



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.

- 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 (*AN rear = N TOTE*),
- είναι άδειος, ελέγχοντας αν οι δύο δείκτες έχουν την τιμή μηδέν (*ΑΛΛΙΩΣ_ΑΝ front = 0 ΚΑΙ rear = 0 TOTE*) οπότε οι δύο δείκτες παίρνουν την τιμή 1, καθώς είναι το πρώτο στοιχείο που εισάγεται στη θέση *rear* (*table[rear] ← X*),
- έχει χώρο (*ΑΛΛΙΩΣ*), οπότε αυξάνεται ο πίσω δείκτης κατά ένα (*rear ← rear + 1*) και στη θέση που δείχνει τοποθετείται το νέο στοιχείο (*table[rear] ← X*).

Το τμήμα εντολών που υλοποιεί την εισαγωγή φαίνεται δίπλα.

ΔΙΑΒΑΣΕ X ! αν ουρά γεμάτη

ΑΝ *rear = N TOTE*

ΓΡΑΨΕ 'Ουρά γεμάτη'

ΑΛΛΙΩΣ_ΑΝ *front = 0 ΚΑΙ rear = 0*

TOTE ! αν ουρά άδεια

front ← 1

rear ← 1

table[rear] ← X

ΑΛΛΙΩΣ ! έχει στοιχεία

rear ← rear + 1 ! και χωράει

table[rear] ← 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
ΑΝ rear = N **ΤΟΤΕ**
ΓΡΑΨΕ 'Ουρά γεμάτη'
ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ *! επανάληψη*
ΔΙΑΒΑΣΕ X
ΑΝ 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
ΑΝ 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
ΑΝ 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
ΑΝ rear = N **ΤΟΤΕ**
ΓΡΑΨΕ 'Ουρά γεμάτη'
ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Θέλεις να εισάγεις στοιχείο;'
ΔΙΑΒΑΣΕ απάντηση
ΑΝ απάντηση = 'ΝΑΙ' **ΤΟΤΕ**
ΔΙΑΒΑΣΕ X
ΑΝ front = 0 **ΚΑΙ** rear = 0 **ΤΟΤΕ**
front ← 1
rear ← 1
table[rear] ← X
rear ← rear + 1
table[rear] ← X
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ απάντηση = 'ΟΧΙ' **Ή** rear = N
ΑΝ rear = N **ΤΟΤΕ**
ΓΡΑΨΕ 'Ουρά γεμάτη'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

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

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



29.2. Αλγόριθμος υλοποίησης της λειτουργίας της εξαγωγής.

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', $table[front]$

$front \leftarrow 0$

$rear \leftarrow 0$

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Εξάγεται το', $table[front]$

$front \leftarrow front + 1$

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`, διάφοροι του μηδενός, σημαίνει ότι ο πίνακας έχει ένα μόνο στοιχείο μέσα, άρα και η ουρά έχει ένα μόνο στοιχείο.
 - Αν `0 < rear < 100`, σημαίνει ότι ο πίνακας έχει κάποια στοιχεία μέσα και η ουρά δεν είναι γεμάτη.
 - Αν `front > 1` και `rear=100`, θεωρούμε ότι η ουρά είναι γεμάτη στο πίσω μέρος και έχει κενές θέσεις μπροστά.