

## ΗΥ240: Δομές Δεδομένων

### Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-2019

### Προγραμματιστική Εργασία - 2ο Μέρος

**Ημερομηνία Παράδοσης:** Δευτέρα, 13 Μαΐου 2019, ώρα 23:59

**Τρόπος Παράδοσης:** Χρησιμοποιώντας το πρόγραμμα turnin. Πληροφορίες για το πώς λειτουργεί το πρόγραμμα turnin παρέχονται στην ιστοσελίδα του μαθήματος.



(Source <http://www.clevernotes.ie>)

#### Γενική περιγραφή:

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που θα προσομοιώνει καταστήματα αγορών και καταναλωτές. Τα καταστήματα κατηγοριοποιούνται σύμφωνα με τον τύπο τους. Στα πλαίσια της εργασίας υπάρχουν πέντε κατηγορίες. Επιπλέον διαθέτουν ένα σύνολο προϊόντων διαθέσιμα προς τους καταναλωτές. Κάθε καταναλωτής διαθέτει μια λίστα με προϊόντα που επιθυμεί να αγοράσει. Ένας καταναλωτής μπορεί να αγοράσει ένα ή περισσότερα προϊόντα από αυτά που συμπεριλαμβάνονται στην προσωπική του λίστα από τα καταστήματα που τα διαθέτουν.

#### Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Τα καταστήματα είναι οργανωμένα σε ένα **δυναμικό δένδρο αναζήτησης με κόμβο φρουρό**, το οποίο ονομάζεται δένδρο καταστημάτων. Το δένδρο είναι ταξινομημένο με βάση το αναγνωριστικό του καταστήματος. Ο κάθε κόμβος του δένδρου αποτελεί μια εγγραφή τύπου `struct Shop` με τα ακόλουθα πεδία:

- **sid:** Αναγνωριστικό (τύπου `int`) που χαρακτηρίζει μοναδικά το κατάστημα.
- **type:** Ακέραιος αριθμός που δηλώνει τον τύπο του καταστήματος.
- **lc:** Δείκτης (τύπου `struct Shop`) στο αριστερό παιδί του κόμβου.
- **rc:** Δείκτης (τύπου `struct Shop`) στο δεξί παιδί του κόμβου.
- **products:** Δείκτης (τύπου `struct Product`) στην αρχή μιας **απλά συνδεδεμένης, ταξινομημένης** λίστας ως προς το αναγνωριστικό του προϊόντος, που ονομάζεται **λίστα προϊόντων καταστήματος**. Το κάθε στοιχείο της λίστας αποτελεί μία εγγραφή τύπου `struct Product` με τα ακόλουθα πεδία:

- **pid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το προϊόν.
- **qty**: Ακέραιος αριθμός που δηλώνει την ποσότητα (quantity) του διαθέσιμου προϊόντος στο κατάστημα.
- **price**: Ακέραιος αριθμός που δηλώνει την τιμή του προϊόντος.
- **next**: Δείκτης (τύπου struct Product) στον επόμενο κόμβο της λίστας προϊόντων καταστήματος.

Ο κόμβος φρουρός είναι και αυτός τύπου Shop με την ιδιαιτερότητα ότι το αναγνωριστικό του καταστήματος (sid) και ο τύπος του έχουν τιμή -1, ενώ οι δείκτες (lc, rc, products) έχουν τιμή NULL.

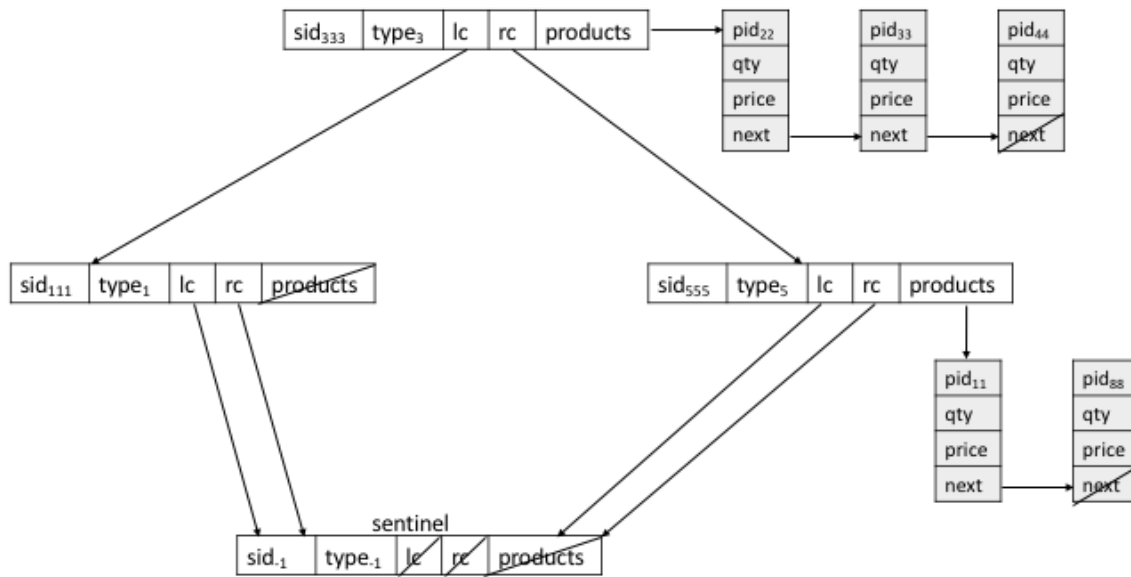
Στα πλαίσια της εργασίας θεωρούμε ότι υπάρχουν πέντε κατηγορίες καταστημάτων. Συνεπώς μπορούμε να θεωρήσουμε ότι το πρώτο ψηφίο του αναγνωριστικού ενός καταστήματος υποδεικνύει τον τύπο του καταστήματος. Για παράδειγμα το κατάστημα με αναγνωριστικό 512, αποτελεί κατάστημα που ανήκει στην κατηγορία 5. Το κατάστημα με αναγνωριστικό 223 αντιστοιχεί στην κατηγορία 2. Η ιδιότητα αυτή θα είναι ιδιαίτερα χρήσιμη σε κάποια από τα ζητούμενα γεγονότα της εργασίας.

Το σχήμα 1 παρουσιάζει το δένδρο καταστημάτων με τις λίστες προϊόντων κάθε καταστήματος.

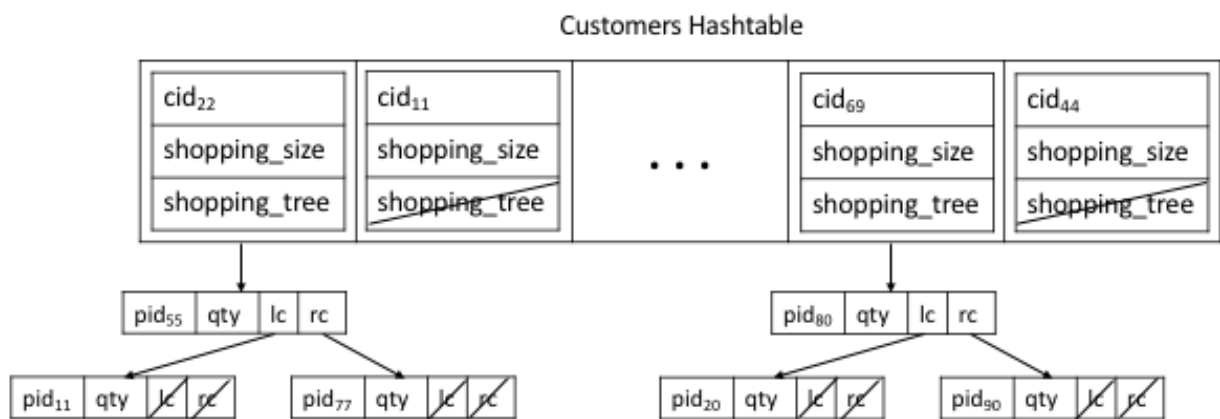
Οι καταναλωτές αποθηκεύονται σε έναν πίνακα κατακερματισμού που ονομάζεται **πίνακας κατακερματισμού καταναλωτών** (customers hash table). Για την επίλυση των συγκρούσεων χρησιμοποιεί την τεχνική του διπλού κατακερματισμού με ανοικτή διευθυνσιοδότηση (double hashing). Το μέγεθος του πίνακα κατακερματισμού θα πρέπει να επιλεγεί από εσάς προσεκτικά και θα πρέπει να είστε σε θέση να δικαιολογήσετε την επιλογή σας. Το μέγιστο πλήθος των καταστημάτων που μπορεί να υπάρχουν θα σας παρέχεται μέσω της μεταβλητής max\_customers\_g. Το μέγεθος του hash table θα πρέπει να επιλέγεται από ένα πίνακα πρώτων αριθμών που θα σας παρέχεται (int primes\_g[160]). Ο κάθε καταναλωτής αποτελεί μία εγγραφή τύπου struct Consumer με τα ακόλουθα πεδία:

- **cid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον καταναλωτή.
- **shopping\_tree**: Δείκτης (τύπου struct C\_Product) στη ρίζα ενός δυαδικού δένδρου αναζήτησης ταξινομημένου βάσει του αναγνωριστικού του προϊόντος. Το δένδρο αυτό ονομάζεται δένδρο προϊόντων καταναλωτή. Ο κάθε κόμβος του δένδρου αποτελεί μία εγγραφή τύπου struct C\_Product με τα ακόλουθα πεδία:
  - **pid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το προϊόν που επιθυμεί να αγοράσει ο καταναλωτής.
  - **qty**: Ακέραιος αριθμός που δηλώνει την ποσότητα του προϊόντος που επιθυμεί να αγοράσει ο καταναλωτής.
  - **lc**: Δείκτης (τύπου struct C\_Product) στο αριστερό παιδί του κόμβου.
  - **rc**: Δείκτης (τύπου struct C\_Product) στο δεξί παιδί του κόμβου.
- **shopping\_size**: Ακέραιος αριθμός που δηλώνει το μέγεθος του δένδρου προϊόντων καταναλωτή.

Στο σχήμα 2 παρουσιάζεται ο πίνακας κατακερματισμού καταναλωτών και τα αντίστοιχα δέντρα προϊόντων του κάθε καταναλωτή.



Σχήμα 1 το δένδρο καταστημάτων με κόμβο φρουρό και οι λίστες προϊόντων κάθε καταστήματος.



Σχήμα 2 ο πίνακας κατακερματισμού καταναλωτών και τα αντίστοιχα δέντρα προϊόντων του κάθε καταναλωτή.

**Τρόπος Λειτουργίας Προγράμματος**

Το πρόγραμμα που θα δημιουργηθεί θα πρέπει να εκτελείται καλώντας την ακόλουθη εντολή:

**<executable> <input-file>**

όπου <executable> είναι το όνομα του εκτελέσιμου αρχείου του προγράμματος (π.χ. a.out) και <input-file> είναι το όνομα ενός αρχείου εισόδου (π.χ. testfile) το οποίο περιέχει γεγονότα των ακόλουθων μορφών:

**R <sid> <type>**

Γεγονός τύπου register shop που υποδηλώνει τη δημιουργία ενός νέου καταστήματος με αναγνωριστικό <sid>. Η παράμετρος <type> υποδηλώνει τον τύπο του καταστήματος. Κατά το γεγονός αυτό, θα γίνεται εισαγωγή ενός νέου κόμβου τύπου struct Shop στο δένδρο καταστημάτων. Αρχικά, η λίστα προϊόντων του καταστήματος είναι κενή. Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
R <sid> <type>
  Shops = <sid1,type1>, <sid2,type2>, <sid3,type3>, . . . , <sidn,typen>
DONE
```

όπου  $n$  είναι ο αριθμός των κόμβων στο δένδρο καταστημάτων και για κάθε  $i \in \{1, \dots, n\}$ , <sid <sub>$i$</sub> > και <type <sub>$i$</sub> > είναι το αναγνωριστικό και ο τύπος του καταστήματος, αντίστοιχα, που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου καταστημάτων (βάσει της ενδοδιατεταγμένης διάσχισης).

**S <sid> <pid> <quantity> <price>**

Γεγονός τύπου supply shop που υποδηλώνει την εισαγωγή ενός νέου προϊόντος με αναγνωριστικό <pid> στη λίστα προϊόντων του καταστήματος με αναγνωριστικό <sid>. Οι παράμετροι <quantity> και <price> υποδηλώνουν την ποσότητα και την τιμή του νέου προϊόντος, αντίστοιχα. Κατά το γεγονός αυτό, αρχικά θα πρέπει να αναζητάτε το κατάστημα με αναγνωριστικό <sid> στο δένδρο καταστημάτων. Στη συνέχεια, θα εισάγετε στη λίστα προϊόντων του καταστήματος το νέο προϊόν. Μετά από κάθε εισαγωγή, η λίστα προϊόντων καταστήματος θα πρέπει να μένει ταξινομημένη. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
S <sid> <pid> <quantity> <price>
  Shop1 = <pid1,1,quantity1,1,price1,1>, <pid1,2,quantity1,2,price1,2>, . . . , <pid1,m1,quantity1,m1,price1,m1>
  Shop2 = <pid2,1,quantity2,1,price2,1>, <pid2,2,quantity2,2,price2,2>, . . . , <pid2,m2,quantity2,m2,price2,m2>
  . . .
  Shopn = <pidn,1,quantityn,1,pricen,1>, <pidn,2,quantityn,2,pricen,2>, . . . , <pidn,mn,quantityn,mn,pricen,mn>
DONE
```

όπου  $n$  είναι το πλήθος των κόμβων στο δένδρο καταστημάτων και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $m_i$  είναι το πλήθος των κόμβων της λίστας προϊόντων του  $i$ -οστού καταστήματος σύμφωνα με την ενδοδιατεταγμένη διάσχιση, και για κάθε  $j \in [1, \dots, m_i]$ , pid <sub>$i,j$</sub> , quantity <sub>$i,j$</sub>  και price <sub>$i,j$</sub>  είναι το αναγνωριστικό, η ποσότητα και η τιμή του προϊόντος, αντίστοιχα, του  $j$ -οστού κόμβου της λίστας προϊόντων του  $i$ -οστού καταστήματος.

**C <cid>**

Γεγονός τύπου register customer που υποδηλώνει την εισαγωγή ενός νέου καταναλωτή με αναγνωριστικό <cid> στον πίνακα κατακερματισμού καταναλωτών. Κατά το γεγονός αυτό, θα γίνεται εισαγωγή ενός νέου κόμβου τύπου struct Customer στη λίστα καταστημάτων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
C <cid>
  Customers = <cid1>, <cid2>, <cid3>, . . . , <cidn>
DONE
```

όπου  $n$  είναι ο αριθμός των πελατών στον πίνακα κατακερματισμού των πελατών και για κάθε  $i \in \{1, \dots, n\}$ , <cid <sub>$i$</sub> > είναι το αναγνωριστικό του καταναλωτή που αντιστοιχεί στον  $i$ -οστό καταναλωτή στον πίνακα (αν διασχίσουμε τον πίνακα από τη θέση 0 μέχρι και την τελευταία θέση του).

**L <cid> <pid> <quantity>**

Γεγονός τύπου add to shopping list που υποδηλώνει την εισαγωγή ενός νέου προϊόντος με αναγνωριστικό <pid> και ποσότητα <quantity> στο δένδρο προϊόντων του καταναλωτή με αναγνωριστικό <cid>. Κατά το γεγονός αυτό, θα πρέπει να αναζητείτε στον πίνακα κατακερματισμού καταναλωτών τον καταναλωτή με αναγνωριστικό <cid>. Στη συνέχεια θα εισάγετε στο δένδρο προϊόντων του καταναλωτή ένα νέο προϊόν καταναλωτή (C\_Product) με αναγνωριστικό <pid> και ποσότητα <quantity>. Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L <cid> <pid> <quantity>
  Customer1 = <pid1,1,quantity1,1>, <pid1,2,quantity1,2>, . . . , <pid1,m1,quantity1,m1>
  Customer2 = <pid2,1,quantity2,1>, <pid2,2,quantity2,2>, . . . , <pid2,m2,quantity2,m2>
  . . .
  Customern = <pidn,1,quantityn,1>, <pidn,2,quantityn,2>, . . . , <pidn,mn,quantityn,mn>
DONE
```

όπου  $n$  είναι το πλήθος των έγκυρων στοιχείων που περιέχει ο πίνακας κατακερματισμού και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $m_i$  είναι το πλήθος των κόμβων του δένδρου προϊόντων του  $i$ -οστού τέτοιου καταναλωτή, και για κάθε  $j \in [1, \dots, m_i]$ ,  $pid_{i,j}$ , και  $quantity_{i,j}$  είναι το αναγνωριστικό και η ποσότητα του προϊόντος που αντιστοιχεί στον  $j$ -οστό κόμβο του δένδρου προϊόντων (σύμφωνα με την ενδοδιατεταγμένη διάσχιση) του  $i$ -οστού καταναλωτή.

**D <cid> <pid>**

Γεγονός τύπου delete from shopping list που υποδηλώνει τη διαγραφή ενός προϊόντος με αναγνωριστικό <pid> από το δένδρο προϊόντων του καταναλωτή με αναγνωριστικό <cid>. Για να εκτελέσετε το γεγονός αυτό, θα πρέπει, αρχικά, να αναζητείτε στον πίνακα κατακερματισμού καταναλωτών τον καταναλωτή με αναγνωριστικό <cid>. Στη συνέχεια θα πρέπει να διαγράψετε από το δένδρο προϊόντων του καταναλωτή τον κόμβο με αναγνωριστικό <pid>. Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
D <cid> <pid>
  Customer1 = <pid1,1,quantity1,1>, <pid1,2,quantity1,2>, . . . , <pid1,m1,quantity1,m1>
  Customer2 = <pid2,1,quantity2,1>, <pid2,2,quantity2,2>, . . . , <pid2,m2,quantity2,m2>
  . . .
  Customern = <pidn,1,quantityn,1>, <pidn,2,quantityn,2>, . . . , <pidn,mn,quantityn,mn>
DONE
```

όπου  $n$  είναι το πλήθος των στοιχείων του πίνακα κατακερματισμού καταναλωτών, για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $m_i$  είναι το πλήθος των κόμβων του δένδρου προϊόντων του  $i$ -οστού καταναλωτή και για κάθε  $j \in [1, \dots, m_i]$ ,  $pid_{i,j}$ , και  $quantity_{i,j}$  είναι το αναγνωριστικό και η ποσότητα του προϊόντος που αντιστοιχεί στον  $j$ -οστό κόμβο του δένδρου προϊόντων (σύμφωνα με την ενδοδιατεταγμένη διάσχιση) του  $i$ -οστού καταναλωτή.

**G <cid>**

Γεγονός τύπου go shopping που υποδηλώνει την αναζήτηση των προϊόντων από το δένδρο προϊόντων του καταναλωτή με αναγνωριστικό <cid> και την αγορά τους από τις λίστες προϊόντων καταστημάτων. Κατά το γεγονός αυτό, θα εφαρμόσετε τα παρακάτω βήματα:

1. Αρχικά θα αναζητήσετε στον πίνακα κατακερματισμού καταναλωτών τον κόμβο με αναγνωριστικό <cid>.
2. Στη συνέχεια θα αντιγράψετε τα προϊόντα του καταναλωτή από το δένδρο του σε έναν βοηθητικό πίνακα που θα δημιουργήσετε. Το μέγεθος του πίνακα θα είναι ίσο με τον αριθμό των κόμβων που περιέχει το δένδρο (κάνοντας χρήση του μετρητή shopping\_size από τη δομή του καταναλωτή). Κατά την αντιγραφή των κόμβων του δένδρου στον πίνακα πρέπει να τοποθετείτε τα προϊόντα ταξινομημένα ως προς το αναγνωριστικό τους. Η διαδικασία αυτή πρέπει να εκτελείται σε χρόνο  $O(n)$ , με το  $n$  να αντιστοιχεί στον αριθμό των κόμβων του δένδρου.
3. Έπειτα, θα διατρέξετε το δένδρο των καταστημάτων εφαρμόζοντας ενδοδιατεταγμένη διάσχιση.
  - a. Για κάθε κόμβο που επισκέπτεσθε, θα διατρέχετε τη λίστα προϊόντων του καταστήματος παράλληλα με τον πίνακα που δημιουργήσατε στο προηγούμενο βήμα. Για κάθε προϊόν που υπάρχει στον πίνακα αλλά και στην λίστα προϊόντων του καταστήματος, θα ενημερώνετε κατάλληλα το πεδίο quantity του

προϊόντος τόσο του καταναλωτή, όσο και του καταστήματος, υποδηλώνοντας έτσι την αγορά του προϊόντος. Η διαδικασία αυτή πρέπει να εκτελείται σε χρόνο  $O(m)$ , όπου  $m$  είναι  $\min(n, l)$ , όπου  $l$  είναι το μέγεθος της λίστας προϊόντων του καταστήματος.

Κάθε φορά που διατρέχετε τη λίστα προϊόντων ενός καταστήματος που επισκέπτεστε στο δένδρο καταστημάτων, θα διατρέχετε τον βοηθητικό πίνακα προϊόντων απ' την αρχή.

Το βήμα 3.α εκτελείται μέχρι να αγοραστούν όλα τα προϊόντα του καταναλωτή στις επιθυμητές ποσότητες (το πεδίο quantity για όλα τα προϊόντα του πίνακα να είναι 0) ή μέχρι να διασχίσετε όλο το δένδρο καταστημάτων, διασφαλίζοντας έτσι ότι δεν υπάρχουν διαθέσιμα περισσότερα προϊόντα από αυτά που επιθυμεί να αγοράσει ο καταναλωτής σε κανένα κατάστημα.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
G <cid>
  Customercid = <pid1, quantity1>, <pid2, quantity2>, . . . , <pidn, quantityn>

  Shop1 = <pid1,1, quantity1,1, price1,1>, <pid1,2, quantity1,2, price1,2>, . . . , <pid1,m1, quantity1,m1, price1,m1>
  Shop2 = <pid2,1, quantity2,1, price2,1>, <pid2,2, quantity2,2, price2,2>, . . . , <pid2,m2, quantity2,m2, price2,m2>
  . . .
  Shopn = <pidk,1, quantityk,1, pricek,1>, <pidk,2, quantityk,2, pricek,2>, . . . , <pidk,mn, quantityk,mn, pricek,mn>

DONE
```

όπου  $n$  είναι το πλήθος των κόμβων του δένδρου προϊόντων του καταναλωτή με αναγνωριστικό  $\langle \text{cid} \rangle$ , για κάθε  $1 \leq i \leq n$ ,  $\text{pid}_i$ , και  $\text{quantity}_i$  είναι το αναγνωριστικό και η ποσότητα του προϊόντος που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου προϊόντων καταναλωτή με αναγνωριστικό  $\langle \text{cid} \rangle$ , σύμφωνα με την ενδοδιατεταγμένη διάσχιση. Επιπλέον θεωρείστε ότι το πλήθος των κόμβων στο δένδρο καταστημάτων είναι  $k$  και για κάθε  $i$ ,  $1 \leq i \leq k$ ,  $m_i$  είναι το πλήθος των κόμβων της λίστας προϊόντων καταστήματος του  $i$ -οστού καταστήματος (σύμφωνα με την ενδοδιατεταγμένη διάσχιση), και για κάθε  $j \in [1, \dots, m_i]$ ,  $\text{pid}_{i,j}$ ,  $\text{quantity}_{i,j}$  και  $\text{price}_{i,j}$  είναι το αναγνωριστικό, η ποσότητα και η τιμή του προϊόντος αντίστοιχα του  $j$ -οστού κόμβου της λίστας προϊόντων του  $i$ -οστού καταστήματος.

## M $\langle \text{cid}_1 \rangle \langle \text{cid}_2 \rangle$

Γεγονός που υποδηλώνει ότι δύο καταναλωτές πάνε για ψώνια μαζί και συνεπώς συνενώνουν τις επιθυμίες τους για αγορές.

Κατά το γεγονός αυτό, αρχικά θα αναζητήσετε τους καταναλωτές με αναγνωριστικά  $\langle \text{cid}_1 \rangle$  και  $\langle \text{cid}_2 \rangle$  στον πίνακα κατακερματισμού καταναλωτών. Στη συνέχεια θα συνενώσετε τα δέντρα προϊόντων των δύο καταναλωτών εκτελώντας τα παρακάτω βήματα:

1. Θα δημιουργήσετε έναν βοηθητικό πίνακα με μέγεθος ίσο με τον αριθμό των κόμβων του δένδρου προϊόντων του καταναλωτή με αναγνωριστικό  $\langle \text{cid}_1 \rangle$  (κάνοντας χρήση του μετρητή `shopping_size` από τη δομή του καταναλωτή).
2. Θα αντιγράψετε τα προϊόντα του δένδρου στον πίνακα. Τα προϊόντα πρέπει να εισάγονται ταξινομημένα στον πίνακα. Η διαδικασία αυτή πρέπει να εκτελείται σε χρόνο  $O(n)$ , όπου  $n$  είναι το μέγεθος του δένδρου.
3. Στη συνέχεια θα εφαρμόσετε τα βήματα (1) και (2) ξανά για τον καταναλωτή με αναγνωριστικό  $\langle \text{cid}_2 \rangle$ .
4. Επιπρόσθετα, θα δημιουργήσετε έναν νέο βοηθητικό πίνακα με μέγεθος ίσο με το συνολικό μέγεθος των προηγούμενων πινάκων
5. Θα συνενώσετε του δύο πρώτους ταξινομημένους πίνακες και το αποτέλεσμα θα αποθηκευτεί στον πίνακα του βήματος (4). Ο πίνακας που προκύπτει πρέπει να είναι επίσης ταξινομημένος. Η διαδικασία αυτή πρέπει να εκτελεστεί με χρονική πολυπλοκότητα  $O(n+m)$ , με  $n$  και  $m$  είναι τα μεγέθη των πινάκων που προκύπτουν από τα βήματα (1) και (3), αντίστοιχα.
6. Από τον πίνακα που προκύπτει στο βήμα (5), θα δημιουργήσετε ένα νέο δένδρο προϊόντων καταναλωτή που θα αντικαταστήσει το αντίστοιχο δένδρο του καταναλωτή με αναγνωριστικό  $\langle \text{sid}_1 \rangle$ . Το δένδρο που θα δημιουργηθεί θα πρέπει να έχει το ελάχιστο δυνατό ύψος.
7. Θα διαγράψετε τον καταναλωτή με αναγνωριστικό  $\langle \text{cid}_2 \rangle$  από τον πίνακα κατακερματισμού καταναλωτών.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
M <cid1> <cid2>
  Customercid1 = <pid1, quantity1, price1>, <pid2, quantity2, price2>, . . . , <pidn, quantityn, pricen>

DONE
```

όπου  $n$  είναι το πλήθος των κόμβων του δένδρου προϊόντων του καταναλωτή με αναγνωριστικό  $\langle cid1 \rangle$  και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $pid_i$ ,  $quantity_i$  και  $price_i$  είναι το αναγνωριστικό, η ποσότητα και η τιμή του προϊόντος που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου του καταναλωτή με αναγνωριστικό  $\langle cid1 \rangle$  σύμφωνα με την ενδοδιατεταγμένη διάσχιση.

**X**

Γεγονός τύπου `print shops` που υποδηλώνει την εκτύπωση όλων των καταστημάτων από το δένδρο καταστημάτων (εφαρμόζοντας ενδοδιατεταγμένη διάσχιση) καθώς και την εκτύπωση των λιστών προϊόντων τους. Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
X
Shop1 = <pid1,1,quantity1,1,price1,1>, <pid1,2,quantity1,2,price1,2>, . . . , <pid1,m1,quantity1,m1,price1,m1>
Shop2 = <pid2,1,quantity2,1,price2,1>, <pid2,2,quantity2,2,price2,2>, . . . , <pid2,m2,quantity2,m2,price2,m2>
. . .
Shopn = <pidn,1,quantityn,1,pricen,1>, <pidn,2,quantityn,2,pricen,2>, . . . , <pidn,mn,quantityn,mn,pricen,mn>
DONE
```

όπου  $n$  είναι το πλήθος των κόμβων στο δένδρο καταστημάτων και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $m_i$  είναι το πλήθος των κόμβων της λίστας προϊόντων καταστήματος του  $i$ -οστού καταστήματος (σύμφωνα με την ενδοδιατεταγμένη διάσχιση) και για κάθε  $j \in [1, \dots, m_i]$ ,  $pid_{i,j}$ ,  $quantity_{i,j}$  και  $price_{i,j}$  είναι το αναγνωριστικό, η ποσότητα και η τιμή του προϊόντος αντίστοιχα του  $j$ -οστού κόμβου της λίστας προϊόντων του  $i$ -οστού καταστήματος.

**Y**

Γεγονός τύπου `print customers` που υποδηλώνει την εκτύπωση όλων των καταναλωτών από τον πίνακα κατακερματισμού καταναλωτών καθώς και την εκτύπωση των δεντρών προϊόντων τους. Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
Y
Customer1 = <pid1,1,quantity1,1>, <pid1,2,quantity1,2>, . . . , <pid1,m1,quantity1,m1>
Customer2 = <pid2,1,quantity2,1>, <pid2,2,quantity2,2>, . . . , <pid2,m2,quantity2,m2>
. . .
Customern = <pidn,1,quantityn,1>, <pidn,2,quantityn,2>, . . . , <pidn,mn,quantityn,mn>
DONE
```

όπου  $n$  είναι το πλήθος των έγκυρων κόμβων στον πίνακα κατακερματισμού καταναλωτών και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $m_i$  είναι το πλήθος των κόμβων του δένδρου προϊόντων του  $i$ -οστού έγκυρου καταναλωτή, και για κάθε  $j \in [1, \dots, m_i]$ ,  $pid_{i,j}$  και  $quantity_{i,j}$  είναι το αναγνωριστικό και η ποσότητα του προϊόντος που αντιστοιχεί στον  $j$ -οστό κόμβο του δένδρου προϊόντων του  $i$ -οστού καταναλωτή (εφαρμόζοντας ενδοδιατεταγμένη διάσχιση).

**T [bonus, 15%]**

Γεγονός τύπου `categorize stores` που υποδηλώνει την κατηγοριοποίηση των καταστημάτων με βάση τον τύπο τους. Κατά το γεγονός αυτό, θα διαχωρίσετε το δένδρο καταστημάτων σε 5 μικρότερα δέντρα ίδιου τύπου. Το κάθε δένδρο θα περιέχει καταστήματα του ίδιου τύπου. Η διαδικασία αυτή θα πρέπει να εκτελείται με χρονική πολυπλοκότητα  $O(h)$ , όπου  $h$  το ύψος του δένδρου καταστημάτων. Για να το πετύχετε αυτό λάβετε υπόψη σας την ιδιότητα των αναγνωριστικών των καταστημάτων όπως παρουσιάζεται στην περιγραφή των δομών παραπάνω. Κάθε διαχωρισμός πρέπει να επιτυγχάνεται διασχίζοντας ένα μόνο μονοπάτι (διαφορετικό για κάθε τύπο καταστήματος) του δένδρου καταστημάτων.

Μετά το πέρας της εκτέλεσης του γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
T
Type1
Shop1,1 = <pid1,1,1,quantity1,1,1,price1,1,1>, . . . , <pidl1,1,1,quantityl1,1,1,pricel1,1,1>
. . .
Shopn1,1 = <pid1,n1,1,quantity1,n1,1,price1,n1,1>, . . . , <pidln1,1,1,quantityln1,1,1,priceln1,1,1>
. . .
Type5
Shop1,5 = <pid1,1,5,quantity1,1,5,price1,1,5>, . . . , <pidl1,1,5,quantityl1,1,5,pricel1,1,5>
. . .
Shopn5,5 = <pid1,n5,5,quantity1,n5,5,price1,n5,5>, . . . , <pidln5,n5,5,quantityln5,n5,5,priceln5,n5,5>
DONE
```

για κάθε  $t$ ,  $1 \leq t \leq 5$ ,  $n_t$  είναι το πλήθος των καστημάτων της κατηγορίας  $t$ . Για κάθε  $i$ ,  $1 \leq i \leq n_t$ ,  $\text{Shop}_{i,t}$  είναι το αναγνωριστικό του  $i$ -στού καταστήματος (σύμφων με την ενδοδιατεταγμένη διάσχιση) του δένδρου κατηγορίας  $t$  και  $l_{i,t}$  είναι το μέγεθος της λίστας προϊόντων του  $i$ -στού καταστήματος σύμφωνα με την ενδοδιατεταγμένη διάσχιση του δένδρου της  $t$ -στής κατηγορίας. Για κάθε  $j \in [1, \dots, l_{i,t}]$ ,  $\text{pid}_{j,i,t}$ ,  $\text{quantity}_{j,i,t}$ ,  $\text{price}_{j,i,t}$  είναι το αναγνωριστικό, η ποσότητα και η τιμή αντίστοιχα του  $j$ -στού προϊόντος της λίστας προϊόντων του  $i$ -στού καταστήματος (σύμφωνα με την ενδοδιατεταγμένη διάσχιση) που ανήκει στη  $t$ -στη κατηγορία.

## Βαθμολογία Γεγονότων

<b>R</b>	<b>8 (10)</b>
<b>S</b>	<b>10 (12)</b>
<b>C</b>	<b>9 (14)</b>
<b>L</b>	<b>10 (12)</b>
<b>D</b>	<b>12</b>
<b>G</b>	<b>15 (16)</b>
<b>M</b>	<b>15 (16)</b>
<b>X</b>	<b>4</b>
<b>Y</b>	<b>4</b>
<b>T (Bonus)</b>	<b>15 (10)</b>



## Δομές Δεδομένων

Στην υλοποίησή σας δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες δομές δεδομένων (πχ., ArrayList) είτε η υλοποίηση πραγματοποιηθεί στη C είτε στη Java. Στη συνέχεια παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας.

```
struct Shop {
    int sid;
    int type;
    struct Product *products;
    struct Shop *lc;
    struct Shop *rc;
};

struct Product {
    int pid;
    int qty;
    int price;
    struct Product *next;
};

struct Customer {
    int cid;
    int shopping_size;
    struct C_Product *shopping_tree;
};

struct C_Product {
    int pid;
    int qty;
    struct Product *lc;
    struct Product *rc;
};

/* global, ο κόμβος root του δυαδικού δένδρου καταστημάτων */
struct Shop shopRoot;

/* global, ο κόμβος φρουρός του δυαδικού δένδρου καταστημάτων */
struct Shop shopSentinel;

/* Το μέγιστο πλήθος των καταστημάτων */
int max_customers_g
```