

Δομές Δεδομένων

[Στοίβα - Ουρά]

{B1} Κεφ. 3ο (Σελ. 60-63)

{B2} Κεφ. 1ο (Σελ. 13-36)

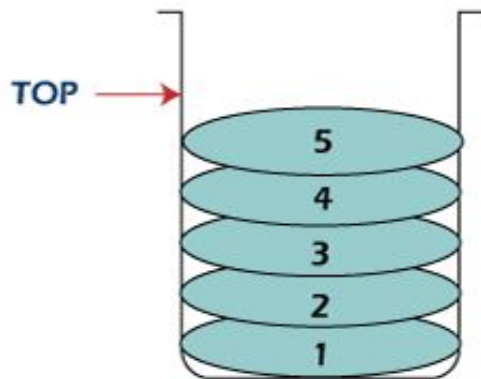
3.4 | Στοιίβα



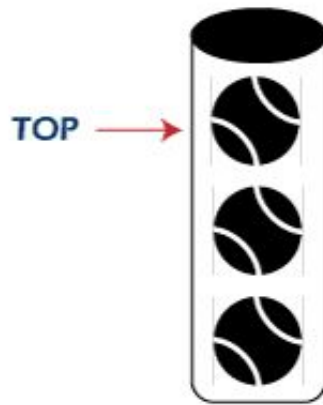
Στοιίβα (stack), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.



Stack of Coins



Stack of Plates

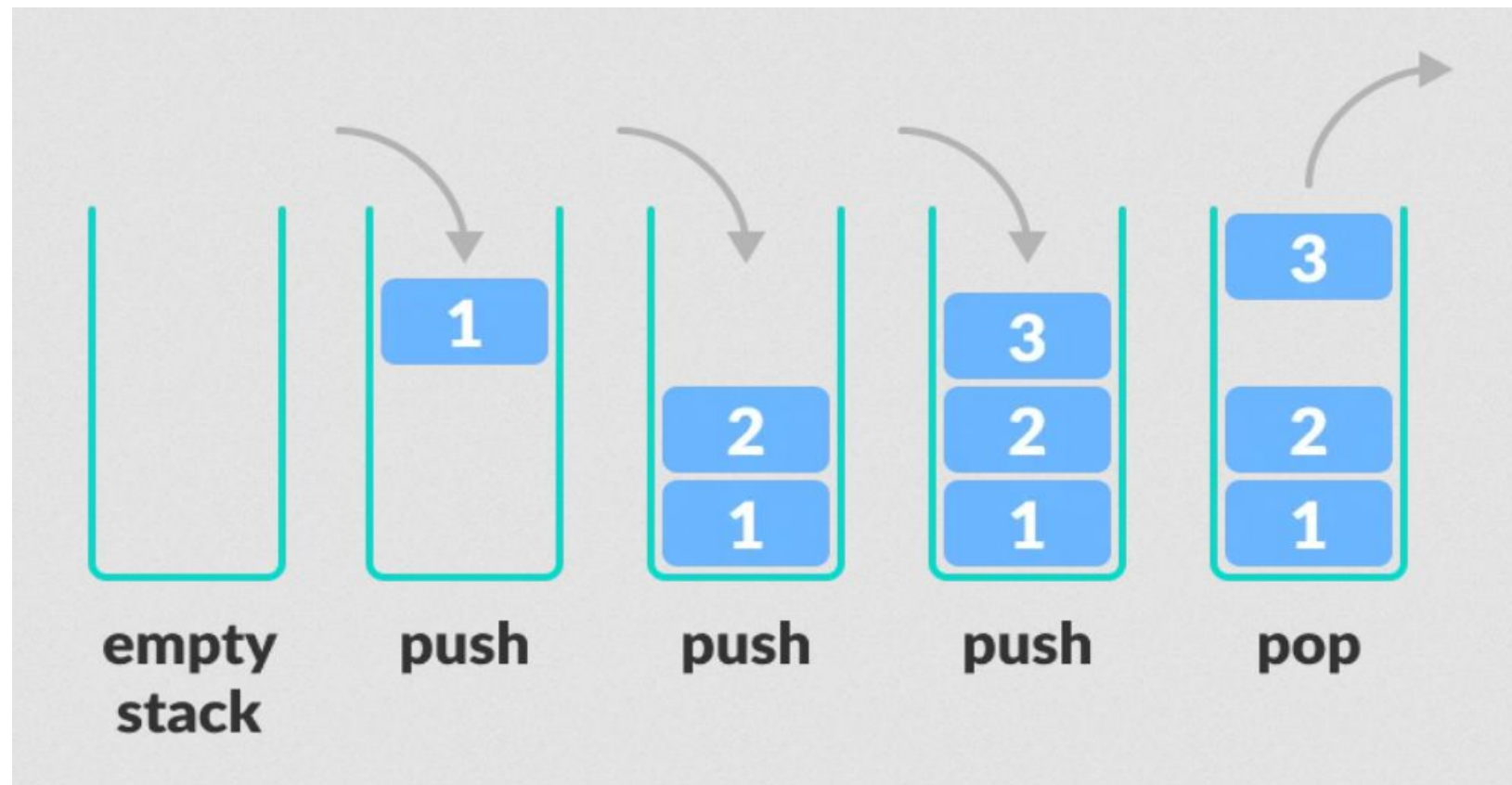


Can of Tennis Balls



Stack of Books

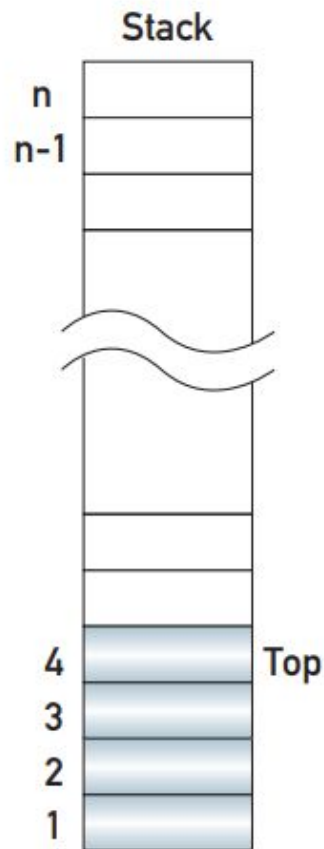
Η παραπάνω μέθοδος ονομάζεται **Τελευταίο Μέσα, Πρώτο Έξω** ή **LIFO** (=Last In First Out).





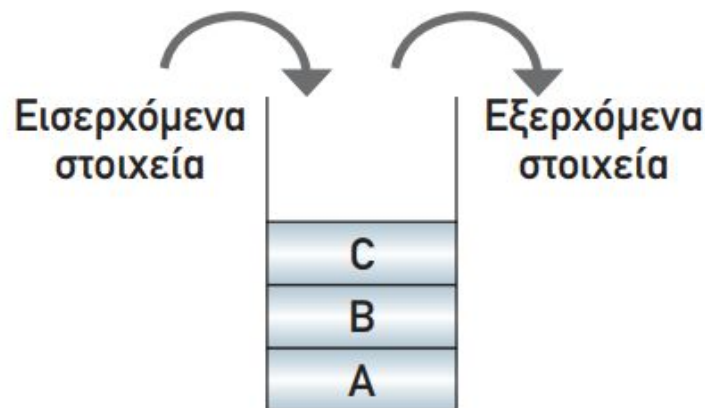
Last
In
First
Out



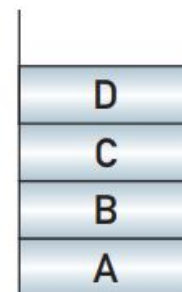


Δύο είναι οι κύριες λειτουργίες σε μία στοίβα:

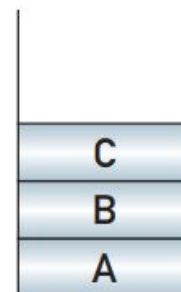
- ① η ώθηση (push) στοιχείου στην κορυφή της στοίβας, και
- ② η απώθηση (pop) στοιχείου από τη στοίβα.



Μια στοίβα με
τρία στοιχεία



Η στοίβα μετά
την εισαγωγή
του στοιχείου D
(λειτουργία
ώθηση)



Η στοίβα μετά
την εξαγωγή του
στοιχείου D
(λειτουργία
απώθηση)

Σχ. 3.3 Υλοποίηση στοίβας
με χρήση πίνακα

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.

```
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
```

???

ctrl + c

ctrl + v



stackoverflow

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.

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

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

2016 Επαναληπτικές

A5. Σε μια κενή στοίβα πρόκειται να εισαχθούν τα στοιχεία M, Δ, K, με αυτή τη σειρά. Δίνονται οι ακόλουθες σειρές διαδοχικών πράξεων (να θεωρήσετε ότι η λειτουργία της ώθησης παριστάνεται με το γράμμα ω και η λειτουργία της απώθησης παριστάνεται με το γράμμα α):

1. $\omega, \omega, \omega, \alpha, \alpha, \alpha \longrightarrow \boxed{K, \Delta, M}$

2. $\omega, \alpha, \omega, \alpha, \omega, \alpha \longrightarrow \boxed{M, \Delta, K}$

3. $\omega, \omega, \alpha, \alpha, \omega, \alpha \longrightarrow \boxed{\Delta, M, K}$

$\boxed{\Delta, K, M} \longleftarrow 4. \omega, \omega, \alpha, \omega, \alpha, \alpha$

5. $\omega, \alpha, \omega, \omega, \alpha, \alpha \longrightarrow \boxed{M, K, \Delta}$

Για καθεμιά από τις παραπάνω σειρές πράξεων να γράψετε στο τετράδιό σας τον αριθμό της (1 έως 5) και, δίπλα, μόνο τα στοιχεία που θα απωθηθούν με τη σειρά απώθησής τους.

Μονάδες 10

2020 - A3

- α) Μια στοίβα έξι θέσεων, ύστερα από μερικές ωθήσεις και απωθήσεις, έχει την παρακάτω μορφή:

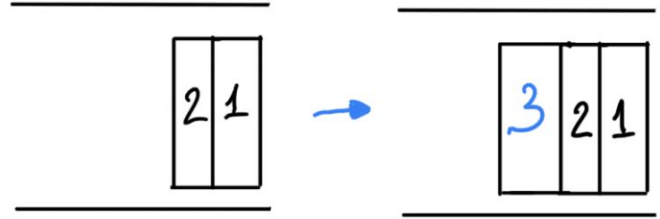
6		
5	8	
4	6	
3	7	← top
2	5	
1	2	

- i) Πόσες απωθήσεις πρέπει να γίνουν προκειμένου να αδειάσει η στοίβα; **ΤΡΕΙΣ (3)**
ΑΠΩΘΗΣΕΙΣ (μονάδες 2)
- ii) Να αιτιολογήσετε την απάντησή σας. (μονάδες 3)

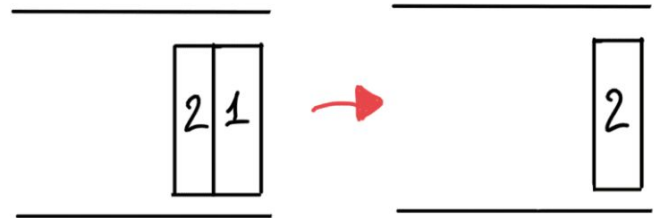
3.5 | Ουρά



ENQUEUE 3



DEQUEUE

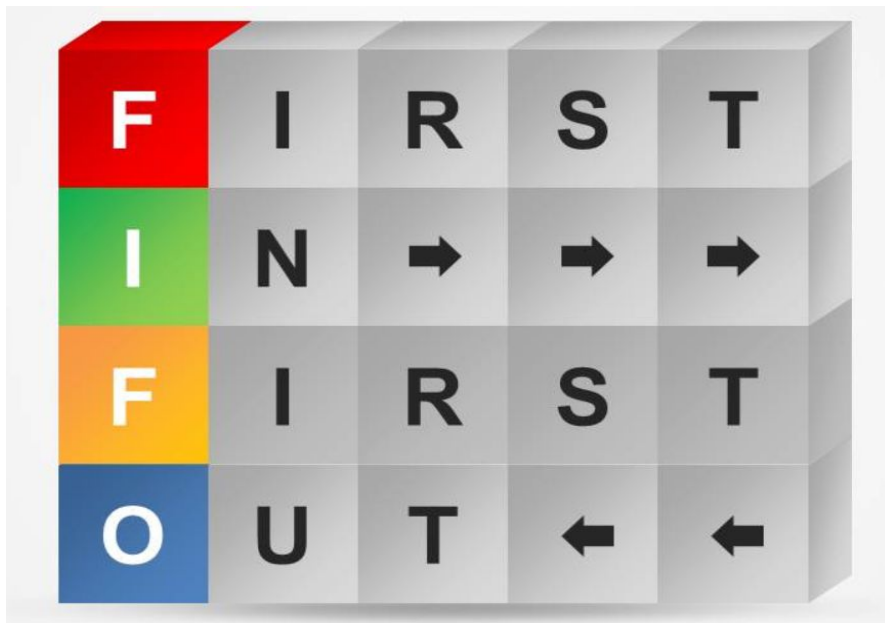


1



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.

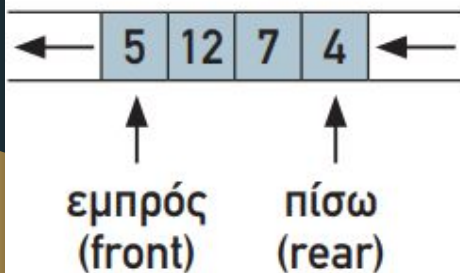
Η παραπάνω μέθοδος ονομάζεται **Πρώτο Μέσα, Πρώτο Έξω ή FIFO** (=First In First Out).



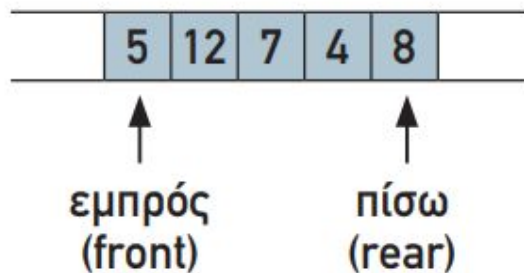


Οι **κύριες λειτουργίες** που εκτελούνται σε μια ουρά είναι δύο:

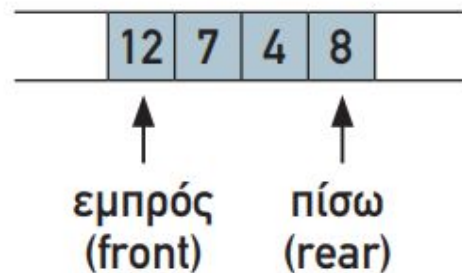
1. Η **εισαγωγή** (enqueue) στοιχείου στο πίσω άκρο της ουράς.
2. Η **εξαγωγή** (dequeue) στοιχείου από το εμπρός άκρο της ουράς.



α) Μια ουρά
με 4 στοιχεία



β) Η ουρά μετά
την εισαγωγή του
στοιχείου 8



γ) Η ουρά μετά
την εξαγωγή του
στοιχείου 5

Εισαγωγή

```
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
```

Κυκλική Ουρά (Αριστερή Ολίσθηση)



```
!Πριν την εισαγωγή  
ΑΝ rear = N ΚΑΙ front > 1 ΤΟΤΕ  
  k <- 0 !πλήθος  
  ΓΙΑ i ΑΠΟ front ΜΕΧΡΙ rear  
    k <- k + 1  
    ΟΥΡΑ[k] <- ΟΥΡΑ[i]  
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
  front <- 1  
  rear <- k  
ΤΕΛΟΣ_ΑΝ
```

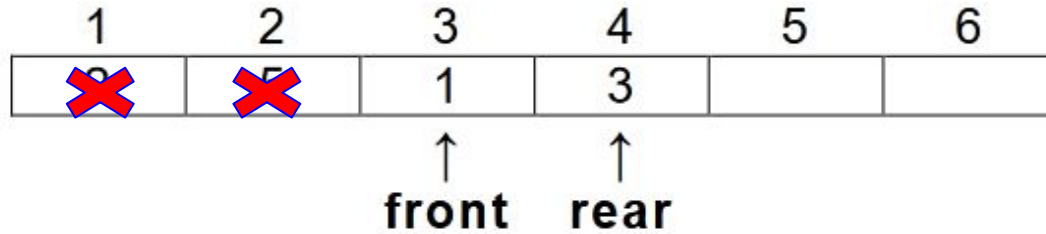
Εξαγωγή

```
1  AN (front = 0 KAI rear = 0) ΤΟΤΕ
2      ΓΡΑΨΕ 'Άδεια ουρά'
3  ΑΛΛΙΩΣ AN (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
```

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

2020 - A3

- β) Μια ουρά έξι θέσεων, ύστερα από μερικές εισαγωγές και εξαγωγές, έχει την παρακάτω μορφή:



- i) Πόσες εξαγωγές πρέπει να γίνουν προκειμένου να αδειάσει η ουρά; **ΔΥΟ (2) ΕΞΑΓΩΓΕΣ** (μονάδες 2)
- ii) Να αιτιολογήσετε την απάντησή σας. (μονάδες 3)

2021 - B2

Το παρακάτω τμήμα αλγορίθμου υλοποιεί τη λειτουργία της εξαγωγής στοιχείου από ουρά με χρήση μονοδιάστατου πίνακα A. Να γράψετε στο τετράδιό σας τους αριθμούς (1) έως (4) που αντιστοιχούν στα κενά του

```
ΑΝ front = 0 ΚΑΙ rear = 0 ΤΟΤΕ  
    ΓΡΑΨΕ 'Άδεια ουρά'  
ΑΛΛΙΩΣ_ΑΝ front = rear ΤΟΤΕ  
    ΓΡΑΨΕ 'Εξαγεται το στοιχείο:', A[front]  
    front ← 0  
    rear ← 0  
ΑΛΛΙΩΣ  
    ΓΡΑΨΕ 'Εξάνεται το στοιχείο:', A[front]  
    front ← front + 1  
ΤΕΛΟΣ_ΑΝ
```

Μονάδες 8

2023 - B3

Έστω στοίβα που υλοποιείται με μονοδιάστατο πίνακα $\Sigma[N]$ και ουρά που υλοποιείται με μονοδιάστατο πίνακα $O[N]$. Για τη διαχείριση των λειτουργιών της στοίβας χρησιμοποιείται μεταβλητή με όνομα **top**, ενώ για τη διαχείριση των λειτουργιών της ουράς χρησιμοποιούνται μεταβλητές με ονόματα **front** και **rear**. Δίνονται στη συνέχεια κάποιες περιπτώσεις ελέγχων που αφορούν στη στοίβα και στην ουρά. Για καθέναν από τους ελέγχους να γράψετε στο τετράδιό σας τον αριθμό του (1 έως 4) και δίπλα τη συνθήκη που υλοποιεί τον αντίστοιχο έλεγχο σε ψευδογλώσσα.

Μονάδες 4

1. Η στοίβα είναι άδεια.  **top = 0**
2. Η ουρά είναι γεμάτη.  **rear = N**
3. Η στοίβα έχει ένα στοιχείο.  **top = 1**
4. Η ουρά έχει δύο στοιχεία.  **rear - front = 1**

ΕΞΤΡΑ ΥΛΙΚΟ - LINKS

- <https://lalibi.github.io/aepp-presentation/#610>
- <https://aepp.edu.gr/lymenes-askiseis/stoives/>
- [Παρουσίαση Στοίβας \(Stack\) σε βίντεο](#)
- [Παρουσίαση Ουράς \(Queue\) σε βίντεο](#)
- [Ασκήσεις με Κυκλικές Ουρές \(Ολίσθηση\)](#)