

19. Να ορίσετε την δομή δεδομένων στοίβα. Ποιά λειτουργία επιτελεί η μέθοδος επεξεργασίας LIFO.

**Στοίβα** (stack) ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιον τρόπο, ώστε τα στοιχεία που βρίσκονται στη κορυφή της στοίβας να λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας να λαμβάνονται τελευταία, όπως μια στοίβα πιάτων. Η στοίβα χρησιμοποιεί τη μέθοδο επεξεργασίας "τελευταίο μέσα, πρώτο έξω" LIFO (Last In, First Out), δηλαδή το στοιχείο που εισέρχεται τελευταίο εξέρχεται πρώτο.

Για την υλοποίηση της στοίβας με μονοδιάστατο πίνακα χρησιμοποιείται ένας δείκτης που δείχνει την θέση του τελευταίου στοιχείου που έχει εισαχθεί (κορυφή -top) αλλά και τη θέση του πρώτου στοιχείου προς απώθηση.

20. Να περιγράψετε τις κύριες λειτουργίες μιας στοίβας.

Οι κύριες λειτουργίες είναι η **ώθηση** στοιχείου στην κορυφή της στοίβας και η **απώθηση** στοιχείου από την κορυφή της στοίβας. Κατά την ώθηση πρέπει να ελέγχεται αν υπάρχει χώρος για εισαγωγή νέου στοιχείου (έλεγχος υπερχειλίσης) και κατά την απώθηση αν υπάρχει τουλάχιστον ένα τουλάχιστον στοιχείο για διαγραφή (έλεγχος υποχειλίσης). Ο λόγος εμφάνισης υπερχειλίσης ή υποχειλίσης είναι ότι για την υλοποίηση της δομής της στοίβας στη ΓΛΩΣΣΑ χρησιμοποιείται μονοδιάστατος πίνακας, που είναι στατική δομή και έχει σταθερό μέγεθος.

21. Να ορίσετε την δομή δεδομένων ουρά. Ποιά λειτουργία επιτελεί η μέθοδος επεξεργασίας FIFO.

**Ουρά** (queue) ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιον τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα. Η ουρά λειτουργεί όπως μια ουρά ανθρώπων σε μια τράπεζα. Χρησιμοποιεί την τεχνική "πρώτο μέσα, πρώτο έξω" FIFO (First In, First Out), δηλαδή το στοιχείο που εισέρχεται πρώτο εξέρχεται πρώτο.

Για την υλοποίηση της ουράς με μονοδιάστατο πίνακα χρησιμοποιούνται δυο δείκτες ο εμπρός (front) και ο πίσω (rear) που δείχνουν αντίστοιχα την θέση του πρώτου στοιχείου που θα εξαχθεί και την θέση του τελευταίου στοιχείου που έχει εισαχθεί.

22. Να περιγράψετε τις κύριες λειτουργίες μιας ουράς.

Οι κύριες λειτουργίες είναι η **εισαγωγή** στοιχείου στο πίσω άκρο της ουράς και η **εξαγωγή** στοιχείου από το μπροστινό άκρο της ουράς. Για την υλοποίηση της δομής της ουράς στη ΓΛΩΣΣΑ χρησιμοποιείται μονοδιάστατος πίνακας, που είναι στατική δομή και έχει σταθερό μέγεθος. Κατά την εισαγωγή πρέπει να ελέγχεται αν υπάρχει ελεύθερος χώρος στον πίνακα (έλεγχος για το εάν η ουρά είναι γεμάτη) και κατά την εξαγωγή αν υπάρχει τουλάχιστον ένα τουλάχιστον στοιχείο στην ουρά (έλεγχος για το εάν η ουρά είναι άδεια).

### ΘΕΜΑ Γ

Στο κρεοπωλείο ενός supermarket, πωλούνται μόνο δύο είδη κρέατος, χοιρινό ( αξίας 6,5 / κιλό ) και μοσχάρι ( αξίας 9.5 € / κιλό ). Οι πελάτες εξυπηρετούνται με βάση τη σειρά άφιξής τους αυστηρά, από έναν μόνο υπάλληλο. Σημειώνεται ότι η ουρά αναμονής δεν μπορεί να ξεπερνά τα 150 άτομα (μπορούν να εξυπηρετηθούν το πολύ 150 άτομα σε μία ημέρα).

Καλείσθε να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ στο οποίο:

Γ1 Υπάρχει κατάλληλο τμήμα δηλώσεων.

(Μονάδες 2)

Για κάθε πελάτη του κρεοπωλείου:

Γ2.α Να γίνεται δεκτή ως είσοδος από το χρήστη η επιλογή με αποδεκτές τιμές 1. ΕΙΣΑΓΩΓΗ, για είσοδο πελάτη στην ουρά αναμονής και 2. ΕΠΟΜΕΝΟΣ για εξυπηρέτηση ενός πελάτη.

β Αν δοθεί για επιλογή η τιμή 1 τότε να εισάγεται σε μία ουρά ο αύξων αριθμός του πελάτη. Κατόπιν να εμφανίζεται το πλήθος των ατόμων που περιμένουν στην ουρά πριν από αυτόν. Στην περίπτωση που η ουρά είναι γεμάτη να εμφανίζεται κατάλληλο μήνυμα.

γ Αν δοθεί η τιμή 2 τότε να εμφανίζεται το μήνυμα «Εξυπηρέτηση» και δίπλα ο αύξων αριθμός του προς εξυπηρέτηση πελάτη. Κατόπιν να ζητείται το είδος του κρέατος «χ» για χοιρινό και «μ» για μοσχάρι και η αντίστοιχη ποσότητα.

(Μονάδες 6)

Γ3. Η παραπάνω διαδικασία να επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλοι οι πελάτες που εισήχθησαν στην ουρά αναμονής.

(Μονάδες 4)

Στο τέλος της ημέρας να εμφανίζονται τα εξής στατιστικά στοιχεία:

Γ4. α Τα ποσοστά των πελατών που αγόρασε χοιρινό κρέας

(Μονάδες 4)

β Τα συνολικά έσοδα του κρεοπωλείου

(Μονάδες 4)

### ΠΑΡΑΤΗΡΗΣΕΙΣ

Θεωρείστε ότι τουλάχιστον ένας πελάτης εισήχθη στην ουρά

Δεν απαιτείται έλεγχος εγκυρότητας.

Θεωρείστε ότι κάθε πελάτης αγοράζει μόνο ένα είδος κρέατος.

### ΘΕΜΑ Γ

ΠΡΟΓΡΑΜΜΑ ΚΡΕΟΠΩΛΕΙΟ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ΕΜΠΡΟΣ, ΠΙΣΩ, ΕΠΙΛΟΓΗ, ΠΕΡΙΜΕΝΟΥΝ, ΟΥΡΑ[250], ΠΛ, & ΠΛΧ

ΠΡΑΓΜΑΤΙΚΕΣ: ΠΟΣΟΣΤΟ,Σ

ΧΑΡΑΚΤΗΡΕΣ: ΕΙΔΟΣ

ΛΟΓΙΚΕΣ: ΕΞΥΠΗΡΕΤΗΘΗΚΑΝ\_ΟΛΟΙ

ΑΡΧΗ

ΕΞΥΠΗΡΕΤΗΘΗΚΑΝ\_ΟΛΟΙ ← ΨΕΥΔΗΣ

ΠΛΧ ← 0

ΠΛ ← 0

ΕΜΠΡΟΣ ← 0

ΠΙΣΩ ← 0

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'ΔΩΣΕ 1 ΓΙΑ ΕΙΔΑΓΩΓΗ 2 ΓΙΑ ΕΠΟΜΕΝΟΣ'

ΔΙΑΒΑΣΕ ΕΠΙΛΟΓΗ

ΑΝ ΕΠΙΛΟΓΗ = 1 ΤΟΤΕ

ΑΝ ΠΙΣΩ < 200 ΤΟΤΕ

ΑΝ ΕΜΠΡΟΣ = 0 ΚΑΙ ΠΙΣΩ = 0 ΤΟΤΕ

ΕΜΠΡΟΣ ← 1

ΠΙΣΩ ← 1

ΠΕΡΙΜΕΝΟΥΝ ← 0

ΑΛΛΙΩΣ

ΠΙΣΩ ← ΠΙΣΩ + 1

ΠΕΡΙΜΕΝΟΥΝ ← ΠΙΣΩ - ΕΜΠΡΟΣ

ΤΕΛΟΣ\_ΑΝ

ΟΥΡΑ[ΠΙΣΩ] ← ΠΙΣΩ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΟΥΡΑ ΓΕΜΑΤΗ'

ΤΕΛΟΣ\_ΑΝ

ΑΛΛΙΩΣ\_ΑΝ ΕΠΙΛΟΓΗ = 2 ΤΟΤΕ

ΑΝ ΕΜΠΡΟΣ <= ΠΙΣΩ ΤΟΤΕ

ΓΡΑΨΕ 'ΕΞΥΠΗΡΕΤΗΣΗ',ΟΥΡΑ[ΕΜΠΡΟΣ]

ΓΡΑΨΕ 'ΔΩΣΕ ΕΙΔΟΣ Χ Ή Μ ΚΑΙ ΠΟΣΟΤΗΤΑ'

ΔΙΑΒΑΣΕ ΕΙΔΟΣ, ΠΟΣΟΤΗΤΑ

ΑΝ ΕΙΔΟΣ = 'Χ' ΤΟΤΕ

ΠΛΧ ← ΠΛΧ + 1

Σ ← Σ + 6.5\*ΠΟΣΟΤΗΤΑ

ΑΛΛΙΩΣ

Σ ← Σ + 9.5\*ΠΟΣΟΤΗΤΑ

ΤΕΛΟΣ\_ΑΝ

ΕΜΠΡΟΣ ← ΕΜΠΡΟΣ + 1

ΑΛΛΙΩΣ

ΕΞΥΠΗΡΕΤΗΘΗΚΑΝ\_ΟΛΟΙ ← ΑΛΗΘΗΣ

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΑΝ

ΜΕΧΡΙΣ\_ΟΤΟΥ ΕΞΥΠΗΡΕΤΗΘΗΚΑΝ\_ΟΛΟΙ = ΑΛΗΘΗΣ

ΠΟΣΟΣΤΟ ← ΠΛΧ / ΠΛ \* 100

ΓΡΑΨΕ ΠΟΣΟΣΤΟ, Σ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

Αρχή

flag ← yes  
nλx ← 0  
Σ ← 0  
nλx ← 0  
rear ← 0  
front ← 0

Αρχή - ερωτήσεις

frage '1. ΕΙΣΑΓΩΓΗ'  
frage '2. ΕΞΥΝΗΡΕΤΗΣΗ'  
frage 'Δώστε ερωτημα 1..2'

Διάγραμμα Ερωτημα  
Αν ερωτημα = 1 τότε

Αν rear = LSO τότε  
frage 'ΟΥΡΑ ΚΕΜΑΤΗ'  
Αρχή  
rear ← rear + 1  
Αν front = 0 και rear = 0 τότε  
front ← 1  
rear ← 1  
Αναλογια ← 0  
Αρχή  
Αναλογια ← rear - front  
rear ← rear + 1  
τέλος  
ΟΥΡΑ[rear] ← rear  
frage 'Απόψε σου Αποφώνι; Αποφώνι  
τέλος

Αρχή

Αν front = 0 και rear = 0 τότε  
frage 'ΟΥΡΑ ΑΔΕΙΑ'  
flag ← Αρχή  
Αρχή  
frage 'εμφανισμι:' ΟΥΡΑ[front]  
frage 'Δώστε x για κορίτσι ή y για ποδαρικό'  
Διάγραμμα ΕΙΣΟΣ  
frage 'Δώστε ποσοστά'  
Διάγραμμα ΠΟΣΟΤ  
Αν εισος = x' τότε  
Σ ← Σ + ποσοτ \* 6,5  
nλx ← nλx + 1  
Αρχή  
Σ ← Σ + ποσοτ \* 9,5  
τέλος  
Αν front = rear τότε  
front ← 0  
rear ← 0  
Αρχή  
front ← front + 1  
τέλος

Μεταβλητές flag = Αρχή

noz ← nλx \* 100 / nλx  
frage 'Ποσοτό νεφταίνου του νηπου κοριπνο; noz %'

frage 'ΕΣΟΔΑ: Σ

τέλος

Πρόγραμμα ΘΕΜΑΤ

Αριθμοί: n, NATIO, nNAT, nIKA, AE, en  
 Πρωταρχειές: ΣOAE, ΣIKA, ΣNAT, Total, Σ, nOZ, En-NAT, En-IKA, En-OAE  
 Χαρακτήρες: ON, τύπος

Αρχή

nNATIO ← 0  
 nNATE ← 0  
 nIKA ← 0  
 ΣOAE ← 0  
 ΣIKA ← 0  
 ΣNAT ← 0  
 Total ← 0

Διάγραμμα ON

Όσο ON < ΤΥΠΟΣ επαναλάβετε

Αρχή επανάληψης

Διάγραμμα ΤΥΠΟΣ

Με βάση του τύπος = NAT τότε ΤΥΠΟΣ = OAE Τύπος = IKA  
 Διάγραμμα AE

Σ ← 0

Για en από 1 μέχρι AE

Διάγραμμα ΠΟΣΟ

Σ ← Σ + ΠΟΣΟ

ΤΕ

Total ← Total + Σ

Αν τύπος = NAT τότε

ΣNAN ← ΣNAT + Σ

Αφού τύπος = IKA' εν

ΣIKA ← ΣIKA + Σ

nIKA ← nIKA + 1

Αφού

ΣOAE ← ΣOAE + Σ

τέλος w

Διάγραμμα ON

Τέλος επανάληψης  
 Γραμμή Total

En-NAT ← ΣNAT \* 0,8

En-IKA ← ΣIKA \* 0,75

En-OAE ← ΣOAE \* 0,7

Γραμμή En-NAT, En-IKA, En-OAE

Γραμμή "Από IKA": nIKA

ΚΑΤΕΣΤ ΕΜΒΛΗΜΑΤΗ-ΠΟΣ (nNAT, nNATIO)

Τέλος Προγράμματος

Διωνυμικά ΕΜΒΛΗΜΑΤΗ-ΠΟΣ (n1, n2)

Μειωθέντες

Αριθμοί: n1, n2

Πρωταρχειές: ΠΟΣ

Αρχή

Αν n1 < 0 τότε

ΠΟΣ ← n2 \* 100 / n1

Γραμμή ΠΟΣ

Αφού

Γραμμή "Δωμικη κωδικοποίηση" + n1 με 10 ενσωμάτωση

τέλος w

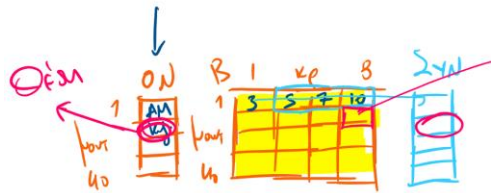
τέλος Διωνυμικών

ΘΕΜΑΤ

Αρχή

Για hour ως 1 ή/και 40  
 Δύο ON (hour)  
 Για xp ως 1 ή/και 8  
 Αρχή-φράση  
 Δύο B (hour, xp)  
 $\{B(hour, xp) \geq 1 \text{ και } B(hour, xp) \leq 10\}$

Επίσκεψη Δ2  
 Για hour ως 1 ή/και 40  
 Για i ως 2 ή/και 8  
 Για j ως 8 ή/και i ή/και 8-1  
 Αν  $B(hour, j) > B(hour, j-1)$  τότε  
 $temp \leftarrow B(hour, j)$   
 $B(hour, j) \leftarrow B(hour, j-1)$   
 $B(hour, j-1) \leftarrow temp$   
 τέλος



Επίσκεψη Δ3  
 Για hour ως 1 ή/και 40  
 $SYN(hour) \leftarrow 0$   
 Για xp ως 1 ή/και 8  
 $SYN(hour) \leftarrow SYN(hour) + B(hour, xp)$

Επίσκεψη Δ4  
 Δύο Name  
 $\Theta \leftarrow ANAZ(Name, ON)$   
 Αν  $\Theta = 0$  τότε  
 φράση 'Δεν βρέθηκε το όνομα'  
 Αρχή  
 φράση  $SYN(\Theta), B(\Theta, 8)$

**ΘΕΜΑ Α**

Σύνοψη

Συνάρτηση ANAZ (key, ονοματα): Αρχική  
 Ημερομηνία  
 Αρχή pos, i  
 Τοποθεσίες: key, ονοματα  
 Ημερομηνία: flag

Αρχή

flag ← γράμμα  
 pos ← 0  
 i ← 1  
 Όσο i ≤ 40 και flag = γράμμα τότε  
 Αν ονοματα[i] = key τότε  
 key ← A[i] pos  
 pos ← i  
 Αρχή  
 i ← i + 1  
 τέλος  
 $ANAZ \leftarrow pos$   
 Τύπος-ήλιος