0

rear $\leftarrow$ 0

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ ενέργεια = 'Δ'

ΑΝ πλ1 > πλ2 ΤΟΤΕ

ΓΡΑΨΕ 'Ταμείο 1'

ΑΛΛΙΩΣ_ΑΝ πλ1 < πλ2 ΤΟΤΕ

ΓΡΑΨΕ 'Ταμείο 2'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Ίδιο αριθμό πελατών εξυπηρέτησαν

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Εξυπηρετήθηκαν:', πλ1+πλ2, 'Περιμένουν:', (rear - front) + 1

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΟΥ

29.2. Συμπληρώστε τα κενά των παρακάτω προτάσεων:

1. Η δομή δεδομένων ουρά χρησιμοποιεί τη μέθοδο επεξεργασίας _____ μέσα, _____ έξω.
2. Η λειτουργία της εισαγωγής εφαρμόζεται στη δομή δεδομένων _____.
3. Όταν η ουρά είναι άδεια δεν μπορούμε να εκτελέσουμε τη λειτουργία της _____.

29.3. Δίνεται ο διπλανός πίνακας A[9] που υλοποιεί μια ουρά δεδομένων.

Στις περιπτώσεις β, γ και δ εκτελείται μια λειτουργία κάθε φορά. Να συμπληρώσετε την τιμή των δεικτών «εμπρός» και «πίσω» της ουράς και το όνομα της λειτουργίας που εκτελείται.

- α. εμπρός = __ (1) __ πίσω = __ (2) __
- β. εκτέλεση λειτουργίας: __ (3) __
εμπρός = __ (4) __ πίσω = __ (5) __
- γ. εκτέλεση λειτουργίας: __ (6) __
εμπρός = __ (7) __ πίσω = __ (8) __
- δ. εκτέλεση λειτουργίας: __ (9) __
εμπρός = __ (10) __ πίσω = __ (11) __

α.	1	2	3	4	5	6	7	8	9

β.	1	2	3	4	5	6	7	8	9
	3								

γ.	1	2	3	4	5	6	7	8	9
	3	4							

δ.	1	2	3	4	5	6	7	8	9
		4							

29.4. Δίνεται ο διπλανός πίνακας A[10] που υλοποιεί μία ουρά γραμμάτων. Επίσης δίνεται και το διπλανό τμήμα προγράμματος που διαχειρίζεται τη διπλανή ουρά δεδομένων.

Να γράψετε τους αριθμούς (1) έως (8) που αντιστοιχούν στα κενά του διπλανού τμήματος προγράμματος και δίπλα σε κάθε αριθμό ό,τι πρέπει να συμπληρωθεί, ώστε να εκτελεί συνεχώς τη λειτουργία της εξαγωγής, μέχρι να αδειάσει η ουρά.

1	2	3	4	5	6	7	8	9	10
B	Φ	A	Z	K	Ψ				

```

front ← __ (1) __
rear ← __ (2) __
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Υ ← A [ __ (3) __ ]
    ΓΡΑΨΕ Υ
    __ (4) __ ← __ (5) __ + 1
ΜΕΧΡΙΣ_ΟΤΟΥ front __ (6) __ rear
front ← __ (7) __
rear ← __ (8) __
    
```


ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

29.5. Επιλέξτε τη σωστή απάντηση βάζοντας σε κύκλο το κατάλληλο γράμμα:

1. Η εισαγωγή ενός στοιχείου στην ουρά γίνεται:
 - α. στο εμπρός άκρο της.
 - β. στο μέσον της.
 - γ. στο πίσω άκρο της.
2. Όταν μία «ουρά» είναι κενή μπορούμε να εκτελέσουμε:
 - α. μόνο τη λειτουργία της εισαγωγής.
 - β. πρώτα τη λειτουργία της εξαγωγής και έπειτα της εισαγωγής.
 - γ. μόνο τη λειτουργία της εξαγωγής.
3. Όταν οι δείκτες εμπρός (front) και πίσω (rear) έχουν την τιμή 3, τότε η ουρά:
 - α. έχει 2 στοιχεία.
 - β. είναι κενή.
 - γ. έχει 3 στοιχεία.
 - δ. έχει 1 στοιχείο.
4. Όταν μια «ουρά» έχει ένα δεδομένο, μεταξύ των δεικτών front και rear ισχύει:
 - α. η τιμή του δείκτη front είναι μεγαλύτερη από αυτή του δείκτη rear.
 - β. η τιμή του δείκτη front είναι μικρότερη από αυτή του δείκτη rear.
 - γ. η τιμή του δείκτη front είναι ίση με αυτή του δείκτη rear.
5. Σε μία ουρά 10 θέσεων όταν $\text{rear} > \text{front}$, τότε η ουρά:
 - α. έχει 10 στοιχεία.
 - β. έχει ένα στοιχείο.
 - γ. έχει μηδέν στοιχεία.
 - δ. δεν είναι άδεια.

29.6. Σε μία ουρά 20 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία C, D και F στην 1η, 2η και 3η θέση. Βάλτε σε κύκλο το γράμμα που χρειάζεται για να δείξετε ποιο θα είναι το περιεχόμενο της ουράς, μετά την εκτέλεση των λειτουργιών: εξαγωγή, εξαγωγή, εισαγωγή F, εισαγωγή D, εξαγωγή, εισαγωγή C.

- α. C, D και F στην 4η, 5η και 6η θέση.
- β. D, F και C στην 1η, 2η και 3η θέση.
- γ. F, D και C στην 4η, 5η και 6η θέση.
- δ. F, D και C στην 3η, 4η και 5η θέση.

29.7. Επιλέξτε τις σωστές απαντήσεις, βάζοντας σε κύκλο όσα γράμματα χρειάζονται ώστε, αν ο πίνακας $\text{P}[10]$ χρησιμοποιείται για να υλοποιήσουμε μία ουρά και οι μεταβλητές εμπρός και πίσω για να τη διαχειριστούμε, τότε:

- α. όταν $\text{empros} = 1$ και $\text{piso} = 10$, η ουρά είναι γεμάτη.
- β. η μεταβλητή εμπρός δεν δείχνει πάντα το πρώτο στοιχείο που εισήλθε στην ουρά.
- γ. η μεταβλητή πίσω δείχνει το πρώτο στοιχείο που θα εξέλθει από την ουρά.
- δ. όταν $\text{empros} = \text{piso} = 0$, μπορούμε να εξάγουμε το 1^ο στοιχείο.
- ε. όταν $\text{empros} = 2$ και $\text{piso} = 2$, μπορούμε να εξάγουμε το πολύ ένα στοιχείο.
- ζ. όταν $\text{piso} = 10$, μπορούμε να εισάγουμε ένα στοιχείο.

ΜΕΤΑΤΡΟΠΗΣ

29.8. Το διπλανό τμήμα εντολών εκτελεί τη λειτουργία της εισαγωγής στοιχείων σε μια ουρά 100 θέσεων. Να κάνετε τις απαραίτητες αλλαγές, ώστε το τμήμα προγράμματος:

- να πραγματοποιεί εισαγωγή στοιχείου στην ουρά μετά από την καταφατική απάντηση ενός χρήστη (ΝΑΙ ή ναι) και ύστερα από την εμφάνιση κατάλληλου μηνύματος,
- να τερματίζει την εισαγωγή όταν η ουρά γεμίσει ή όταν ο χρήστης δώσει αρνητική απάντηση στην εισαγωγή στοιχείων (ΟΧΙ ή όχι).

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
    front ← 1
    rear ← 1
    table[rear] ← Χ
ΑΛΛΙΩΣ_ΑΝ rear < 100 ΤΟΤΕ
    rear ← rear + 1
    table[rear] ← Χ
ΤΕΛΟΣ_ΑΝ
ΑΝ rear = 100 ΤΟΤΕ
    ΓΡΑΨΕ 'Ουρά γεμάτη'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ rear = 100
```

ΑΝΤΙΣΤΟΙΧΙΣΗΣ

29.9. Στον παρακάτω πίνακα η Στήλη Α περιέχει δομές δεδομένων και η Στήλη Β περιέχει λειτουργίες. Να γράψετε στο τετράδιό σας τους αριθμούς της Στήλης Α και δίπλα τα γράμματα της Στήλης Β που αντιστοιχούν σωστά. Ας σημειωθεί ότι σε κάποιες δομές δεδομένων μπορεί να αντιστοιχούν περισσότερες από μία λειτουργίες. (Ημερήσια 2002)

Στήλη Α	Στήλη Β
1. Ουρά	α. Απώθηση
2. Στοίβα	β. Εξαγωγή
	γ. Ώθηση
	δ. Εισαγωγή

29.10. Να γράψετε τους αριθμούς της στήλης Α και δίπλα το γράμμα της στήλης Β που αντιστοιχεί σωστά. (Ημερήσια 2013)

Στήλη Α	Στήλη Β
1. Χαρακτήρες	α. Λογική τιμή
2. Ελεύθερο κείμενο	β. Ουρά
3. Ώθηση	γ. Κριτήριο αλγορίθμου
4. Αληθής	δ. Επανάληψη
5. FIFO	ε. Τύπος μεταβλητής
6. Αποτελεσματικότητα	ζ. Στοίβα
7. Βρόχος	η. Τρόπος αναπαράστασης αλγορίθμου

29.11. Να γράψετε στο τετράδιό σας τους αριθμούς της Στήλης Α και δίπλα σε κάθε αριθμό ένα από τα γράμματα της Στήλης Β, ώστε να προκύπτει η σωστή αντιστοίχιση (στη Στήλη Β περισσεύουν δύο γράμματα). (Επαναληπτικές Εσπ .2005)

Στήλη Α	Στήλη Β
1. Ουρά	α. Δομή επιλογής
2. $x \leftarrow 1$ Όσο $x < 5$ επανάλαβε Εμφάνισε x $x \leftarrow x + 1$ Τέλος_επανάληψης	β. Δομή επανάληψης
3. Στοίβα	γ. FIFO
4. Επίλεξε Τέλος_επιλογών	δ. LIFO
5. ΚΑΙ	ε. Αριθμητικός τελεστής
	ζ. Λογικός τελεστής
	η. Συνάρτηση

29.12. Να γράψετε στο τετράδιό σας καθέναν από τους αριθμούς της Στήλης Α και δίπλα του ένα γράμμα της Στήλης Β, ώστε να προκύπτει η σωστή αντιστοίχιση. (Εσπερινά 2008)

Στήλη Α	Στήλη Β
1. Ουρά	α. Ώθηση
2. Λογικός τελεστής	β. ΑΛΗΘΗΣ
3. Στοίβα	γ. ΚΑΙ
4. Λογική σταθερά	δ. Δύο δείκτες

ΕΜΦΑΝΙΣΗΣ ΤΙΜΩΝ

29.13. Κατά την είσοδό τους σε μια τράπεζα οι πελάτες παίρνουν διαδοχικούς αριθμούς προτεραιότητας 1, 2, 3... που καθορίζουν τη σειρά τους στην ουρά του μοναδικού ταμείου. Κάθε 2 λεπτά της ώρας προσέρχεται ένας νέος πελάτης και προστίθεται στην ουρά. Ο ταμίας εξυπηρετεί κάθε φορά τον πρώτο πελάτη στην ουρά και η εξυπηρέτησή του διαρκεί 3 λεπτά ακριβώς. Μετά την εξυπηρέτησή του ο πελάτης αποχωρεί από την ουρά. Κατά την αρχή της διαδικασίας (χρόνος 0) στην ουρά υπάρχει μόνο ο πελάτης με αριθμό προτεραιότητας 1. Να γράψετε διαδοχικά, σε ξεχωριστές γραμμές, με τη σωστή σειρά, τους αριθμούς προτεραιότητας των πελατών που βρίσκονται στην ουρά του ταμείου αμέσως μετά το 1ο, 2ο, 3ο, 4ο, 5ο και 6ο λεπτό. (Ημερήσια 2016)

29.14. Στην ουρά Α, 20 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία Γ, Ε, Λ, Ο, Σ, στην 1η, 2η,...5η θέση. Να γράψετε τι θα εμφανίσει κατά την εκτέλεσή του, το διπλανό τμήμα προγράμματος.

```
front ← 1
rear ← 5
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ουρά άδεια'
ΑΛΛΙΩΣ_ΑΝ front = rear ΤΟΤΕ
    ΓΡΑΨΕ Α[front]
front ← 0
```



```

rear ← 0
ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ A [front]
  front ← front + 1
ΜΕΧΡΙΣ_ΟΤΟΥ front = 3
ΤΕΛΟΣ_ΑΝ

```

29.15. Το διπλανό κύριο πρόγραμμα διαχειρίζεται τη δομή δεδομένων ουρά με τη βοήθεια του πίνακα O[20]. Να σχεδιάσετε τη μορφή που θα έχει η δομή δεδομένων, μετά την εκτέλεση του διπλανού κύριου προγράμματος, αν ως είσοδος δοθούν οι παρακάτω τιμές:

N, ΑΘΑΝΑΣΙΟΣ, Ο, Ο, Ν, ΓΙΩΡΓΟΣ, Ο, Ν, ΠΕΤΡΟΣ

```

ΑΡΧΗ
front ← 0
rear ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ Υ
  ΑΝ Υ = 'Ν' ΤΟΤΕ
    ΔΙΑΒΑΣΕ Χ
    ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
      front ← 1
      rear ← 1
      O[rear] ← Χ
    ΑΛΛΙΩΣ_ΑΝ rear < 20 ΤΟΤΕ
      rear ← rear + 1
      O[rear] ← Χ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ front, rear
ΜΕΧΡΙΣ_ΟΤΟΥ rear = 3
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

29.16. Δίνεται ο παρακάτω πίνακας A[10] που υλοποιεί μία ουρά ακέραιων αριθμών:

1	2	3	4	5	6	7	8	9	10
5	8								

Δίνεται και το διπλανό τμήμα προγράμματος που διαχειρίζεται την παραπάνω ουρά δεδομένων.

α. Να γράψετε τί θα εμφανίσει κατά την εκτέλεσή του, αν ως είσοδος δοθούν οι παρακάτω τιμές:

1, 1, 1, 2, 2, 2, 1, 1

β. Να σχεδιάσετε την ουρά μετά το τέλος της εκτέλεσης.

```

front ← 1
rear ← 2
κ ← 0
Χ ← 10
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ Υ
  ΑΝ Υ = 1 ΤΟΤΕ
    ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
      ΓΡΑΨΕ 'Ουρά άδεια'
    ΑΛΛΙΩΣ_ΑΝ front = rear ΤΟΤΕ
      ΓΡΑΨΕ A[front]
      front ← 0
      rear ← 0
  ΑΛΛΙΩΣ
    Υ ← A [front]
    ΓΡΑΨΕ Υ
    front ← front + 1
  ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
  Χ ← Χ + 5

```



```

AN rear = 10 ΤΟΤΕ
  ΓΡΑΨΕ 'Ουρά γεμάτη'
ΑΛΛΙΩΣ_ΑΝ front = 0 ΚΑΙ rear = 0
ΤΟΤΕ
  front ← 1
  rear ← 1
  A[rear] ← X
ΑΛΛΙΩΣ
  rear ← rear + 1
  A[rear] ← X
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ front, rear
κ ← κ + 1
ΜΕΧΡΙΣ_ΟΤΟΥ κ = 8

```

- 29.17. Δίνεται το διπλανό τμήμα εντολών το οποίο χρησιμοποιεί μία δομή δεδομένων ουρά με τη μορφή ενός μονοδιάστατου πίνακα O[8].
Δίνεται και ο παρακάτω πίνακας.

Ουρά	front	rear	Έξοδος								
<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>									0	0	
.....											

Να εκτελέσετε το διπλανό τμήμα εντολών για τις τιμές εισόδου:
A, A, B, A, A, B, B, B
και μετά το τέλος της κάθε επανάληψης να συμπληρώσετε τον πίνακα ως εξής:
Στην πρώτη στήλη να σχεδιάσετε την ουρά με το κατάλληλο περιεχόμενο, στις στήλες front και rear να γράψετε τις τιμές των δεικτών και στην τελευταία στήλη να γράψετε τις τιμές που εμφανίζονται.

```

front ← 0
rear ← 0
k ← 30
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Υ
ΑΝ Υ = 'Α' ΤΟΤΕ
  k ← k div 2
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
  front ← 1
  rear ← 1
  O[rear] ← k
ΑΛΛΙΩΣ_ΑΝ rear < 8 ΤΟΤΕ
  rear ← rear + 1
  O[rear] ← k
ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
  ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ουρά άδεια'
  ΑΛΛΙΩΣ
    k ← O[front]
    front ← front + 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ rear - front + 1, k
ΜΕΧΡΙΣ_ΟΤΟΥ front > rear

```

ΕΥΡΕΣΗΣ ΛΑΘΩΝ

- 29.18. Ένας φαρμακοποιός, όταν παραλαμβάνει φάρμακα από την αποθήκη φαρμάκων, τα τοποθετεί κατά ημερομηνία λήξης και με τέτοιον τρόπο, ώστε πάντα να δίνει στον πελάτη αυτό που έχει την πιο σύντομη ημερομηνία λήξης. Αν σας έχει ανατεθεί να γράψετε ένα πρόγραμμα διαχεί-

ρισης της παραλαβής των φαρμάκων, να γράψετε ποια δομή δεδομένων θα χρησιμοποιήσετε για τον σκοπό αυτό και να δικαιολογήσετε την επιλογή σας.

- 29.19. Το διπλανό κύριο πρόγραμμα υλοποιεί τη λειτουργία της εξαγωγής όλων των στοιχείων από μία ουρά, η οποία αναπαρίσταται από τον μονοδιάστατο πίνακα A[20].
Για κάθε γραμμή που εντοπίζεται λογικό λάθος, να γράψετε, τον αριθμό γραμμής και δίπλα τη σωστή διόρθωση.

```
1. APXH
2. AN front = 1 KAI rear = 20 TOTE
3.   ΓΡΑΨΕ 'Ουρά άδεια'
4. ΑΛΛΙΩΣ_ΑΝ front = rear TOTE
5.   ΓΡΑΨΕ A[front]
6.   front ← 1
7.   rear ← 0
8. ΑΛΛΙΩΣ
9.   ΟΣΟ front > rear ΕΠΑΝΑΛΑΒΕ
10.  ΓΡΑΨΕ A[front]
11.  rear ← rear + 1
12. ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13. front ← 20
14. rear ← 0
15. ΤΕΛΟΣ_ΑΝ
16. ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

- 29.20. Δίνεται η ακολουθία αριθμών 5, 8, 2, 4, 1, 7 τους οποίους τοποθετούμε σε ουρά 10 θέσεων, στην 1η, 2η ... 6η θέση αντίστοιχα.
1. α. Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών στην ουρά;
β. Να σχεδιάσετε την παραπάνω ουρά μετά την τοποθέτηση των αριθμών και να προσδιορίσετε την τιμή του δείκτη «εμπρός» και του δείκτη «πίσω».
- 29.21. Δίνεται η ακολουθία γραμμάτων B, D, E, A, F τα οποία τοποθετούμε σε ουρά 8 θέσεων, στην 1η, 2η ... 5η θέση αντίστοιχα.
1. α. Να σχεδιάσετε την παραπάνω ουρά μετά την τοποθέτηση των αριθμών και να προσδιορίσετε την τιμή του δείκτη «εμπρός» και του δείκτη «πίσω».
β. Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο αριθμών από την ουρά;
γ. Πόσες φορές θα πρέπει να γίνει η παραπάνω λειτουργία στην ουρά για να εξέλθει το γράμμα E ;
- 29.22. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί από την 1η έως την 5η θέση κατά σειρά τα ονόματα MARK, TZORTZ, ΧΑΡΙ, ANTONIO, ΜΑΘΙΟΥ.
1. α. Να σχεδιάσετε την παραπάνω ουρά μετά την τοποθέτηση των στοιχείων και να προσδιορίσετε τις τιμές των δεικτών front και rear.
β. Εάν εφαρμοστούν οι λειτουργίες: εξαγωγή, εξαγωγή, εισαγωγή ΑΝΤΡΙΟΥ, εξαγωγή, εξαγωγή, εισαγωγή ΡΙΤΣΑΡΝΤ, εισαγωγή ΝΤΕΙΒΙΝΤ, να σχεδιάσετε την τελική μορφή της ουράς και να προσδιορίσετε τις νέες τιμές των δεικτών της.

29.34. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που:

3

- α. Θα χρησιμοποιεί τον πίνακα ΟΥΡΑ[50] για την υλοποίηση ουράς 50 θέσεων.
β. Θα εκτελεί επαναληπτικά τη λειτουργία της εισαγωγής και της εξαγωγής ως εξής: διαβάζει το είδος της λειτουργίας που θα εκτελεστεί («ΕΙ» για εισαγωγή και «ΕΞ» για εξαγωγή). Όταν πρέπει να εκτελεστεί η λειτουργία της εισαγωγής, διαβάζει έναν ακέραιο αριθμό και αν η ουρά μπορεί να δεχθεί αριθμούς, τον τοποθετεί σ' αυτή, ενώ τη λειτουργία της εξαγωγής την εκτελεί, αν υπάρχουν αριθμοί στην ουρά. Η επαναληπτική διαδικασία τερματίζεται, όταν γεμίσει η ουρά (δεν μπορεί να γίνει εισαγωγή αριθμών), οπότε και εμφανίζει το μήνυμα «ΓΕΜΑΤΗ ΟΥΡΑ», ή όταν αδειάσει η ουρά, οπότε και εμφανίζει το μήνυμα «ΑΔΕΙΑ ΟΥΡΑ».

29.35. Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

3

- α. Χρησιμοποιεί τον πίνακα Β[30] για να αναπαραστήσει μια ουρά 30 θέσεων.
β. Διαβάζει το είδος της λειτουργίας που θα εκτελεστεί στην ουρά («1» για εισαγωγή και «2» για εξαγωγή). Όταν πρέπει να εκτελεστεί η λειτουργία της εισαγωγής, διαβάζει έναν ακέραιο αριθμό και τον τοποθετεί στη ουρά, αν υπάρχει διαθέσιμος χώρος στο πίσω μέρος, διαφορετικά, εμφανίζει το μήνυμα «ΟΥΡΑ ΓΕΜΑΤΗ». Τη λειτουργία της εξαγωγής την εκτελεί, αν υπάρχουν αριθμοί στην ουρά, διαφορετικά, εμφανίζει το μήνυμα «ΑΔΕΙΑ ΟΥΡΑ».
γ. Δέχεται απάντηση (ΝΑΙ ή ΟΧΙ) για συνέχιση ή τερματισμό της επανάληψης μετά την εμφάνιση κατάλληλου μηνύματος.
δ. Μετά το τέλος της επαναληπτικής διαδικασίας:
1. Αν η ουρά είναι άδεια, εμφανίζει το μήνυμα «ΟΥΡΑ ΑΔΕΙΑ» διαφορετικά εμφανίζει πόσα δεδομένα υπάρχουν μέσα στην ουρά.
2. Εμφανίζει πόσες φορές άδειασε η ουρά.
3. Εμφανίζει το άθροισμα των αριθμών που εξήχθησαν από την ουρά.

29.36. Πρωτοετείς φοιτητές αποφάσισαν να συμμετάσχουν σε εθελοντική αιμοδοσία και γι' αυτόν τον λόγο προσήλθαν στο πλησιέστερο κέντρο αιμοδοσίας. Ο υπεύθυνος της αιμοδοσίας σκέφτηκε να σχηματίσει μία «ουρά αιμοδοσίας» τοποθετώντας κάθε φοιτητή στο πίσω μέρος της ουράς, ενώ αντίστοιχα έπαιρνε τον πρώτο φοιτητή της ουράς για να αιμοδοτήσει. Πριν ο υπεύθυνος τοποθετήσει κάθε φοιτητή στην «ουρά αιμοδοσίας», τον ρωτούσε αν πληρούσε τις προϋποθέσεις αιμοδότησης και αντίστοιχα τον απέρριπτε ή όχι. Να κατασκευάσετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο διαχειρίζεται την «ουρά αιμοδοσίας» εκτελώντας τις παρακάτω ενέργειες:

3

- α. Θα χρησιμοποιεί τον πίνακα ΟΝ[100] για την υλοποίηση της «ουράς αιμοδοσίας».
β. Θα εκτελεί τη λειτουργία της εισαγωγής και της εξαγωγής φοιτητών από την ουρά διαβάζοντας την επιλογή της λειτουργίας που θα εκτελεστεί («ΕΣ» για εισαγωγή νέου φοιτητή και «ΕΞ» για την εξαγωγή φοιτητή). Όταν πρέπει να εκτελεστεί η λειτουργία της εισαγωγής, ρωτάει με κατάλληλο μήνυμα αν ο φοιτητής πληροί τις προϋποθέσεις αιμοδοσίας, οπότε, διαβάζει το όνομα του φοιτητή και τον τοποθετεί στην ουρά, αν αυτή δεν έχει γεμίσει, διαφορετικά εμφανίζει το μήνυμα «Ελάτε την επόμενη φορά». Όταν

πρέπει να εκτελεστεί η λειτουργία της εξαγωγής την εκτελεί, αν υπάρχουν φοιτητές στη ουρά, εμφανίζοντας το όνομα του φοιτητή που αιμοδοτεί. Η διαδικασία επαναλαμβάνεται, μέχρι να αιμοδοτήσουν όλοι οι φοιτητές από την ουρά αιμοδοσίας.

γ. Μετά το τέλος της αιμοδοσίας:

1. Εμφανίζει το ποσοστό των φοιτητών που αιμοδότησαν.
2. Εμφανίζει το πλήθος των φοιτητών που δεν κατάφεραν να αιμοδοτήσουν.

29.37. Ένας τρισδιάστατος εκτυπωτής χρησιμοποιεί μια «ουρά εκτύπωσης», για να τοποθετεί σε αυτήν τα αρχεία προς εκτύπωση με τη σειρά με την οποία στέλνονται. Κάθε φορά εκτυπώνει το αρχείο που βρίσκεται στην αρχή της ουράς εκτύπωσης, το οποίο και εξάγει από αυτήν. Χρησιμοποιώντας τον πίνακα EKT[20] για την υλοποίηση της «ουράς εκτύπωσης» και με δεδομένο ότι τα αρχεία file2.doc, file3.doc, file1.doc έχουν καταφθάσει για εκτύπωση με τη σειρά με την οποία δίνονται, να κατασκευάσετε κύριο πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο:

3

α. Διαβάζει επαναληπτικά το γράμμα «N» που καθορίζει την έλευση νέου αρχείου ή το γράμμα «E» που δηλώνει την προσπάθεια εκτύπωσης ενός αρχείου πραγματοποιώντας έλεγχο ορθής καταχώρισης.

β. Εκτελεί τις παρακάτω ενέργειες:

- κατά την έλευση ενός νέου αρχείου, εξετάζει αν υπάρχει διαθέσιμος χώρος στην «ουρά εκτύπωσης» και τότε μόνο διαβάζει το όνομα του και το καταχωρίζει σε αυτήν με τη λειτουργία της «εισαγωγής», διαφορετικά, εμφανίζει το μήνυμα «Αδύνατη η εισαγωγή νέου αρχείου»,
- όταν ένα αρχείο πρόκειται να εκτυπωθεί, εξετάζεται αν υπάρχουν διαθέσιμα αρχεία και στη συνέχεια με τη λειτουργία της «εξαγωγής» αφαιρείται από την ουρά το κατάλληλο αρχείο και εμφανίζει το μήνυμα «Εκτύπωση» και δίπλα το όνομα του αρχείου που εκτυπώνεται.

Η επαναληπτική διαδικασία ολοκληρώνεται, όταν εκτυπωθούν όλα τα αρχεία που υπάρχουν στην «ουρά εκτύπωσης».

γ. Εμφανίζει:

1. τον συνολικό αριθμό αρχείων που εκτυπώθηκαν.
2. πόσες φορές η «ουρά εκτύπωσης» είχε ένα μόνο αρχείο.

9.38. Διαστημική εταιρεία εκτελεί κάθε μήνα πτήση χωρητικότητας 200 επιβατών, με δρομολόγιο από τη Γη στη Σελήνη. Λόγω της αυξημένης ζήτησης διατηρεί λίστα αναμονής 50 ατόμων, με όσους δεν μπόρεσαν να κλείσουν θέση ώστε αν προκύψει κάποια ακύρωση να ενημερώσουν τον πρώτο της λίστας για να επιβιβαστεί στην πτήση. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

4

α. Χρησιμοποιεί την κατάλληλη δομή δεδομένων (στοίβα ή ουρά) για να διαχειριστεί τη λίστα αναμονής.

β. Εμφανίζει μενού επιλογών:

1. Εγγραφή ατόμου στη λίστα
2. Ακύρωση επιβάτη από την πτήση
3. Έξοδος.

γ. Διαβάζει έναν ακέραιο αριθμού εξασφαλίζοντας ότι λαμβάνει τις τιμές 1 ή 2 ή 3, που καθορίζουν το είδος ενέργειας που θα εκτελεστεί σύμφωνα με το μενού επιλογών και πραγματοποιεί τα παρακάτω:

- Για την εγγραφή ενός ατόμου στη λίστα αναμονής, ζητάει το όνομα του ατόμου που θα προστεθεί σε αυτήν εφόσον η λίστα δεν έχει γεμίσει, διαφορετικά εμφανίζει το μήνυμα «Λίστα πλήρης».
- Για την ακύρωση ενός επιβάτη από την πτήση, εμφανίζει το όνομα του πρώτου ατόμου της λίστας αναμονής που θα επιβιβαστεί, ενώ εμφανίζει το μήνυμα «Άδεια λίστα αναμονής» αν δεν υπάρχουν άτομα σε αυτήν.
- Τερματίζει την παραπάνω επαναληπτική διαδικασία αν επιλεχθεί η έξοδος.

δ. Μετά το τέλος της επαναληπτικής διαδικασίας, εμφανίζει:

1. Το πλήθος των ατόμων που κατάφεραν να επιβιβαστούν στην πτήση από τη λίστα αναμονής.
2. Αν η λίστα αναμονής είναι άδεια, το μήνυμα «Κενή λίστα αναμονής» διαφορετικά το πλήθος των ατόμων που υπάρχουν σε αυτήν.
3. Το μέγιστο πλήθος των ατόμων που περίμεναν στη λίστα αναμονής.

29.39. Κατά την είσοδό τους στον χώρο πληρωμών μιας εταιρείας ύδρευσης οι πολίτες παίρνουν διαδοχικούς αριθμούς προτεραιότητας που καθορίζουν τη σειρά τους στην ουρά του μοναδικού ταμείου η οποία δεν μπορεί να ξεπερνάει τα 50 άτομα. Ο ταμίας εξυπηρετεί κάθε φορά τον πρώτο πολίτη στην ουρά, η εξυπηρέτησή του διαρκεί 4 λεπτά ακριβώς και έπειτα ο πολίτης αποχωρεί από την ουρά. Κάθε πολίτης που περιμένει στην ουρά έχει ένα χρόνο αναμονής που εξαρτάται από το πλήθος των ατόμων που θα εξυπηρετηθούν πριν από αυτόν. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο:

α. Χρησιμοποιεί μια ουρά δεδομένων την οποία υλοποιεί με τον πίνακα ON[50].

β. Δέχεται από τον χρήστη ως είσοδος μια εκ των τιμών «ΕΙΣΟΔΟΣ» ή «ΕΠΟΜΕΝΟΣ» οι οποίες καθορίζουν το είδος της ενέργειας που θα πραγματοποιηθεί:

1. Αν δοθεί η τιμή «ΕΙΣΟΔΟΣ», διαβάζει το ονοματεπώνυμο του πελάτη και αν η ουρά έχει γεμίσει εμφανίζει το μήνυμα «Ελάτε σε μία ώρα», διαφορετικά τον τοποθετεί στην ουρά και εμφανίζει το πλήθος των ατόμων που περιμένουν στην ουρά πριν από αυτόν καθώς και τον χρόνο αναμονής.

2. Αν δοθεί η τιμή «ΕΠΟΜΕΝΟΣ», εμφανίζει τον ονοματεπώνυμο του πολίτη που θα εξυπηρετηθεί.

Η παραπάνω διαδικασία επαναλαμβάνεται, μέχρι να εξυπηρετηθούν όλοι οι πολίτες.

γ. Μετά το τέλος της εξυπηρέτησης εμφανίζει:

1. το πλήθος των πολιτών που εξυπηρετήθηκαν,
2. τον μέσο χρόνο αναμονής των πολιτών,
3. το ονοματεπώνυμο του πολίτη με τον ελάχιστο χρόνο αναμονής (θεωρείστε ότι υπάρχει ένας μόνο τέτοιος πολίτης).