HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2015-16
Διδάσκουσα: Παναγιώτα Φατούρου
Προγραμματιστική Εργασία - 1^{ο} Μέρος

Ημερομηνία Παράδοσης: Δευτέρα, 16 Νοεμβρίου 2015, ώρα 23:59.

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

Γενική Περιγραφή

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που προσομοιώνει τη **διαδικασία επιλογής και αποστολής δώρων του Αγίου Βασιλείου** σε γεωγραφικά διαμερίσματα διάφορων χωρών του κόσμου κατά την περίοδο των Χριστουγέννων.

Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Για την υλοποίηση της προσομοίωσης της διαδικασίας επιλογής και αποστολής δώρων του Αγίου Βασιλείου, παιδιά από γεωγραφικά διαμερίσματα διάφορων χωρών του κόσμου στέλνουν γράμματα με τα στοιχεία και τις προτιμήσεις τους.

Για την καλύτερη διαχείριση των γραμμάτων, τα παιδιά πρέπει να διαχωριστούν σε κατηγορίες με βάση την ηλικία τους. Πιο συγκεκριμένα, υπάρχουν τέσσερις ηλικιακές ομάδες: η πρώτη ομάδα περιέχει παιδιά ηλικίας 0-3 ετών, η δεύτερη 4-7 ετών, η τρίτη 8-11 ετών και η τέταρτη 12-15 ετών. Για τη διαχείριση των ηλικιακών ομάδων, θα δημιουργήσετε έναν πίνακα σταθερού μεγέθους **4 θέσεων**, ο οποίος θα ονομάζεται **πίνακας ηλικιακών κατηγοριών**. Η κάθε θέση του πίνακα θα περιέχει ένα δείκτη (τύπου child) στον πρώτο κόμβο μιας **απλά συνδεδεμένης** λίστας, η οποία θα ονομάζεται **λίστα παιδιών της ηλικιακής κατηγορίας** αυτής. Ο κάθε κόμβος της λίστας αυτής περιέχει πληροφορίες για κάποιο παιδί που ανήκει σε αυτήν την ηλικιακή κατηγορία. Ένας τέτοιος κόμβος θα αποτελεί μια εγγραφή τύπου child με τα ακόλουθα πεδία:

- **cid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το παιδί.
- **age:** Αριθμός (τύπου int) που αντιστοιχεί στην ηλικία του παιδιού.
- **did:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το γεωγραφικό διαμέρισμα στο οποίο κατοικεί το παιδί (περισσότερες λεπτομέρειες για τα γεωγραφικά διαμερίσματα παρακάτω).
- **present_choices:** Ένας πίνακας σταθερού μεγέθους τριών (3) θέσεων με τις προτιμήσεις δώρων του παιδιού. Το κάθε κελί αυτού του πίνακα περιέχει μοναδικά αναγνωριστικά δώρων (τύπου int), σύμφωνα με τις προτιμήσεις του παιδιού.
- **next**: Δείκτης (τύπου child) στον επόμενο κόμβο της **λίστας παιδιών της κατηγορίας.**

Η λίστα παιδιών κάθε ηλικιακής κατηγορίας είναι **ταξινομημένη** βάσει των αναγνωριστικών των παιδιών. Στο Σχήμα 1 παρουσιάζεται ο πίνακας ηλικιακών κατηγοριών, σταθερού μεγέθους 4 θέσεων, και η λίστα των παιδιών που δεικτοδοτείται από το εκάστοτε στοιχείο του.

Ο Άγιος Βασίλειος διατηρεί μια αποθήκη με αποθέματα δώρων. Για το σκοπό αυτό θα δημιουργήσετε μια **μη-ταξινομημένη, διπλά συνδεδεμένη** λίστα **με κόμβο φρουρό**, η οποία θα ονομάζεται **λίστα αποθεμάτων**. Κάθε κόμβος της λίστας αποθεμάτων θα περιέχει μια εγγραφή τύπου present με τα ακόλουθα πεδία:

- **pid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το δώρο.
- **stock_cnt:** Αριθμός (τύπου int) που αντιστοιχεί στο απόθεμα του δώρου που είναι διαθέσιμο στην αποθήκη.
- **request_cnt:** Αριθμός (τύπου int) που αντιστοιχεί στο πλήθος των παιδιών που έχουν εκφράσει επιθυμία για το δώρο. Αρχικά, ο μετρητής αυτός θα έχει τιμή 0 και θα αυξάνεται καθώς ο Άγιος Βασίλειος δέχεται τα γράμματα των παιδιών.
- **prev**: Δείκτης (τύπου present) στον προηγούμενο κόμβο της λίστας.
- **next**: Δείκτης (τύπου present) στον επόμενο κόμβο της λίστας.

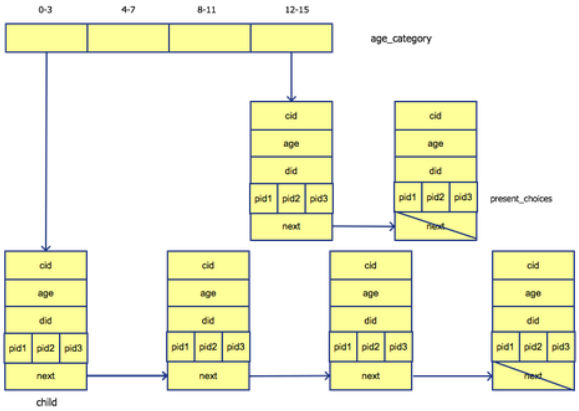
Ο κόμβος φρουρός της λίστας είναι ένας διαχειριστικός κόμβος για τον οποίο ισχύουν τα εξής: ο δείκτης next του κόμβου αυτού δείχνει πάντα στο πρώτο στοιχείο της λίστας, ο δείκτης prev δείχνει πάντα στο τελευταίο στοιχείο της λίστας. Ο κόμβος φρουρός είναι και αυτός τύπου present με την ιδιαιτερότητα ότι το αναγνωριστικό (pid) και οι μετρητές του (stock_cnt, request_cnt) έχουν τιμή -1. Το Σχήμα 2 απεικονίζει τη λίστα αποθεμάτων των δώρων (presents). Στη λίστα αυτή εφαρμόζεται το ακόλουθο ευριστικό αναδιάρθρωσης της λίστας το οποίο αποσκοπεί στη διατήρηση των κόμβων που αναζητούνται πιο συχνά στην αρχή της λίστας και άρα εφαρμόζεται για λόγους απόδοσης. Ο κόμβος που αναζητήθηκε θα μεταφέρεται 5 θέσεις προς την αρχή της λίστας ξεκινώντας από την τρέχουσα θέση του στη λίστα. Αν ένας

κόμβος βρίσκεται σε μια από τις 4 πρώτες θέσεις της λίστας, θα πρέπει απλά να μεταφέρεται στην πρώτη θέση.

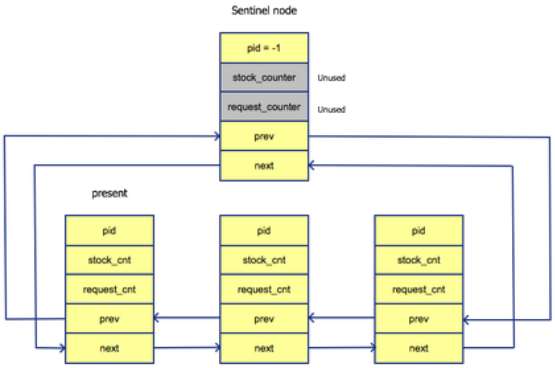
Για να ικανοποιήσει τις επιθυμίες των παιδιών, ο Άγιος Βασίλειος διαχωρίζει τις παραδόσεις των δώρων με βάση το γεωγραφικό διαμέρισμα (και άρα και τη χώρα) που διαμένει το κάθε παιδί. Για αυτό το σκοπό θα χρησιμοποιήσετε μια **ταξινομημένη, απλά-συνδεδεμένη** λίστα, η οποία θα ονομάζεται **λίστα γεωγραφικών διαμερισμάτων** (DistrictsL). Η λίστα αυτή είναι ταξινομημένη βάσει του αναγνωριστικού του γεωγραφικού διαμερίσματος. Κάθε στοιχείο της **λίστας γεωγραφικών διαμερισμάτων** είναι μια εγγραφή τύπου district με τα ακόλουθα πεδία:

- **did:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά ένα γεωγραφικό διαμέρισμα και τη χώρα στην οποία ανήκει.
- **assignL:** Δείκτης (τύπου present_assign) στο πρώτο στοιχείο μιας **μη ταξινομημένης, απλά συνδεδεμένης** λίστας, η οποία ονομάζεται **λίστα ανάθεσης δώρων.** Η λίστα αυτή περιέχει τα δώρα που θα λάβουν τα παιδιά του γεωγραφικού διαμερίσματος με αναγνωριστικό did. Κάθε στοιχείο της λίστας αυτής αποτελεί μια εγγραφή τύπου present_assignment με τα ακόλουθα πεδία:
  - **cid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το παιδί.
  - **pid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το δώρο.
  - **s_degree:** Αριθμός (τύπου int) που αντιστοιχεί στο βαθμό ικανοποίησης του παιδιού. Η μεταβλητή αυτή λαμβάνει τιμές από 1 μέχρι 5, με 1 να αντιστοιχεί στον ελάχιστο βαθμό ικανοποίησης του παιδιού και 5 στο μέγιστο βαθμό ικανοποίησης.
  - **next**: Δείκτης (τύπου present_assignment) στον επόμενο κόμβο της **λίστας ανάθεσης δώρων.**
- **next**: Δείκτης (τύπου district) στον επόμενο κόμβο της **λίστας γεωγραφικών διαμερισμάτων.**

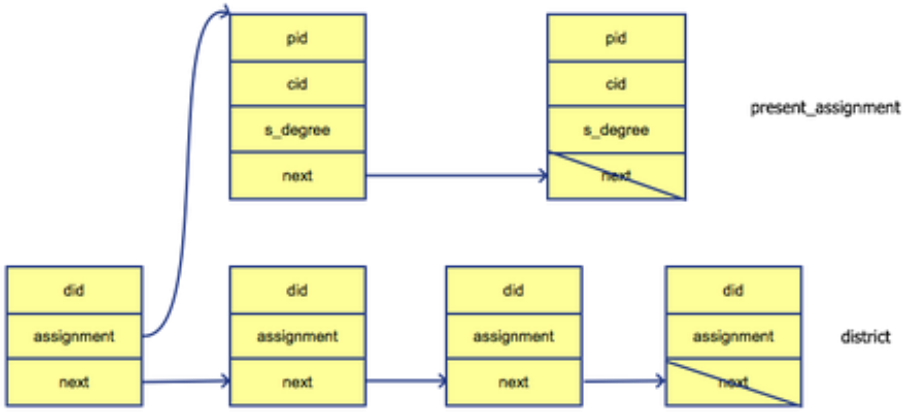
Στο Σχήμα 3 παρουσιάζεται σχηματικά η λίστα γεωγραφικών διαμερισμάτων και οι αντίστοιχες λίστες ανάθεσης δώρων.



Σχήμα 1: Πίνακας ηλικιακών κατηγοριών και αντίστοιχη ταξινομημένη (βάσει cid), απλά συνδεδεμένη λίστα παιδιών για κάθε ηλικιακή κατηγορία



Σχήμα 2: Διπλά -συνδεδεμένη λίστα αποθεμάτων των δώρων



Σχήμα 3: Ταξινομημένη (βάσει did), απλά-συνεδεδεμένη λίστα γεωγραφικών διαμερισμάτων και μη ταξινομημένη, απλά - συνεδεμένη λίστα ανάθεσης δώρων για κάθε γεωγραφικό διαμέρισμα

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:

– **B <pid> <stock_cnt>**

Γεγονός τύπου *buy present* το οποίο σηματοδοτεί την αγορά αποθέματος για το δώρο (present) με αναγνωριστικό <pid> από τον Άγιο Βασίλειο. Συγκεκριμένα, ο Άγιος Βασίλειος αγοράζει **απόθεμα** του δώρου με αναγωριστικό <pid> ίσο με <stock_cnt>. Μια εγγραφή για το δώρο για το οποίο αγοράστηκε απόθεμα θα πρέπει να εισάγεται στη λίστα **αποθεμάτων των δώρων**. Θεωρήστε ότι κάθε αναγνωριστικό δώρου που αναφέρεται από οποιοδήποτε γεγονός τύπου B ενός test-file θα είναι μοναδικό και σχεδιάστε την Insert() στη λίστα απθοθεμάτων των δώρων ώστε να είναι όσο το δυνατόν πιο αποδοτική. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
B <pid> <stock_cnt>
    Presents = <pid_1>, <pid_2>, ..., <pid_n>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα αποθεμάτων και για κάθε $i \in \{1, \ldots, n\}$, $<pid_i>$ είναι το αναγνωριστικό του δώρου που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής.

– **L <cid> <age> <did> <pid_1> <pid_2> <pid_3>**

Γεγονός τύπου *letter received* το οποίο σηματοδοτεί την παραλαβή από τον Άγιο Βασίλειο ενός γράμματος που έχει αποσταλεί από ένα παιδί με αναγνωριστικό <cid>, ηλικίας <age> που διαμένει στην περιοχή με αναγνωριστικό <did>. Τα $<pid_1>$, $<pid_2>$ και $<pid_3>$ είναι αναγνωριστικά δώρων που αντιστοιχούν στις προτιμήσεις του παιδιού. Για την εκτέλεση του γεγονότος αυτού απαιτείται να γίνουν οι ακόλουθες ενέργειες. Θα πρέπει να πραγματοποιηθεί πρόσβαση του κατάλληλου στοιχείου του πίνακα ηλικιακών κατηγοριών από το οποίο ξεκινάει η κατάλληλη λίστα παιδιών όπου πρέπει να αποθηκευτεί η εγγραφή τύπου child για το παιδί (αν δεν υπάρχει ήδη εκεί). Στη συνέχεια, θα πρέπει να γίνει μια αναζήτηση για τις προτιμήσεις του παιδιού στη λίστα αποθεμάτων και για κάθε δώρο της προτίμησης του παιδιού θα πρέπει να αυξάνεται ο μετρητής ζήτησης (request_cnt) κατά ένα. **Η αναζήτηση θα πρέπει να ενημερώνει όλες τις εγγραφές που αντιστοιχούν στις προτιμήσεις του παιδιού πραγματοποιώντας μια μόνο διάσχιση.** Είναι αξιοσημείωτο πως κατά την αναζήτηση εφαρμόζεται το ευριστικό που έχει περιγραφεί στην Ενότητα "Αναλυτική Περιγραφή της Ζητούμενης Υλοποίησης" παραπάνω.

Αν το δώρο με αναγνωριστικό <pid> δεν υπάρχει στη λίστα αποθεμάτων, ο Άγιος Βασίλειος θα πρέπει να προβαίνει στην αγορά 10 αποθεμάτων του δώρου αυτού. Σε αυτή την περίπτωση, ένας νέος κόμβος (που θα αντιστοιχεί σε αυτό το δώρο) θα πρέπει να εισάγεται στη λίστα αποθεμάτων. Η εισαγωγή αυτή θα πρέπει να πραγματοποιείται σε χρόνο O(1) και δε θα πρέπει να καταστρέφει τη διάταξη των κόμβων στη λίστα βάσει της συχνότητάς τους, όπως αυτή έχει επιτευχθεί από το ευριστικό. Μετά το πέρας της

εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L <cid> <age> <did> <pid1> <pid2> <pid3>
DONE
```

– P

Γεγονός τύπου *prepare presents* το οποίο σηματοδοτεί την ανάθεση των δώρων στα παιδιά λαμβάνοντας υπόψη τόσο τις προτιμήσεις τους όσο και το διαθέσιμο απόθεμα στην αποθήκη. Κατά το γεγονός αυτό, θα πρέπει να διατρέξετε τις λίστες των παιδιών από όλες τις ηλικιακές κατηγορίες. Επιπλέον, θα πρέπει να ελέγξετε αν το γεωγραφικό διαμέρισμα που διαμένει το παιδί υπάρχει ήδη στη λίστα γεωγραφικών διαμερισμάτων. Συνεπώς, όταν συναντάτε το πρώτο παιδί ενός γεωγραφικού διαμερίσματος θα πρέπει να εισάγετε ένα στοιχείο που να αντιστοιχεί στη λίστα του γεωγραφικού διαμερίσματος αυτού.

Στη συνέχεια, για κάθε παιδί θα πρέπει να εξετάζετε αν υπάρχει διαθέσιμο απόθεμα για κάθε ένα από τα δώρα της προτίμησής του με σειρά προτεραιότητας. Αν ο μετρητής αποθέματος (stock_cnt) έχει τιμή μεγαλύτερη από το 0 τότε το δώρο ανατίθεται στο παιδί και ο μετρητής αποθέματος μειώνεται κατά ένα. Σε διαφορετική περίπτωση, θα πρέπει να εξετάζονται οι υπόλοιπες προτιμήσεις του παιδιού με τον ίδιο τρόπο. Αν καμιά από τις προτιμήσεις του παιδιού δεν μπορεί να ικανοποιηθεί με βάση τα αποθέματα της αποθήκης, τότε ανατίθεται στο παιδί ένα προκαθορισμένο δώρο με συγκεκριμένο αναγνωριστικό (pid = -2). Η εξέταση ύπαρξης αποθέματος θα πρέπει να πραγματοποιείται διασχίζοντας τη λίστα μια μόνο φορά.

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

```
P
    DISTRICTS:
    <did1>
        ASSIGNMENT: <cid1,1, pid1,1> … <cid1,n1, pid1,n1>
    <did2>
        ASSIGNMENT: <cid2,1, pid2,1> … <cid2,n2, pid2,n2>
    ...
   <didk>
       ASSIGNMENT: <cidk,1, pidk,1> … <cidk,nk, pidk,nk>
DONE
```

όπου k είναι ο αριθμός των γεωγραφικών διαμερισμάτων στη λίστα γεωγραφικών διαμερισμάτων, για κάθε j, $1 \leq j \leq k$, n_j είναι το μέγεθος της λίστας ανάθεσης δώρων του j-οστού γεωγραφικού διαμερίσματος, και για κάθε $i \in \{1, \ldots, n_j\}$, $\langle pid_{k,i}, cid_{k,i} \rangle$ είναι το αναγνωριστικό του παιδιού και του δώρου του i-οστού κόμβου στη λίστα ανάθεσης δώρων του γεωγραφικού διαμερίσματος με αναγνωριστικό k.

– **F <cid> <s_degree>**

Γεγονός τύπου *give feedback* το οποίο σηματοδοτεί την εισαγωγή κάποιου βαθμού <s_degree> ικανοποίησης του παιδιού με αναγνωριστικό <cid> για το δώρο που έλαβε. Η τιμή του s_degree είναι ένας ακέραιος αριθμός από το 1 ως το 5, με το 1 να αντιστοιχεί στον ελάχιστο βαθμό ικανοποίησης του παιδιού και το 5 στο μέγιστο. Κατά το γεγονός αυτό, εντοπίζεται το παιδί με αναγνωριστικό <cid> στη λίστα ανάθεσης δώρων και τίθεται ο βαθμός ικανοποίησής του ώστε να είναι ίσος με <s_degree>. Εξαίρεση αποτελεί η περίπτωση που στο παιδί έχει ανατεθεί το προκαθορισμένο δώρο (με pid=-2). Τότε, ο βαθμός ικανοποίησης θα είναι ένα (1) ανεξάρτητα από την τιμή της παραμέτρου <s_degree>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
F <cid> <s_degree> <pid>
DONE
```

όπου <pid> είναι το αναγνωριστικό του δώρου που έλαβε το παιδί με αναγνωριστικό <cid> και έδωσε βαθμό ικανοποίησης <s_degree>.

– **A**

Γεγονός τύπου *analytics* κατά το οποίο εκτυπώνεται η λίστα αποθεμάτων δώρων ταξινομημένη ως προς τη ζήτηση των δώρων. Για το σκοπό αυτό, η λίστα αποθεμάτων των δώρων στην αποθήκη πρέπει να ταξινομείται σε φθίνουσα διάταξη με βάση τη ζήτηση του κάθε δώρου (πεδίο request_cnt). Η λίστα θα πρέπει να ταξινομείται σε χρόνο $O(n^2)$ υλοποιώντας τον αλγόριθμο InsertionSort() σε λίστες, όπου n το πλήθος των στοιχείων της λίστας. Ο αλγόριθμος InsertionSort() έχει διδαχθεί στην Ενότητα 1 για πίνακες και θα πρέπει να τον τροποποιήσετε κατάλληλα ώστε να λειτουργεί πάνω σε λίστες. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
A
    <pid_i> : <request_cnt_i>
    ...
    <pid_n> : <request_cnt_n>
DONE
```

όπου n είναι το μέγεθος της λίστας αποθεμάτων δώρων, για κάθε $i \in \{1, \ldots, n\}$, $<pid_i>$ είναι το αναγνωριστικό του δώρου που αντιστοιχεί στο i-οστό κόμβο στη λίστα αποθεμάτων δώρων και $<request\_cnt_i>$ είναι ο μετρητής ζήτησης του δώρου αυτού.

– **N**

Γεγονός τύπου *new season* το οποίο σηματοδοτεί την έναρξη νέας περιόδου προετοιμασίας του Αγίου Βασιλείου. Για κάθε παιδί κάθε ηλικιακής κατηγορίας, αυξάνουμε την ηλικία του κατά ένα. Αυτό ενδέχεται να προκαλέσει τη μετακίνηση κάποιων παιδιών σε νέα ηλικιακή κατηγορία. Ο αλγόριθμος μετακίνησης των παιδιών κάθε ηλικιακής κατηγορίας στην επόμενη πρέπει να γίνεται σε $O(n_1+n_2)$ χρόνο, όπου n_1 είναι το μέγεθος της λίστας στην οποία θα ανήκει το παιδί μετά το τέλος του γεγονότος αυτού.

Τα παιδιά που υπερβαίνουν το 15ο έτος της ηλικίας τους θα πρέπει να αφαιρεθούν από τη δομή. Αυτό θα πρέπει να πραγματοποιηθεί με μια μόνο διάσχιση της λίστας παιδιών της μεγαλύτερης ηλικιακής κατηγορίας. Επιπρόσθετα, κάθε ένα από τα στοιχεία του πίνακα προτιμήσεών (present_choices) του παιδιού πρέπει να αρχικοποιηθεί με την τιμή 0. Επιπλέον, για κάθε γεωγραφικό διαμέρισμα (district) της λίστας γεωγραφικών διαμερισμάτων, θα πρέπει να αδειάσει η λίστα των παιδιών που ανήκουν σε αυτή. Επίσης, θα πρέπει να διατρέξετε τη λίστα αποθεμάτων των δώρων μια φορά και να πραγματοποιήσετε τις ακόλουθες ενέργειες: (α) να διαγράψετε τις εγγραφές που αντιστοιχούν σε εκείνα τα δώρα για τα οποία ο μετρητής αποθεμάτων είναι 0 και (β) να μηδενίσετε το μετρητή ζήτησης των υπολοίπων.

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

```
N
DONE
```

– **C**

Γεγονός τύπου *clear list of children* το οποίο σηματοδοτεί τη διαγραφή των παιδιών (child) τα οποία δεν έχουν στείλει γράμμα στον Άγιο Βασίλειο από την τελευταία αλλαγή της σεζόν. Συγκεκριμένα, για κάθε παιδί κάθε ηλικιακής κατηγορίας, θα πρέπει να γίνεται έλεγχος στον πίνακα των προτιμήσεών του (present_choices) και αν όλες οι θέσεις έχουν μηδενικές τιμές, τότε το παιδί θα πρέπει να διαγράφεται από τη δομή αφού αυτό σημαίνει ότι δεν έχει στείλει στο τρέχον έτος γράμμα στον Άγιο Βασίλειο. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
C
    0-3:   <cid1,1> … <cid1,n1>
    4-7:   <cid2,1> … <cid2,n2>
    8-11:  <cid3,1> … <cid3,n3>
    12-15: <cid4,1> … <cid4,n4>
DONE
```

όπου για κάθε $k \in \{1, \ldots, 4\}$, n_k είναι ο αριθμός των κόμβων στη λίστα παιδιών της ηλικιακής

κατηγορίας η οποία δεικτοδοτείται από την k-οστή θέση του πίνακα Age_categories και για κάθε i ∈ {1, …, n}, <$cid_{k,i}$> είναι το αναγνωριστικό του παιδιού που αντιστοιχεί στον i-οστό κόμβο της λίστας παιδιών της ηλικιακής κατηγορίας αυτής.

– **S <pid>**

Γεγονός τύπου *search present* το οποίο σηματοδοτεί την αναζήτηση του δώρου με αναγνωριστικό <pid> στη λίστα αποθεμάτων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
S <pid>
    <stock_cnt>, <request_cnt>
DONE
```

όπου <stock_cnt> και <request_cnt> είναι ο μετρητής αποθέματος και ο μετρητής ζήτησης του δώρου με αναγνωριστικό <pid>, αντίστοιχα. Αν δεν υπάρχει το δώρο στη λίστα, θα πρέπει να εμφανίζεται και στους δυο μετρητές η τιμή 0.

– **H**

Γεγονός τύπου *print child* το οποίο σηματοδοτεί την εκτύπωση της λίστας παιδιών όλων των ηλικιακών κατηγοριών. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
H
    0-3:   <cid1,1> … <cid1,n1>
    4-7:   <cid2,1> … <cid2,n2>
    8-11:  <cid3,1> … <cid3,n3>
    12-15: <cid4,1> … <cid4,n4>
DONE
```

όπου για κάθε k ∈ {1, …, 4}, n_k είναι ο αριθμός των κόμβων στη λίστα παιδιών της ηλικιακής κατηγορίας η οποία δεικτοδοτείται από την k-οστή θέση toy πίνακα Age_categories και για κάθε i ∈ {1, …, n}, <$cid_{k,i}$> είναι το αναγνωριστικό του παιδιού που αντιστοιχεί στον i-οστό κόμβο της λίστας παιδιών της ηλικιακής κατηγορίας αυτής.

– **I**

Γεγονός τύπου *print district* το οποίο σηματοδοτεί την εκτύπωση των αναγνωριστικών των παιδιών και των αναγνωριστικών των δώρων που έχουν λάβει από τον Άγιο Βασίλειο που κατοικούν σε όλα τα γεωγραφικά διαμερίσματα. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
I

    DISTRICTS:

    <did_1>
        ASSIGNMENT: <cid_1,1, pid_1,1> … <cid_1,n1, pid_1,n1>

    <did_2>
        ASSIGNMENT: <cid_2,1, pid_2,1> … <cid_2,n2, pid_2,n2>

    ...

   <did_k>
       ASSIGNMENT: <cid_k,1, pid_k,1> … <cid_k,nk, pid_k,nk>

DONE
```

όπου k είναι ο αριθμός των γεωγραφικών διαμερισμάτων στη λίστα γεωγραφικών διαμερισμάτων, για κάθε j, $1 \leq j \leq k$, n_j είναι το μέγεθος της λίστας ανάθεσης δώρων του j-οστού γεωγραφικού διαμερίσματος, και για κάθε $i \in \{1, \ldots, n_j\}$, $<pid_{k,i}, cid_{k,i}>$ είναι το αναγνωριστικό του παιδιού και του δώρου του i-οστού κόμβου στη λίστα ανάθεσης δώρων του γεωγραφικού διαμερίσματος με αναγνωριστικό k.

– **T**

Γεγονός τύπου *print stock* το οποίο σηματοδοτεί την εκτύπωση των αναγνωριστικών των αδιάθετων δώρων στη λίστα αποθεμάτων των δώρων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
T

Stock = <pid_1>, …, <pid_n>

DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα αποθεματικών και για κάθε $i \in \{1, \ldots, n\}$, $<pid_i>$ είναι το αναγνωριστικό του δώρου που αντιστοιχεί στο i-οστό κόμβο στη λίστα αυτή.

– **D (BONUS)**

Γεγονός τύπου *advanced analytics* κατά το οποίο εκτυπώνεται το δημοφιλέστερο δώρο σε κάθε γεωγραφικό διαμέρισμα. Θα πρέπει να διατρέξετε τη λίστα γεωγραφικών διαμερισμάτων και για κάθε γεωγραφικό διαμέρισμα, θα πρέπει να κάνετε κατάλληλη ταξινόμηση στη λίστα ανάθεσης δώρων που αντιστοιχεί σε αυτό το γεωγραφικό διαμέρισμα ώστε να μπορείτε να βρείτε το δημοφιλέστερο δώρο στο γεωγραφικό διαμέρισμα χρησιμοποιώντας απλά δύο μετρητές και διασχίζοντας τη λίστα (αφού γίνει η ταξινόμηση) μόνο μια φορά. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
D
        Districts =
    <did_i> : <pid_1>
    …
    <did_n> : <pid_n>
DONE
```

όπου <did> το αναγνωριστικό και <pid> το αναγνωριστικό του δημοφιλέστερου δώρου του i-οστού κόμβου στη λίστα γεωγραφικών διαμερισμάτων για κάθε $i \in \{1, \ldots, n\}$.

Βαθμολογία

B	Buy present	8
L	Letter received	14
P	Prepare presents	14
F	Give feedback	10
A	Analytics	14
N	New season	14
C	Clear list of children	8
S	Search present	5
H	Print child	1
I	Print district	1
T	Print stock	1
D	Advanced analytics	15
Δεν κάνει compile		5
Δεν τρέχει και δεν τρέχουν τα test-files		5

Δομές Δεδομένων

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

```
#define M 3
#define N 4

struct child {
    int cid;
    int age;
    int did;
    int present_choices[M];
    struct child *next; /* Singly-linked, sorted by id */
};

struct present {
    int pid;
    int stock_cnt;
    int request_cnt;
    struct present *prev;       /* Double-linked with sentinel, unsorted, */
                                /* a variant of the Transpose Heuristic is applied during LookUp()*/
    struct present *next;
};

struct present_assign {
    int cid;
    int pid;
    int s_degree;
    struct present_assign *next; /* Singly-linked, unsorted */
};
```

```
struct district {
    int did;
    struct present_assign *assignL;
    struct district *next; /* Singly-linked, sorted by did */
};

struct child *Age_categories[N];

struct present *stock_list; /*global variable which is a pointer to the beginning of the list “stock”*/
struct present *stock_sentinel; /*global variable of pointer to the sentinel node of the list “stock” */
```