

ΑΣΚΗΣΕΙΣ ΤΑΞΗ

Στοίβα και Ουρά

ΘΕΩΡΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ

1. ΕΚΦΩΝΗΣΗ

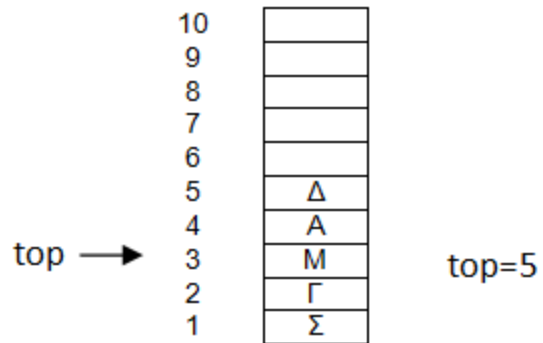
- ◉ Σε μια στοίβα 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
- ◉ Να προσδιορίσετε την τιμή του δείκτη top της παραπάνω στοίβας και να την σχεδιάσετε.
- ◉ Αν εφαρμόσουμε τις παρακάτω λειτουργίες: Απώθηση, Απώθηση, Απώθηση, Ώθηση X , Ώθηση Δ και Απώθηση ποιά είναι η νέα τιμή της top και ποιά η τελική μορφή της στοίβας;

ΛΥΣΗ

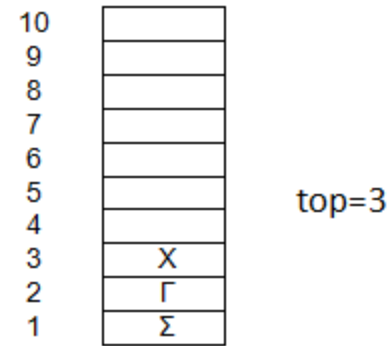


1)

i)



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



`top` →

2. ΕΚΦΩΝΗΣΗ

- ◉ Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
- ◉ Να προσδιορίσετε τις τιμές των δεικτών rear και front της παραπάνω ουράς και να την σχεδιάσετε.
- ◉ Αν εφαρμόσουμε τις παρακάτω λειτουργίες: Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Χ, Εισαγωγή Δ και Εξαγωγή ποιές είναι οι νέες τιμές των δεικτών rear και front και ποιά η τελική μορφή της ουράς;

ΛΥΣΗ

2)

i) Η αρχική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Σ	Γ	Μ	Α	Δ					
front				rear					

οι τιμές της front και της rear είναι: front=1 και rear=5

ii) Η τελική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
				Δ	X	Δ			
				front		rear			

και οι τιμές της front και της rear γίνονται: front=5 και rear=7

3. ΕΚΦΩΝΗΣΗ

- Σε μια άδεια στοίβα 10 θέσεων ωθούμε τα στοιχεία Ο, Σ, Ε, Τ, Λ. Με ποιό τρόπο πρέπει να ωθηθούν και να απωθηθούν τα δεδομένα ώστε η στοίβα να περιέχει τα δεδομένα Τ, Ε, Λ, Ο, Σ (σε αύξουσες θέσεις του πίνακα).

ΛΥΣΗ

Η αρχική μορφή της στοίβας είναι:

10	
9	
8	
7	
6	
5	Λ
4	Τ
3	Ε
2	Σ
1	Ο

top → 5

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

10	
9	
8	
7	
6	
5	Σ
4	Ο
3	Λ
2	Ε
1	Τ

top → 5

4. ΕΚΦΩΝΗΣΗ

- Σε μια άδεια ουρά 10 θέσεων εισάγουμε τα στοιχεία Ο, Σ, Ε, Τ, Λ. Με ποιό τρόπο πρέπει να εισαχθούν και να εξαχθούν τα δεδομένα ώστε η στοίβα να περιέχει τα δεδομένα Τ, Ε, Λ, Ο, Σ (σε αύξουσες θέσεις του πίνακα).

ΛΥΣΗ

4) i) Η αρχική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Ο	Σ	Ε	Τ	Λ					
front				rear					

οι τιμές της front και της rear είναι: front=1 και rear=5

ii) Εκτελώντας τις λειτουργίες : **Εξαγωγή, Εξαγωγή, Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Τ, Εισαγωγή Ε, Εισαγωγή Λ, Εισαγωγή Ο, Εισαγωγή Σ**, τότε η τελική μορφή της ουράς γίνεται

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
					Τ	Ε	Λ	Ο	Σ
					front				
					rear				

και οι τιμές της front
γίνονται: front=6 και rear=10.

και της rear

5. ΕΚΦΩΝΗΣΗ

- Σε μια κενή στοίβα πρόκειται να εισαχθούν τα στοιχεία A, M, D, K, L, B με τη σειρά που δίνονται (A πρώτο, B τελευταίο). Ακολουθεί μια σειρά πράξεων που είναι:
- α) Ώθηση δύο στοιχείων στη στοίβα και απώθηση ενός
- β) Ώθηση δύο στοιχείων στη στοίβα και απώθηση ενός
- γ) Ώθηση δύο στοιχείων στη στοίβα και απώθηση ενός
- Ποια στοιχεία και με ποια σειρά, περιέχει η στοίβα μετά τις πράξεις αυτές;
- Η ανωτέρω άσκηση μπορεί να υλοποιηθεί σε γλώσσα με χρήση ενός πίνακα 10 θέσεων

ΛΥΣΗ

A(Ωθηση)	A(Απώθηση)	B(Ωθηση)	B(Απώθηση)	Γ(Ωθηση)	Γ(Απώθηση)
10					
9					
8					
7					
6					
5					
4				B	
3		K		L	L
2	M	D	D	D	D
1	A	A	A	A	A
top=2	top=1	top=3	top=2	top=4	top=3

6. ΕΚΦΩΝΗΣΗ

- Σε μια κενή ουρά πρόκειται να εισαχθούν τα στοιχεία A, M, D, K, L, B με τη σειρά που δίνονται (A πρώτο, B τελευταίο). Ακολουθεί μια σειρά πράξεων που είναι:
- α) Εισαγωγή δύο στοιχείων στη στοίβα και εξαγωγή ενός
- β) Εισαγωγή δύο στοιχείων στη στοίβα και εξαγωγή ενός
- γ) Εισαγωγή δύο στοιχείων στη στοίβα και εξαγωγή ενός
- Ποια στοιχεία και με ποια σειρά, περιέχει η ουρά μετά τις πράξεις αυτές

ΛΥΣΗ

	1	2	3	4	5	6	7	8	9	10
A(Εισαγωγή)	A	M								
A(Εξαγωγή)		M								

front	rear
1	2
2	2

B(Εισαγωγή)		M	D	K						
B(Εξαγωγή)			D	K						

2	4
3	4

Γ(Εισαγωγή)			D	K	L	B				
Γ(Εξαγωγή)				K	L	B				

3	6
4	6

ΑΣΚΗΣΕΙΣ ΣΕ ΓΛΩΣΣΑ

1. ΕΚΦΩΝΗΣΗ

- Να γραφεί κομμάτι προγράμματος που υλοποιεί την ώθηση και την απώθηση σε στοίβα A. Η στοίβα έχει την μορφή πίνακα 10 θέσεων

ΛΥΣΗ

Γράψε 'ΩΘΗΣΗ'

ΓΡΑΨΕ 'Δώσε στοιχείο για να εισαχθεί στη στοίβα A:'

ΔΙΑΒΑΣΕ στοιχείο

ΑΝ $top < 10$ ΤΟΤΕ

$top \leftarrow top + 1$

$A[top] \leftarrow \text{στοιχείο}$

ΑΛΛΙΩΣ

 ΓΡΑΨΕ 'Υπερχείλιση στοίβας'

ΤΕΛΟΣ_ΑΝ

Γράψε 'ΑΠΩΘΗΣΗ'

ΑΝ $top \geq 1$ ΤΟΤΕ

$\text{Στοιχείο} \leftarrow A[top]$

$top \leftarrow top - 1$

ΑΛΛΙΩΣ

 ΓΡΑΨΕ 'Υποχείλιση στοίβας'

ΤΕΛΟΣ_ΑΝ

2. ΕΚΦΩΝΗΣΗ

- Να γραφεί διαδικασία που υλοποιεί την ώθηση και την απώθηση σε στοίβα A, Η στοίβα έχει την μορφή πίνακα 10 θέσεων

ΛΥΣΗ

ΔΙΑΔΙΚΑΣΙΑ ΩΘΗΣΗ (A, στοιχείο, top, done)

ΣΤΑΘΕΡΕΣ

N = 10

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top

ΧΑΡΑΚΤΗΡΕΣ: στοιχείο, A[N]

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top < N ΤΟΤΕ

top <-- top + 1

A[top] <-- στοιχείο

done <-- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <-- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΠΩΘΗΣΗ (A, στοιχείο, top, done)

ΣΤΑΘΕΡΕΣ

N = 10

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top

ΧΑΡΑΚΤΗΡΕΣ: στοιχείο, A[N]

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top >= 1 ΤΟΤΕ

στοιχείο <-- A[top]

top <-- top - 1

done <-- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <-- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

3. ΕΚΦΩΝΗΣΗ

- ⊙ Ένα οχηματαγωγό πλοίο, χωρητικότητας 250 αυτοκινήτων, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ - ΑΙΓΙΝΑ. Τα οχήματα που επιβιβάζονται πρώτα είναι αυτά που θα αποβιβαστούν τελευταία. Στο λιμάνι του Πειραιά προσέρχονται τα αυτοκίνητα για αναχώρηση. Να γίνει πρόγραμμα το οποίο:
 - ⊙ 1. Να υπάρχει μενού επιλογής:

1. Επιβίβαση2. Αποβίβαση3. Έξοδος
 - ⊙ 2. Στη περίπτωση που επιλεγθεί η Επιβίβαση θα διαβάζει τον αριθμό κυκλοφορίας καθενός από τα αυτοκίνητα που προσέρχονται και ο αριθμός κυκλοφορίας του να καταχωρείται στη στοίβα ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα αυτοκίνητο να τυπώνεται το ερώτημα "Υπάρχει άλλο αυτοκίνητο (N/O); ". Αν ο χρήστης απαντήσει N (=ΝΑΙ), επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει Ο (=ΟΧΙ), σταματά η διαδικασία επιβίβασης και επιστρέφει το πρόγραμμα στο μενού Επιλογής.
 - ⊙ 3. Αν το πλοίο γεμίσει η επιβίβαση σταματά εμφανίζεται κατάλληλο μήνυμα και επιστρέφει το πρόγραμμα στο μενού επιλογής.
 - ⊙ 4. Στη περίπτωση που επιλεγθεί η Αποβίβαση, εξάγει και εμφανίζει από την στοίβα ΟΧΗΜΑΤΑ όλους τους αριθμούς αυτοκινήτων που είχαν επιβιβαστεί στον ΠΕΙΡΑΙΑ, με τη σειρά που αποβιβάζονται. Στο τέλος να τυπώνεται το πλήθος των αυτοκινήτων που αποβιβάστηκαν στο λιμάνι της ΑΙΓΙΝΑΣ

ΛΥΣΗ

ΠΡΟΓΡΑΜΜΑ Λιμάνι

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, η[5]

ΑΡΧΗ

τοπ <- 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ' Μενού Επιλογών'

ΓΡΑΨΕ ' 1. Επιβίβαση'

ΓΡΑΨΕ ' 2. Αποβίβαση'

ΓΡΑΨΕ ' 3. Έξοδος'

ΓΡΑΨΕ ' Δώσε επιλογή:'

ΔΙΑΒΑΣΕ επ1

ΑΝ επ1 = 1 ΤΟΤΕ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ' Υπάρχει αυτοκίνητο για

επιβίβαση (Ν/Ο);'

ΔΙΑΒΑΣΕ επ2

ΑΝ επ2 <> 'Ν' ΚΑΙ επ2 <> 'ν' ΚΑΙ επ2 <> 'Ο'

ΚΑΙ επ2 <> 'ο' ΤΟΤΕ

ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ επ2 = 'Ο' Η επ2 = 'ο' Η επ2 = 'Ν'

Η επ2 = 'ν'

ΑΝ επ2 = 'Ν' Η επ2 = 'ν' ΤΟΤΕ

ΑΝ τοπ < 5 ΤΟΤΕ

ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του αυτοκινήτου:'

ΔΙΑΒΑΣΕ αρ

τοπ <- τοπ + 1

η[τοπ] <- αρ

ΑΝ τοπ = 5 ΤΟΤΕ

ΓΡΑΨΕ 'Το πλοίο γέμισε και δεν χωρά άλλα
αμάξια'

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Το πλοίο είναι γεμάτο'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ τοπ = 5 Η επ2 = 'Ο' Η επ2 = 'ο'

ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ

πλ <- 0

ΟΣΟ τοπ >= 1 ΕΠΑΝΑΛΑΒΕ

ΓΡΑΨΕ 'Αποβιβάζεται το αυτοκίνητο με αριθμό
κυκλοφορίας:', η[τοπ]

η[τοπ] <- ' '

τοπ <- τοπ - 1

πλ <- πλ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Πλήθος οχημάτων που αποβιβάστηκαν στο
λιμάνι της ΑΙΓΙΝΑΣ:', πλ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Λιμάνι

4. ΕΚΦΩΝΗΣΗ

- ◉ Να δημιουργηθεί πρόγραμμα σε γλώσσα το οποίο θα ωθεί 5 αριθμούς σε στοίβα και στη συνέχεια θα τους απωθεί και θα τους εμφανίζει με τη σειρά που απωθούνται. Η ώθηση και η απώθηση να γραφτούν σε διαδικασίες

ΠΡΟΓΡΑΜΜΑ ΠΕΝΤΕ_ΑΡΙΘΜΟΙ_ΣΕ_ΣΤΟΙΒΑ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Stack[100], i, top, number, item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

top <- 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5

ΔΙΑΒΑΣΕ number

ΚΑΛΕΣΕ ΩΘΗΣΗ (Stack, number, top, done)

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5

ΚΑΛΕΣΕ ΑΠΩΘΗΣΗ (Stack, item, top, done)

ΓΡΑΨΕ item

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΩΘΗΣΗ (Stack, item, top, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Stack[100], item, top

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top < 100 ΤΟΤΕ

top <- top + 1

Stack[top] <- item

done <- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΠΩΘΗΣΗ (Stack, item, top, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Stack[100], item, top

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top <> 0 ΤΟΤΕ ! ή top > = 1

item <- Stack[top]

top <- top - 1

done <- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

4. ΕΚΦΩΝΗΣΗ (ΕΚΤΟΣ ΠΑΝΕΛΛΗΝΙΩΝ 2015)

- α) Να γράψετε πρόγραμμα που θα διαβάσει 10 ακεραίους αριθμούς και θα κάνει εισαγωγή των αριθμών σε μια ουρά 10 θέσεων, στη συνέχεια να εξαχθούν 6 αριθμοί οι οποίοι θα εμφανίζονται με τη σειρά που εξάγονται και να εισαχθούν άλλοι 3 ακέραιοι αριθμοί.
- Να εμφανιστούν κατά σειρά οι αριθμοί που βρίσκονται στο τέλος μέσα στην ουρά.
- β) Το πρόγραμμα θα καλεί τις διαδικασίες ΕΙΣΑΓΩΓΗ και ΕΞΑΓΩΓΗ που θα υλοποιούν τις αντίστοιχες λειτουργίες της ουράς τις οποίες και θα γράψετε. Αν η ουρά γεμίσει αλλά έχει άδειες θέσεις μπροστά, τότε να μεταφέρονται τα στοιχεία που βρίσκονται στην ουρά μπροστά ώστε να συνεχίζεται η διαδικασία της εισαγωγής.
- Θα επιστρέφονται από τις διαδικασίες οι λογικές τιμές ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ αναλόγως αν η εισαγωγή ή η εξαγωγή στοιχείου στην ουρά έγιναν επιτυχώς ή όχι.

ΛΥΣΗ

ΠΡΟΓΡΑΜΜΑ ΚΥΚΛΙΚΗ_ΟΥΡΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: front, rear, i, number, item, Queue[10]

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

front <- - 0

rear <- - 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΓΡΑΨΕ 'Δώστε ακέραιο αριθμό'

ΔΙΑΒΑΣΕ number

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ (Queue, number, front, rear, done)

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6

ΚΑΛΕΣΕ ΕΞΑΓΩΓΗ (Queue, item, front, rear, done)

ΓΡΑΨΕ item

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3

ΓΡΑΨΕ 'Δώστε ακέραιο αριθμό'

ΔΙΑΒΑΣΕ number

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ (Queue, number, front, rear, done)

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ front ΜΕΧΡΙ rear

ΓΡΑΨΕ Queue[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΑΓΩΓΗ (Queue, item, front, rear, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Queue[10], item, front, rear, S, i

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ rear < 10 ΤΟΤΕ

rear <- rear + 1

Queue[rear] <- item

ΑΝ front = 0 ΤΟΤΕ

front <- 1

ΤΕΛΟΣ_ΑΝ

done <- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ_ΑΝ front = 1 ΤΟΤΕ

done <- ΨΕΥΔΗΣ

ΑΛΛΙΩΣ

S <- rear - front + 1

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ S

Queue[i] <- Queue[front]

front <- front + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

front <- 1

rear <- S + 1 ! ή rear <-- i

Queue[rear] <- item

done <- ΑΛΗΘΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΞΑΓΩΓΗ (Queue, item, front, rear, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Queue[10], item, front, rear

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ front <= rear ΚΑΙ front <> 0 ΤΟΤΕ

item <- Queue[front]

front <- front + 1

done <- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ