

ΣΤΟΙΒΑ - ΟΥΡΑ





ΘΕΩΡΙΑ

ΤΙ ΕΙΝΑΙ Η ΣΤΟΙΒΑ;

Η στοίβα είναι μια δομή δεδομένων όπου οι 2 κύριες λειτουργίες της είναι η ώθηση στοιχείου στην κορυφή και η απώθηση στοιχείου από την κορυφή της

Τα στοιχεία της στοίβας είναι διατεταγμένα με τέτοιο τρόπο, ώστε το τελευταίο(πιο πρόσφατο) στοιχείο που εισάγεται σε αυτήν να είναι και το πρώτο στοιχείο το οποίο απωθείται (για αυτό ονομάζεται και δομή δεδομένων LIFO – Last In First Out)

Η στοίβα είναι μια στατική δομή δεδομένων(βασίζεται σε μονοδιάστατο πίνακα σταθερού μεγέθους), που σημαίνει πως όταν θέλουμε να εισάγουμε ένα νέο στοιχείο, θα πρέπει προηγουμένως να εξετάσουμε αν υπάρχει ο κίνδυνος της υπερχείλισης(ήδη γεμάτη στοίβα χωρίς επιπλέον χώρο), ενώ αντίστοιχα όταν απωθούμε ένα στοιχείο, θα πρέπει να εξετάσουμε αν υπάρχει ο κίνδυνος της υποχείλισης(αν η στοίβα είναι άδεια)

ΤΙ ΕΙΝΑΙ Η ΟΥΡΑ;

Η ουρά είναι μια δομή δεδομένων όπου οι 2 κύριες λειτουργίες της είναι η εισαγωγή στοιχείου στο τέλος της ουράς και η εξαγωγή στοιχείου από την αρχή της

Τα στοιχεία της ουράς είναι διατεταγμένα με τέτοιο τρόπο, ώστε το τελευταίο(πιο πρόσφατο) στοιχείο που εισάγεται σε αυτήν να είναι το τελευταίο στοιχείο το οποίο εξάγεται (για αυτό ονομάζεται και δομή δεδομένων FIFO – First In First Out)

ΔΕΙΚΤΕΣ ΓΙΑ ΤΗΝ ΣΤΟΙΒΑ / ΟΥΡΑ

1. Για την στοίβα χρησιμοποιείται ένας δείκτης(top)
2. Για την ουρά χρησιμοποιούνται δύο δείκτες(rear για την εισαγωγή ενός στοιχείου / front για την εξαγωγή)

ΠΑΡΑΤΗΡΗΣΕΙΣ

1. Η στοίβα και η ουρά είναι στατικές δομές δεδομένων
2. Σε μια δομή δεδομένων ουρά, το πρώτο στοιχείο που έχει εισαχθεί είναι και αυτό που θα εξαχθεί πρώτο κατά την πρώτη εξαγωγή (FIFO)
3. Η εξαγωγή σε μια ουρά γίνεται από το εμπρός άκρο της ουράς ενώ η εισαγωγή από το πίσω άκρο
4. Ο δείκτης εμπρός(front) μιας ουράς μας δίνει την θέση του στοιχείου το οποίο στην επόμενη εξαγωγή θα εξαχθεί
5. Το παραπάνω σημαίνει πως στην υλοποίηση ουράς με μονοδιάστατο πίνακα, ο δείκτης front(εμπρός) έχει τιμή πάντα μικρότερη ή ίση του δείκτη rear(πίσω)
6. Αν οι δύο δείκτες της ουράς (front / rear) είναι ίσες, τότε συμβαίνει ένα από τα παρακάτω:
 - a) Αν οι 2 δείκτες είναι 0, τότε η ουρά είναι άδεια
 - b) Αν οι 2 δείκτες είναι ίσοι αλλά διάφοροι του 0, τότε η ουρά περιέχει 1 στοιχείο
7. Αν οι δείκτες εμπρός και πίσω μιας ουράς διαφέρουν κατά μία μονάδα, τότε η ουρά έχει σίγουρα 2 στοιχεία
8. Το τελευταίο στοιχείο που ωθήθηκε σε μια στοίβα το καθορίζει η μεταβλητή top(κορυφή)
9. Σε μια στοίβα, ο δείκτης top δείχνει τη θέση στην οποία τοποθετήθηκε το τελευταίο στοιχείο στη στοίβα(δείχνει στην κορυφή της στοίβας)



ΠΡΑΚΤΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΤΟΙΒΑ

! Για την ώθηση σε στοίβα

```
AN top < N TOTE  
    top <- top + 1  
    STACK[top] <- element
```

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΥΠΕΡΧΕΙΛΙΣΗ, δεν υπάρχει χώρος στη στοίβα'

ΤΕΛΟΣ_ΑΝ

- top (δείκτης που δείχνει την κορυφή της στοίβας – πάντα δείχνει ποιο θα είναι το πρώτο στοιχείο που θα απωθηθεί στο ενδεχόμενο απώθησης)
- N είναι το συνολικό μέγεθος του μονοδιάστατου πίνακα που υλοποιεί τη στοίβα
- STACK είναι το όνομα του πίνακα

Εφόσον λοιπόν η κορυφή της στοίβας ισούται με το συνολικό μέγεθος του πίνακα, δεν υπάρχει ο χώρος για περαιτέρω ώθηση(υπερχείλιση)

! Για την απώθηση από στοίβα

```
AN top > 0 TOTE  
    ΓΡΑΨΕ 'Το στοιχείο που απωθήθηκε είναι το ', STACK[top]  
    top <- top - 1
```

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΥΠΟΧΕΙΛΙΣΗ, δεν υπάρχει στοιχείο για απώθηση'

ΤΕΛΟΣ_ΑΝ

Αν ο δείκτης της κορυφής είναι μεγαλύτερος του 0, τότε εμφανίζουμε το στοιχείο που απωθήθηκε και μειώνουμε τον δείκτη top κατά μία μονάδα (στην πραγματικότητα δεν διαγράφουμε το στοιχείο που ήταν στην κορυφή από τη στοίβα απλά δείχνουμε μέσω του δείκτη top στο από κάτω του στοιχείο)

Σας δίνεται το δεδομένο πως έχετε μια στοίβα με μέγιστη χωρητικότητα 1000 θέσεων($N = 1000$). Γνωρίζετε επίσης πως αυτή η στοίβα **αρχικά δεν είναι άδεια**. Γράψτε τα αντίστοιχα τμήματα εντολών ώστε να επιτυγχάνεται τα παρακάτω:

- Να εμφανίζεται το πλήθος των στοιχείων της στοίβας
- Να ελέγχει αν η στοίβα είναι κενή και να εμφανίζεται σχετικό μήνυμα
- Να διαβάζει μια τιμή από το χρήστη και να πραγματοποιεί ώθηση της στη στοίβα
- Να πραγματοποιεί απώθηση στοιχείου από τη στοίβα
- Να διαβάζει πολλαπλές τιμές από το χρήστη και να τις ωθεί στη στοίβα(το διάβασμα των στοιχείων θα σταματάει όταν ο χρήστης δώσει την τιμή 'ΟΧΙ' ή μέχρι να γεμίσει η στοίβα
- Να πραγματοποιεί συνεχόμενες απωθήσεις από τη στοίβα μέχρι να δώσει ο χρήστης την τιμή 'ΟΧΙ' ή μέχρι να αδειάσει η στοίβα
- Να πραγματοποιεί άδειασμα της στοίβας χωρίς να εμφανίζει τα στοιχεία της

ΕΠΙΛΥΣΗ

Για να εμφανίσουμε το πλήθος των στοιχείων μιας στοίβας, αρκεί να προβάλλουμε τον δείκτη top
ΓΡΑΨΕ top

Για να ελέγχουμε αν μια στοίβα είναι κενή ή όχι(αν περιέχει δηλαδή στοιχεία ή είναι άδεια), αρκεί να κάνουμε έλεγχο για την τιμή του δείκτη top

```
ΑΝ top = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Η στοίβα είναι κενή'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Η στοίβα περιέχει στοιβεία, ', top
ΤΕΛΟΣ_ΑΝ
```

Για την διαδικασία της ώθησης, θα πρέπει πάντα να ελέγχουμε την περίπτωση υπερχείλισης (Αν δηλαδή ο δείκτης top δείχνει στο τέλος του πίνακα) και τότε να πραγματοποιήσουμε ώθηση

```
ΑΝ top < N ΤΟΤΕ
    ΔΙΑΒΑΣΕ user_input
    top <- top + 1
    STACK[top] <- user_input
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Η στοίβα είναι γεμάτη, υπερχείλιση'
ΤΕΛΟΣ_ΑΝ
```

Για την διαδικασία της απώθησης από μια στοίβα, θα πρέπει να ελέγχουμε πάντα την περίπτωση υποχείλισης (Αν ο δείκτης top έχει μηδενική τιμή τότε έχουμε να κάνουμε με άδεια στοίβα και δεν μπορεί να πραγματοποιηθεί απώθηση)

```
ΑΝ top >= 1 ΤΟΤΕ !ελέγχουμε αν η στοίβα δεν είναι άδεια
    ΓΡΑΨΕ 'Το στοιχείο που απωθείται είναι το ', STACK[top]
    top <- top - 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Η στοίβα είναι άδεια, υποχείλιση'
ΤΕΛΟΣ_ΑΝ
```

Για την διαδικασία της πολλαπλής ώθησης στοιχείων

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ top < N ΤΟΤΕ
        ΓΡΑΨΕ 'Δώσε το νέο στοιχείο που θέλεις να εισάγεις...'
        ΔΙΑΒΑΣΕ user_element
        top <- top + 1
        STACK[top] <- user_element
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Υποχείλιση'
    ΤΕΛΟΣ_ΑΝ

    ΓΡΑΨΕ 'Θέλεις εισαγωγή νέου στοιχείου;'
    ΔΙΑΒΑΣΕ user_input
    ΜΕΧΡΙΣ_ΟΤΟΥ user_input = 'ΟΧΙ' Η top = N
```

Για την διαδικασία της πολλαπλής απώθησης στοιχείων

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ top >= 1 ΤΟΤΕ
        ΓΡΑΨΕ 'Γίνεται απώθηση του στοιχείου...'
        ΓΡΑΨΕ STACK[top]
        top <- top - 1
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Υποχείλιση'
    ΤΕΛΟΣ_ΑΝ

    ΓΡΑΨΕ 'Θέλεις απώθηση του επόμενου στοιχείου;'
    ΔΙΑΒΑΣΕ user_input
    ΜΕΧΡΙΣ_ΟΤΟΥ user_input = 'ΟΧΙ' Η top = 0
```

10

12

top = 4

65

3

6

2

15

1



ΑΣΚΗΣΕΙΣ- ΣΤΟΙΒΑ

ΑΣΚΗΣΗ 1

Ένα πλοίο τύπου ferry boat με μια είσοδο/έξοδο έχει χωρητικότητα 150 οχημάτων, που προσέρχονται και επιβιβάζονται. Θα αποβιβαστούν πρώτα αυτά που θα επιβιβαστούν τελευταία. ΝΑΠ σε ΓΛΩΣΣΑ το οποίο:

- a) Να εμφανίζει μενού με τις επιλογές
1. Επιβίβαση 2. Αποβίβαση 3. Έξοδος
- b) Στην περίπτωση που επιλεγθεί Επιβίβαση, να ζητά εισαγωγή του αριθμού κυκλοφορίας κάθε οχήματος και να τον καταχωρίζει σε στοίβα. Στην συνέχεια θα πρέπει να τυπώνεται το ερώτημα “Υπάρχει άλλο όχημα για επιβίβαση;”. Αν ο χρήστης απαντήσει ΝΑΙ, τότε να επαναλαμβάνεται η διαδικασία επιβίβασης και να εμφανίζεται το μενού
- c) Στην περίπτωση που επιλεγθεί η Αποβίβαση, να τυπώνει τους αριθμούς κυκλοφορίας όλων των οχημάτων με τη σειρά με την οποία αποβιβάζονται από το πλοίο

```

1 ΠΡΟΓΡΑΜΜΑ FERRY
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: top, επιλογή
4   ΧΑΡΑΚΤΗΡΕΣ: ΣΤΟΙΒΑ[150], αρ_κυκ, απ
5 ΑΡΧΗ
6   top <- 0
7   ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
8     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
9       ΓΡΑΨΕ '1. Επιβίβαση'
10      ΓΡΑΨΕ '2. Αποβίβαση'
11      ΓΡΑΨΕ '3. Έξοδος'
12      ΓΡΑΨΕ 'Δώσε την επιλογή σου'
13      ΔΙΑΒΑΣΕ επιλογή
14      ΜΕΧΡΙΣ_ΟΤΟΥ (επιλογή>=1) ΚΑΙ (επιλογή<=3)
15
16      ΑΝ επιλογή = 1 ΤΟΤΕ
17
18        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
19          ΑΝ top<150 ΤΟΤΕ
20            ΓΡΑΨΕ 'Δώσε αριθμό κυκλ για όχημα'
21            ΔΙΑΒΑΣΕ αρ_κυκ
22            top <- top + 1
23            ΣΤΟΙΒΑ[top] <- αρ_κυκ
24          ΤΕΛΟΣ_ΑΝ
25
26          ΑΝ top=150 ΤΟΤΕ
27            ΓΡΑΨΕ 'Το πλοίο έχει γεμίσει'
28          ΤΕΛΟΣ_ΑΝ
29
30          ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
31            ΓΡΑΨΕ 'Υπάρχει άλλο αμάξι για επιβίβαση;'
32            ΔΙΑΒΑΣΕ απ
33            ΜΕΧΡΙΣ_ΟΤΟΥ (απ='ΝΑΙ') Η (απ='ΟΧΙ')
34            ΜΕΧΡΙΣ_ΟΤΟΥ (top=150) Η (απ='ΟΧΙ')
35
36          ΑΛΛΙΩΣ_ΑΝ επιλογή = 2 ΤΟΤΕ
37            ΟΣΟ top >= 1 ΕΠΑΝΑΛΑΒΕ
38              ΓΡΑΨΕ 'Αποβιβάζεται το όχημα ', ΣΤΟΙΒΑ[top]
39              top <- top - 1
40            ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
41          ΤΕΛΟΣ_ΑΝ
42
43      ΜΕΧΡΙΣ_ΟΤΟΥ επιλογή=3
44
45      ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```





ΠΡΑΚΤΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ - ΟΥΡΑ

```

! Εισαγωγή στοιχείου σε ουρά QUEUE
! N είναι το μέγεθος του πίνακα που υλοποιεί την ουρά
AN rear = N TOTE ! Σημαίνει πως δεν μπορεί να εισαχθεί νέο στοιχείο
    ΓΡΑΨΕ 'Γεμάτη ουρά'
ΑΛΛΙΩΣ
    AN front = 0 ΚΑΙ rear = 0 TOTE ! Σημαίνει πως η ουρά είναι άδεια και θα εισαχθεί το πρώτο στοιχείο
        front <- 1 ! ΣΕ ΑΥΤΗ την περίπτωση θα πρέπει ο δείκτης front να δείχνει την τιμή 1
    ΤΕΛΟΣ_ΑΝ
    rear <- rear + 1
    QUEUE[rear] <- element
ΤΕΛΟΣ_ΑΝ

```

Η κωδικοποίηση για την **ΕΙΣΑΓΩΓΗ** ενός νέου στοιχείου σε ουρά QUEUE. Υπάρχουν 2 δείκτες:

- front που δείχνει ποιο στοιχείο είναι αυτό που θα εξαχθεί (με την μέθοδο FIFO)
- rear που δείχνει το τελευταίο στοιχείο της ουράς

Προσοχή:

1. Αν ο δείκτης rear(πίσω) έχει φτάσει στο τέλος του πίνακα, τότε σημαίνει πως η ουρά είναι γεμάτη και δεν μπορεί να εισαχθεί κάποιο νέο στοιχείο
2. Σε περίπτωση που μπορεί να εισαχθεί ένα στοιχείο, θα πρέπει να ελέγξουμε αν πρόκειται για το πρώτο στοιχείο που θα εισαχθεί στην ουρά(σε αυτήν και μόνο την περίπτωση θα πρέπει να θέσουμε τον δείκτη front στην τιμή 1 για να δείχνει σε αυτό το πρώτο στοιχείο για ενδεχόμενη εξαγωγή)
3. Σε κάθε περίπτωση όπου η ουρά δεν είναι γεμάτη και η ώθηση μπορεί να γίνει, ο δείκτης rear θα πρέπει να αυξηθεί κατά 1 και σε αυτή τη θέση να εισαχθεί το νέο στοιχείο

```

! Εξαγωγή στοιχείου από την ουρά QUEUE
AN front=0 ΚΑΙ rear=0 TOTE
    ΓΡΑΨΕ 'Η ουρά είναι άδεια'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Γίνεται εξαγωγή στοιχείου από ουρά...'
    ΓΡΑΨΕ QUEUE[front]
    AN front=rear TOTE ! Αυτό σημαίνει πως εξάγαμε το τελευταίο στοιχείο της ουράς
        front=0
        rear=0
    ΑΛΛΙΩΣ
        front <- front + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

```

Η κωδικοποίηση για την **ΕΞΑΓΩΓΗ** ενός στοιχείου από ουρά

Προσοχή:

1. Αν οι δείκτες front και rear είναι 0, σημαίνει πως η ουρά είναι άδεια και δεν υπάρχει στοιχείο για να γίνει εξαγωγή
2. Σε περίπτωση που το στοιχείο που εξάγουμε είναι το τελευταίο, τότε προκύπτει μια άδεια ουρά και μηδενίζουμε τους δείκτες front και rear για να το υποδηλώσουμε αυτό
3. Αν δεν ήταν το τελευταίο στοιχείο αυτό που εξάγουμε, τότε απλά αυξάνουμε τον δείκτη front ώστε αυτός να δείχνει το επόμενο στοιχείο για εξαγωγή



ΑΣΚΗΣΕΙΣ- ΟΥΡΑ

ΑΣΚΗΣΗ

Σε μια δημόσια υπηρεσία υπάρχει μια σειρά για την εξυπηρέτηση των φορολογούμενων. Η σειρά εξυπηρέτησης έχει μέγιστη χωρητικότητα 300 ατόμων, τα οποία εισέρχονται και εξέρχονται. Στην σειρά εξυπηρετούνται πρώτα οι φορολογούμενοι που εισήλθαν πρώτοι. Η διαδικασία αυτή θα πρέπει να επιβλέπεται από πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο:

1. Θα εμφανίζει μενού με τις επιλογές ΕΙΣΑΓΩΓΗ | ΑΠΟΧΩΡΗΣΗ | ΑΠΟΧΩΡΗΣΗ_ΟΛΩΝ | ΕΞΟΔΟΣ
2. Στην περίπτωση που επιλεγθεί ΕΙΣΑΓΩΓΗ, θα πρέπει να διαβάζεται ο Α.Δ.Τ του φορολογούμενου και να καταχωρίζεται στην ουρά. Αφού γίνει η εισαγωγή θα πρέπει να ερωτάται ο χρήστης αν υπάρχει άλλος φορολογούμενος για εισαγωγή και η διαδικασία θα επαναλαμβάνεται αν ο χρήστης απαντήσει “ΝΑΙ”, ενώ αν απαντήσει “ΟΧΙ” τότε η διαδικασία θα σταματά και η ροή του προγράμματος θα επιστρέφει στο αρχικό μενού
3. Αν επιλεγθεί ΑΠΟΧΩΡΗΣΗ, το πρόγραμμα θα εμφανίζει τον Α.Δ.Τ του φορολογουμένου που εξυπηρετήθηκε. Στην συνέχεια ο χρήστης θα ερωτάται αν επιθυμεί νέα αποχώρηση και η διαδικασία θα επαναλαμβάνεται αν ο χρήστης επιλέξει “ΝΑΙ” ενώ σε περίπτωση που επιλέξει “ΟΧΙ” η ροή του προγράμματος θα επιστρέφει στο αρχικό μενού
4. Αν επιλεγθεί ΑΠΟΧΩΡΗΣΗ ΟΛΩΝ τότε το πρόγραμμα θα εξάγει από την ουρά και θα εμφανίζει όλους τους Α.Δ.Τ των φορολογούμενων που εξυπηρετήθηκαν

ΕΠΙΛΥΣΗ



```
ΠΡΟΓΡΑΜΜΑ ΦΟΡΟΛΟΓΟΥΜΕΝΟΙ_ΟΥΡΑ
ΣΤΑΘΕΡΕΣ
    N = 300
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: user_choice, user_input, id, QUEUE[N]
    ΑΚΕΡΑΙΕΣ: front, rear
ΑΡΧΗ
    ΓΡΑΨΕ 'Αρχικοποίηση άδειας ουράς'
    front <- 0
    rear <- 0
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΓΡΑΨΕ 'ΕΠΙΛΕΞΕ ΤΙ ΘΕΣ ΝΑ ΚΑΝΕΙΣ'
            ΓΡΑΨΕ 'ΠΑΤΑ 1 ΓΙΑ ΕΙΣΑΓΩΓΗ ΦΟΡΟΛΟΓΟΥΜΕΝΟΥ'
            ΓΡΑΨΕ 'ΠΑΤΑ 2 ΓΙΑ ΑΠΟΧΩΡΗΣΗ ΦΟΡΟΛΟΓΟΥΜΕΝΟΥ'
            ΓΡΑΨΕ 'ΠΑΤΑ 3 ΓΙΑ ΑΠΟΧΩΡΗΣΗ ΟΛΩΝ ΤΩΝ ΦΟΡΟΛΟΓΟΥΜΕΝΩΝ'
            ΓΡΑΨΕ 'ΠΑΤΑ 4 ΓΙΑ ΕΞΟΔΟ'
            ΔΙΑΒΑΣΕ user_choice
            ΜΕΧΡΙΣ_ΟΤΟΥ user_choice >= '1' ΚΑΙ user_choice <= '4'
            ΕΠΙΛΕΞΕ user_choice
            ΠΕΡΙΠΤΩΣΗ '1'
                ΑΝ rear = N ΤΟΤΕ
                    ΓΡΑΨΕ 'Η ουρά εξυπηρέτησης είναι πλήρης'
                ΑΛΛΙΩΣ
                    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
                        ΓΡΑΨΕ 'Δώσε Α.Δ.Τ φορολογούμενου για εισαγωγή στην ουρά'
                        ΔΙΑΒΑΣΕ id
                        ΑΝ front=0 ΚΑΙ rear=0 ΤΟΤΕ
                            front <- 1
                        ΤΕΛΟΣ_ΑΝ
                        rear <- rear + 1
                        QUEUE[rear] <- id
                        ΓΡΑΨΕ 'Η εισαγωγή φορολογούμενου πραγματοποιήθηκε!'
                        ΓΡΑΨΕ 'Επιθυμείς νέα εισαγωγή - ΝΑΙ Η ΟΧΙ;'
                        ΔΙΑΒΑΣΕ user_input
                        ΜΕΧΡΙΣ_ΟΤΟΥ (user_input='ΟΧΙ' Η rear=N)
                    ΤΕΛΟΣ_ΑΝ
                ΠΕΡΙΠΤΩΣΗ '2'
                    ΑΝ (front=0 ΚΑΙ rear=0) ΤΟΤΕ
                        ΓΡΑΨΕ 'Η ουρά εξυπηρέτησης είναι άδεια'
                    ΑΛΛΙΩΣ
                        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
                            ΓΡΑΨΕ 'Ο φορολογούμενος που εξυπηρετήθηκε ήταν ο ', QUEUE[front]
                            ΑΝ front = rear ΤΟΤΕ
                                front <- 0
                                rear <- 0
                            ΑΛΛΙΩΣ
                                front <- front + 1
                            ΤΕΛΟΣ_ΑΝ
                            ΓΡΑΨΕ 'Επιθυμείς νέα εξυπηρέτηση; Πάτα ΟΧΙ για έξοδο σε μενού'
                            ΔΙΑΒΑΣΕ user_input
                            ΜΕΧΡΙΣ_ΟΤΟΥ user_input='ΟΧΙ'
                        ΤΕΛΟΣ_ΑΝ
                    ΠΕΡΙΠΤΩΣΗ '3'
                        ΑΝ (front=0 ΚΑΙ rear=0) ΤΟΤΕ
                            ΓΡΑΨΕ 'Η ουρά εξυπηρέτησης είναι άδεια'
                        ΑΛΛΙΩΣ
                            ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
                                ΓΡΑΨΕ 'Ο φορολογούμενος που εξυπηρετήθηκε ήταν ο ', QUEUE[front]
                                front <- front + 1
                            ΜΕΧΡΙΣ_ΟΤΟΥ front > rear
                            ΓΡΑΨΕ 'Όλοι οι φορολογούμενοι εξυπηρετήθηκαν'
                            front <- 0
                            rear <- 0
                        ΤΕΛΟΣ_ΑΝ
                    ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
                ΜΕΧΡΙΣ_ΟΤΟΥ user_choice='4'
                ΓΡΑΨΕ 'ΣΕ ΕΥΧΑΡΙΣΤΟΥΜΕ'
            ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```


ΑΣΚΗΣΗ

Κάθε εκτυπωτής χρησιμοποιεί μια ουρά εκτύπωσης για να τοποθετεί σε αυτήν τα αρχεία που λαμβάνει για εκτύπωση με τη σειρά με την οποία στάλθηκαν. Κάθε φορά εκτυπώνεται το αρχείο που βρίσκεται στην αρχή της ουράς εκτύπωσης. Λόγω της περιορισμένης μνήμης του εκτυπωτή θεωρούμε ότι στην ουρά μπορούν να εισαχθούν το πολύ 20 αρχεία. Να αναπτύξετε πρόγραμμα το οποίο :

- 1) Διαβάζει την επιλογή του χρήστη, που θα πρέπει να είναι “N” ή “P” ή “E”. Αν επιλεγθεί το “N” θα ζητάει το πρόγραμμα από το χρήστη το όνομα του αρχείου και θα το εισάγει στην ουρά εκτύπωσης(εφόσον υπάρχει χώρος), εμφανίζοντας κατάλληλο μήνυμα. Σε περίπτωση που δεν υπάρχει χώρος τότε θα εμφανίζει αντίστοιχα κάποιο κατάλληλο μήνυμα.
- 2) Αν ο χρήστης δώσει την επιλογή “P” θα εξετάζεται αν υπάρχει αρχείο προς εκτύπωση και εφόσον υπάρχει θα εμφανίζεται το μήνυμα “Printed” μαζί με το όνομα του αρχείου. Αν δεν υπάρχει αρχείο προς εκτύπωση, τότε θα εμφανίζεται ένα κατάλληλο μήνυμα
- 3) Η διαδικασία αυτή θα ολοκληρώνεται αν ο χρήστης δώσει την επιλογή “E” ή αν αδειάσει η ουρά εκτύπωσης και αφού συμβεί αυτό, θα εκτυπώνονται στην οθόνη ο αριθμός των αρχείων που εκτυπώθηκαν καθώς και ο αριθμός των φορών που η ουρά εκτύπωσης είχε μόνο 1 αρχείο

ΘΕΩΡΗΣΤΕ ΟΤΙ ΣΤΑΛΘΗΚΕ ΤΟΥΛΑΧΙΣΤΟΝ 1 ΑΡΧΕΙΟ

ΕΠΙΛΥΣΗ

Η εκφώνηση της άσκησης μας ενημερώνει πως στάλθηκε τουλάχιστον 1 αρχείο καθώς την πρώτη φορά που φορτώνεται το πρόγραμμα η ουρά εκτύπωσης είναι άδεια και στο μενού επιλογών του χρήστη αν αυτός επιλέξει “P” τότε το πρόγραμμα θα τερματίζει καθώς αρχικά δεν υπάρχει αρχείο στην ουρά εκτύπωσης. Πρακτικά η εκφώνηση αναφέρει πως την πρώτη φορά ο χρήστης εννοείται πως θα πατήσει “N” για εισαγωγή στην ουρά εκτύπωσης ενός αρχείου

```
ΠΡΟΓΡΑΜΜΑ ΦΟΡΟΛΟΓΟΥΜΕΝΟΙ_ΟΥΡΑ
ΣΤΑΘΕΡΕΣ
    N = 20
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: user_input, QUEUE[N], filename
    ΑΚΕΡΑΙΕΣ: front, rear, counter_printed, counter_singleFile
ΑΡΧΗ
    ΓΡΑΨΕ 'Αρχικοποίηση ουράς εκτυπωτή'
    front <- 0
    rear <- 0
    counter_singleFile <- 0
    counter_printed <- 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ 'Τι θέλεις να κάνεις...'
        ΓΡΑΨΕ 'N για λήψη νέου αρχείου'
        ΓΡΑΨΕ 'P για εκτύπωση πρόσφατου αρχείου'
        ΔΙΑΒΑΣΕ user_input
        ΜΕΧΡΙΣ_ΟΤΟΥ user_input = 'N' Η user_input = 'P' Η user_input = 'E'

        ΑΝ user_input = 'N' ΤΟΤΕ
            ΑΝ rear = N ΤΟΤΕ
                ΓΡΑΨΕ 'Δεν υπάρχει χώρος στην ουρά'
            ΑΛΛΙΩΣ
                ΓΡΑΨΕ 'Δώσε το όνομα του αρχείου για εισαγωγή'
                ΔΙΑΒΑΣΕ filename
                ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
                    ΓΡΑΨΕ 'Εισαγωγή πρώτου αρχείου στην ουρά'
                    front <- 1
                ΤΕΛΟΣ_ΑΝ
                rear <- rear + 1
                QUEUE[rear] <- filename
                ΓΡΑΨΕ 'Η εισαγωγή του αρχείου ολοκληρώθηκε!'
            ΤΕΛΟΣ_ΑΝ
            ΑΛΛΙΩΣ_ΑΝ user_input = 'P' ΤΟΤΕ
                ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
                    ΓΡΑΨΕ 'Δεν υπάρχει αρχείο για εκτύπωση'
                ΑΛΛΙΩΣ
                    ΓΡΑΨΕ 'Filename ', QUEUE[front], ' printed'
                    counter_printed <- counter_printed + 1
                    ΑΝ front = rear ΤΟΤΕ
                        ΓΡΑΨΕ 'Η ουρά εκτύπωσης μόλις άδειασε'
                        front <- 0
                        rear <- 0
                    ΑΛΛΙΩΣ
                        front <- front + 1
                    ΤΕΛΟΣ_ΑΝ
                ΤΕΛΟΣ_ΑΝ
            ΑΛΛΙΩΣ_ΑΝ user_input = 'E' ΤΟΤΕ
                ΓΡΑΨΕ 'Σε ευχαριστούμε!'
            ΑΛΛΙΩΣ
                ΓΡΑΨΕ 'Δώσε N, P ή E...'
            ΤΕΛΟΣ_ΑΝ
            ΑΝ (front = rear) ΚΑΙ front <> 0 ΤΟΤΕ
                counter_singleFile <- counter_singleFile + 1
            ΤΕΛΟΣ_ΑΝ
        ΜΕΧΡΙΣ_ΟΤΟΥ (user_input = 'E') Η (front = 0 ΚΑΙ rear = 0)

        ΓΡΑΨΕ 'Συνολικά εκτυπώθηκαν ', counter_printed, ' αρχεία'
        ΓΡΑΨΕ 'Η ουρά εκτύπωσης περιείχε 1 αρχείο ', counter_singleFile, ' φορές'
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```