HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2017-2018
Διδάσκουσα: Παναγιώτα Φατούρου
Προγραμματιστική Εργασία - 2ο Μέρος

Ημερομηνία Παράδοσης: Παρασκευή, 22 Δεκεμβρίου 2017, ώρα 23:59.

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

Γενική Περιγραφή

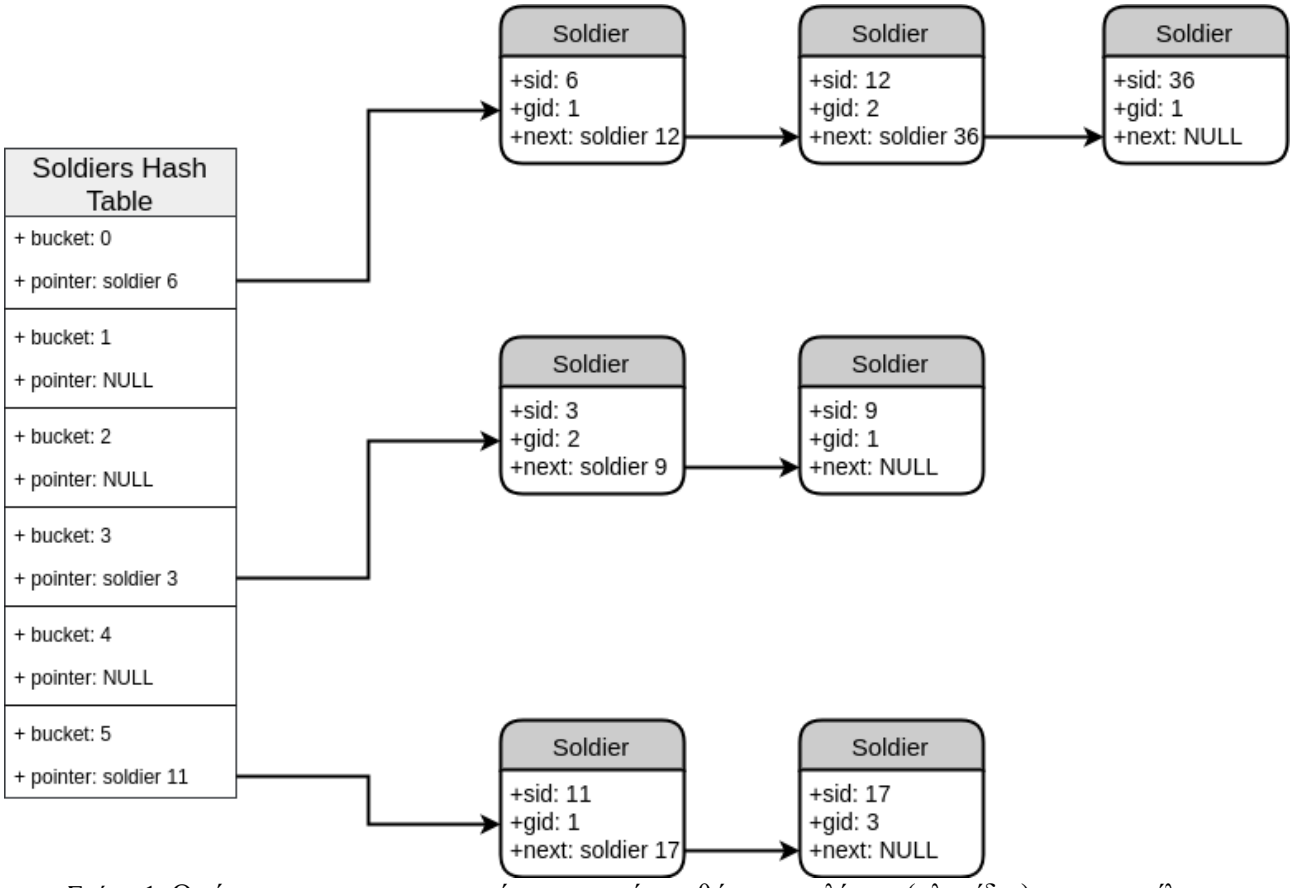
Στο δεύτερο μέρος της εργασίας αυτής, καλείστε και πάλι να υλοποιήσετε ένα πρόγραμμα που θα προσομοιώνει τις τελευταίες ημέρες του τρωικού πολέμου, μεταξύ των Αχαιών και των Τρώων, όπως αυτός περιγράφεται στην Ιλιάδα χρησιμοπιώντας ωστόσο διαφορετικές δομές δεδομένων από αυτές τις πρώτης φάσης.

Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Ο κατάλογος των εμπόλεμων (Ραψωδία Β, 22η ημέρα). «Στη διάρκεια των ετοιμασιών, ο ποιητής επικαλείται τη βοήθεια των μουσών για να παραθέσει τον μακρύ κατάλογο των εμπόλεμων» (Ομηρικά Έπη: Ιλιάδα Β' Γυμνασίου). Ένα από τα καθήκοντα της εργασίας αυτής είναι η υλοποίηση της απογραφής. Πληροφορίες για τους εμπόλεμους του στρατοπέδου των Αχαιών αποθηκεύονται σε έναν *πίνακα κατακερματισμού*. Ο πίνακας αυτός ονομάζεται **πίνακας κατακερματισμού απογραφής** (*registration hash table*). Για την επίλυση των συγκρούσεων θα πρέπει να χρησιμοποιήσετε την τεχνική των *ξεχωριστών αλυσίδων (seperate chaining)* και ως συνάρτηση κατακερματισμού, την $h(k)=k \bmod N$, όπου k είναι το κλειδί και N το μέγεθος του πίνακα κατακερματισμού. Για την υλοποίηση του πίνακα κατακερματισμού, παρέχεται η global μεταβλητή *max_soldiers_g*, η οποία δίνει τον μέγιστο αριθμό στρατιωτών που θα συμμετέχουν στον πόλεμο. Κάθε κόμβος μιας αλυσίδας αντιστοιχεί σε ένα στρατιώτη και αποτελεί μια εγγραφή τύπου soldier με τα ακόλουθα πεδία:

- **sid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον εμπόλεμο άνδρα.
- **gid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον βασιλιά/στρατηγό στον οποίο υπακούει ο άνδρας.
- **next:** Δείκτης (τύπου soldier) στον επόμενο κόμβο της αλυσίδας στην οποία ανήκει.

Στο Σχήμα 1 παρουσιάζεται ο πίνακας κατακερματισμού στρατιωτών (soldiers hash table).



Σχήμα 1: Ο πίνακας κατακερματισμού στρατιωτών καθώς και οι λίστες (αλυσίδες) για την επίλυση συγκρούσεων.

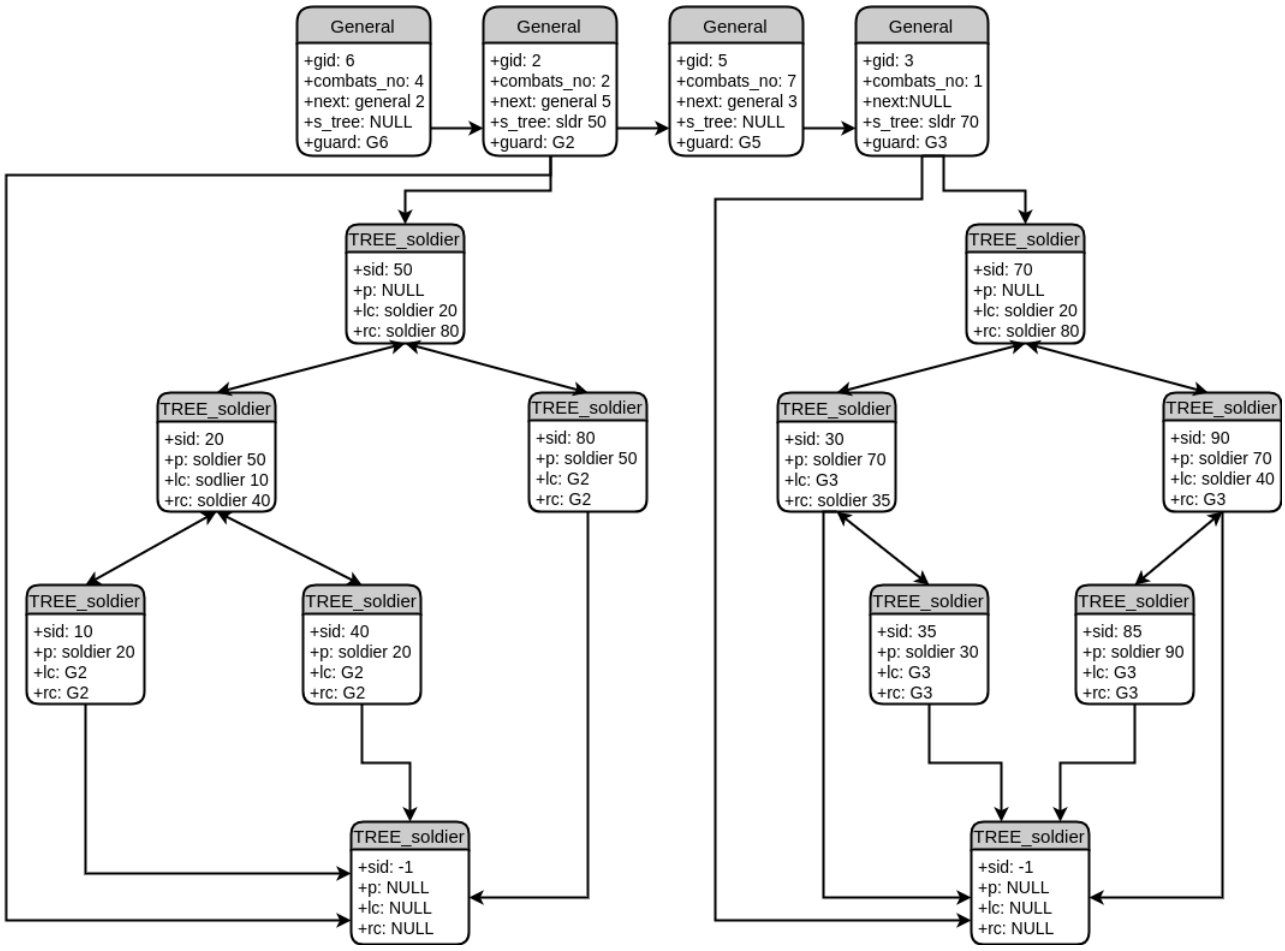
Τα στρατεύματα των Αχαιών είναι οργανωμένα σε παρατάξεις των οποίων ηγούνταν γενναίοι βασιλιάδες και στρατηγοί. Ο κάθε βασιλιάς/στρατηγός έχει το δικό του στράτευμα (σύνολο στρατιωτών) που έχει στρατοπεδεύσει μπροστά από τα καράβια που μετέφεραν τους άνδρες στην Τροία. Πληροφορίες για τους βασιλιάδες/στρατηγούς αποθηκεύονται σε μια απλά συνδεδεμένη, μη ταξινομημένη λίστα. Η λίστα αυτή ονομάζεται λίστα στρατηγών (generals list). Ο κάθε κόμβος της λίστας αυτής, αποτελεί μια εγγραφή τύπου general με τα ακόλουθα πεδία:

- **gid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον στρατηγό.
- **combats_no:** Αριθμός (τύπου int) που αντιπροσωπεύει τον αριθμό των μαχών στις οποίες έχει συμμετάσχει ο στρατηγός.
- **soldiers_S:** Δείκτης τύπου TREE_soldier που δείχνει στον κόμβο φρουρό του δένδρου στρατιωτών του εν λόγω στρατηγού.
- **soldiers_R:** Δείκτης τύπου TREE_soldier που δείχνει στη ρίζα του δέντρου ανδρών που υπακουούν στον στρατηγό. Το δέντρο αυτό είναι *δυαδικό, έχει δείκτες προς τους πατρικούς κόμβους, είναι ταξινομημένο με βάση το αναγνωριστικό του στατιώτη* και ονομάζεται **δέντρο στρατιωτών**. Κάθε στοιχείο του δέντρου αυτού, αποτελεί μια εγγραφή τύπου TREE_soldier με τα ακόλουθα πεδία:

- **sid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον εμπόλεμο άνδρα.
- **rc:** Δείκτης (τύπου TREE_soldier) που δεικτοδοτεί το δεξί παιδί του κόμβου.
- **lc:** Δείκτης (τύπου TREE_soldier) που δεικτοδοτεί το αριστερό παιδί του κόμβου.
- **p:** Δείκτης (τύπου TREE_soldier) που δεικτοδοτεί τον πατέρα του κόμβου.

- **next:** Δείκτης (τύπου general) που δεικτοδοτεί τον επόμενο κόμβο στη λίστα στρατηγών.

Στο Σχήμα 2 παρουσιάζεται η λίστα στρατηγών και το δέντρο των στρατιωτών που δεικτοδοτείται από κάθε στοιχείο της λίστας στρατηγών.



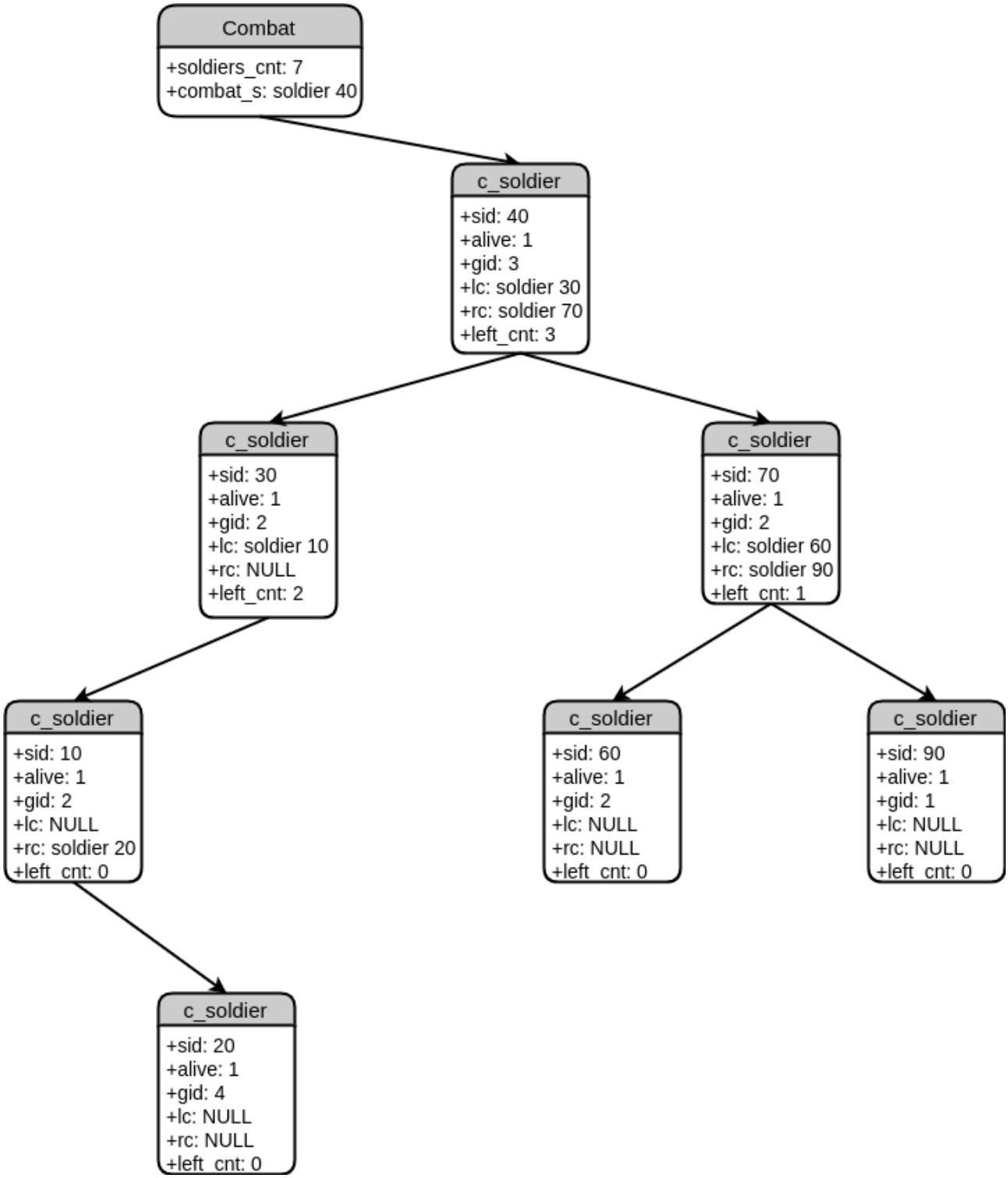
Σχήμα 2: Η μη ταξινομημένη, απλά συνδεδεμένη λίστα στρατηγών. Ο κάθε κόμβος (general) αντοιστοιχεί σε έναν βασιλιά/στρατηγό και περιέχει έναν δείκτη στη ρίζα του δέντρου στρατιωτών. Το δέντρο στρατιωτών του κάθε στρατηγού είναι ταξινομημένο με βάση το αναγνωριστικό των στρατιωτών και έχει κόμβο φρουρό.

Μάχες στις Ραψωδίες Δ και Ε (22η ημέρα), Θ (25η ημέρα), Λ, Μ και Ν (26η ημέρα, οι μάχες γύρω από το τείχος των Αχαιών), Π και Ρ (27η ημέρα, ο Πάτροκλος οδηγεί τους Μυρμηδόνες στη μάχη), Φ (ο Αχιλλέας επιστρέφει στη μάχη).

Ένα ακόμη από τα καθήκοντα της εργασίας αυτής είναι να υλοποιήσετε τις μάχες. Σε κάθε μάχη συμμετέχουν κάποιοι από τους στρατηγούς και τους στρατιώτες τους. Οι στρατιώτες που μάχονται αποθηκεύονται σε μια δομή τύπου combat που περιέχει τα ακόλουθα πεδία:

- **soldiers_cnt**: Ακέραιος που αντιπροσωπεύει το συνολικό αριθμό των στρατιωτών που συμμετέχουν στη μάχη
- **combat_s**: Δείκτης (τύπου c_soldier) που δείχνει στη ρίζα ενός *δυαδικού, ταξινομημένου ως προς το αναγνωριστικό των στρατιωτών,* δέντρου, που ονομάζεται **δέντρο μάχης**. Το κάθε στοιχείο του δέντρου αποτελεί μια εγγραφή τύπου c_soldier με τα ακόλουθα πεδία:
    - **sid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον μαχόμενο άνδρα.
    - **alive**: Δυαδική μεταβλητή που αντιπροσωπεύει την κατάσταση του μαχόμενου άνδρα. Η τιμή 1 σηματοδοτεί ότι ο άνδρας είναι ζωντανός, ενώ η τιμή 0 το αντίθετο.
    - **gid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον στρατηγό στον οποίο υπακούει ο μαχόμενος άνδρας.
    - **rc**: Δείκτης (τύπου c_soldier) που δεικτοδοτεί το δεξί παιδί του κόμβου.
    - **lc**: Δείκτης (τύπου c_soldier) που δεικτοδοτεί το αριστερό παιδί του κόμβου.
    - **left_no**: Το πλήθος των κόμβων του αριστερού υποδέντρου του κόμβου.

Στο Σχήμα 3 παρουσιάζεται η δομή combat που περιέχει το δέντρο μάχης και το μετρητή στρατιωτών που συμμετέχουν στη μάχη.



Σχήμα 3: Η δομή combat που περιέχει το δέντρο μάχης και το μετρητή στρατιωτών που συμμετέχουν στη μάχη.

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> <gid>

Γεγονός κατά το οποίο ένας στρατιώτης απογράφεται **(Ραψωδία Β[48-877], 22η ημέρα)**. Κατά το γεγονός αυτό, γίνεται εισαγωγή ενός νέου στρατιώτη στον πίνακα κατακερματισμού απογραφής. Ο στρατιώτης θα έχει αναγνωριστικό <sid> και θα δρά υπό τις διαταγές του στρατηγού με αναγνωριστικό <gid>. Για την εισαγωγή στον πίνακα κατακερματισμού,θα πρέπει να χρησιμοποιήσετε τη συνάρτηση κατακερματισμού που περιγράφηκε παραπάνω, καθώς και την τεχνική των ξεχωριστών αλυσίδων για την επίλυση συγκρούσεων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
R <sid> <gid>
    Soldiers Hash Table:
    <sid1,1:gid1,1>, <sid1,2:gid1,2>, ..., <sid1,n1:gid1,n1>,
    <sid2,1:gid2,1>, <sid2,2:gid2,2>, ..., <sid2,n2:gid2,n2>,
    ...
    <sidk,1:gidk,1>, <sidk,2:gidk,2>, ..., <sidk,nk:gidk,nk>
DONE
```

όπου για κάθε i, $1 \leq i \leq k$, n_i είναι το πλήθος των κόμβων της i-οστής αλυσίδας του πίνακα κατακερματισμού και για κάθε j, $1 \leq j \leq n_i$, $sid_{i,j}$ είναι το αναγνωριστικό του j-οστού κόμβου της i-οστής αλυσίδας και $gid_{i,j}$ το αναγνωριστικό του στρατηγού στον οποίο υπακούει ο κόμβος (στρατιώτης) αυτός.

- G <gid>

Γεγονός που υποδηλώνει ότι ο βασιλιάς/στρατηγός με αναγνωριστικό <gid> συμμετέχει στην εκστρατεία. Κατά το γεγονός αυτό γίνεται εισαγωγή ενός κόμβου τύπου general στη λίστα στρατηγών. Αρχικά η λίστα στρατιωτών του στρατηγού θα είναι κενή (υπάρχει μόνο ο κόμβος φρουρός). Η εισαγωγή πρέπει να γίνεται με τη μικρότερη χρονική πολυπλοκότητα, δεδομένου ότι το <gid> που ακολουθεί κάθε event τύπου G στα test files είναι μοναδικό. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
G <gid>
    Generals = <gid1>, <gid2>, ..., <gidn>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα στρατηγών και για κάθε $i \in \{1, \ldots, n\}$, $<gid_i>$ είναι το

αναγνωριστικό του στρατηγού που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής.

- D

Γεγονός τύπου *distribute soldiers* το οποίο σηματοδοτεί βραδινή ανάπαυση. Οι στρατιώτες αναπαύονται στα στρατόπεδα όπου ανήκουν (ανάλογα με το στρατηγό στον οποίο υπάγονται). Κατά το γεγονός αυτό, θα πρέπει, διασχίζοντας τον πίνακα κατακερματισμού απογραφής, να δημιουργηθούν τα δέντρα στρατιωτών που υπάγονται σε κάθε στρατηγό. Αυτό μπορεί να γίνει εξετάζοντας το πεδίο gid της κάθε εγγραφής του πίνακα κατακερματισμού και στη συνέχεια να γίνεται αναζήτηση του στρατηγού με αναγνωριστικό gid στη λίστα στρατηγών. Στη συνέχεια θα γίνεται εισαγωγή ενός struct τύπου TREE_soldier το οποίο θα αντιστοιχεί στον στρατιώτη στο δέντρο στρατιωτών του στρατηγού. Μετά από κάθε εισαγωγή, το δέντρο στρατιωτών του κάθε στρατηγού πρέπει να είναι ταξινομημένο με βάση το αναγνωριστικό των στρατιωτών. Στο τέλος του γεγονότος, για όλους τους στρατιώτες του πίνακα κατακερματισμού στρατιωτών, πρέπει να υπάρχουν αντίστοιχες εγγραφές στα κατάλληλα δέντρα στρατιωτών των στρατηγών.

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

```
D

    GENERALS:
    <gid1>: <sid1,1> . . . <sid1,n1>
    <gid2>: <sid2,1> . . . <sid2,n2>
    ...
    <gidk>: <sidk,1> . . . <sidk,nk>

DONE
```

όπου για κάθε i, $1 \leq i \leq k$, n_i είναι το πλήθος των κόμβων του δέντρου στρατιωτών του i-οστού κόμβου της λίστας στρατηγών, και για κάθε j, $1 \leq j \leq n_i$, και $<sid_{i,j}>$ είναι το αναγνωριστικό του j-οστού κόμβου στο δέντρο στρατιωτών του i-οστού στρατηγού (βάσει της ενδοδιατεταγμένης διάσχισης).

- M <gid$_1$> <gid$_2$>

Γεγονός που σηματοδοτεί την αποχώρηση του Αχιλλέα από τη μάχη (μετά τη φιλονικία του με τον Αγαμέμνονα), ενώ ο Πάτροκλος πείθει τον Αχιλλέα να οδηγήσει αυτός τους Μυρμηδόνες στη μάχη.

Θεωρήστε ότι ο Αχιλλέας έχει αναγνωριστικό $<gid_1>$ και ο Πάτροκλος έχει αναγνωριστικό $<gid_2>$. Κατά το γεγονός αυτό πρέπει να διαγράψετε τον κόμβο με αναγνωριστικό $<gid_1>$ από τη λίστα στρατηγών. Επιπλέον, οι άνδρες του Αχιλλέα θα μεταφερθούν στο δέντρο στρατιωτών του Πάτροκλου (ο οποίος έχει αναγνωριστικό $<gid_2>$)[1]. Η συνένωση των δύο δένδρων πρέπει να επιτευχθεί με χρονική πολυπλοκότητα O(n), όπου n είναι το άθροισμα των ανδρών του Αχιλλέα και του Πάτροκλου. Αυτό θα πραγματοποιηεί με δύο βοηθητικούς πίνακες, όπου ο πρώτος θα αποθηκεύει τους άνδρες του Αχιλλέα ενώ ο δεύτερος τους

[1] *Στην Ιλιάδα, ο Πάτροκλος οδηγεί τους Μυρμηδόνες στη μάχη φορώντας την πανοπλία του Αχιλλέα. Για την εξυπηρέτηση των εκπαιδευτικών στόχων του μαθήματος ωστόσο, θεωρήστε ότι ο Πάτροκλος είχε και δικούς του άνδρες οι οποίοι συμμετείχαν στη μάχη μαζί με τους Μυρμηδόνες.*

άνδρες του Πάτροκλου. Οι πίνακες θα πρέπει να είναι ταξινομημένοι ως προς το αναγνωριστικό των στρατιωτών. Στη συνέχεια θα πρέπει να δημιουργείτε ένα τρίτο πίνακα που θα περιέχει τους άνδρες και των δύο στρατηγών και πάλι ταξινομημένους ως προς το αναγνωριστικό τους. Η δημιουργία του τρίτου πίνακα (που είναι ταξινομημένος) θα πρέπει να επιτυγχάνεται σε χρόνο O(n). Στη συνέχεια, βάσει του τρίτου πίνακα θα πρέπει να δημιουργείτε εκ νέου το δένδρο ανδρών του Πάτροκλου το οποίο θα περιέχει όλους τους άνδρες και των δύο στρατηγών. Προσοχή το δένδρο αυτό πρέπει να έχει ύψος O(logn). Μετά το τέλος της διαδικασίας, το δέντρο στρατιωτών του Πάτροκλου πρέπει να είναι ταξινομημένο με βάση το αναγνωριστικό των στρατιωτών.

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

```
M <gid1> <gid2>

    GENERALS:
    <gid1>: <sid1,1> . . . <sid1,n1>
    <gid2>: <sid2,1> . . . <sid2,n2>
    ...
    <gidk>: <sidk,1> . . . <sidk,nk>
DONE
```

όπου για κάθε i, $1 \leq i \leq k$, n_i είναι το πλήθος των κόμβων του δέντρου στρατιωτών του i-οστού κόμβου της λίστας στρατηγών, και για κάθε j, $1 \leq j \leq n_i$, και $<sid_{i,j}>$ είναι το αναγνωριστικό του j-οστού κόμβου στο δέντρο στρατιωτών του i-οστού στρατηγού (βάσει της ενδοδιατεταγμένης διάσχισης).

- P <gid1> <gid2>

Γεγονός τύπου *prepare for battle* το οποίο σηματοδοτεί την επιλογή των ανδρών που θα συμμετέχουν στην επόμενη μάχη. Στη μάχη θα λάβουν μέρος οι στρατηγοί με αναγνωριστικά $<gid_1>$ και $<gid_2>$ με τους άνδρες τους. Επίσης, θα αυξήσετε τον μετρητή combats_no, που υποδηλώνει τον αριθμό των μαχών στις οποίες έχει συμμετάσχει ο κάθε στρατηγός.

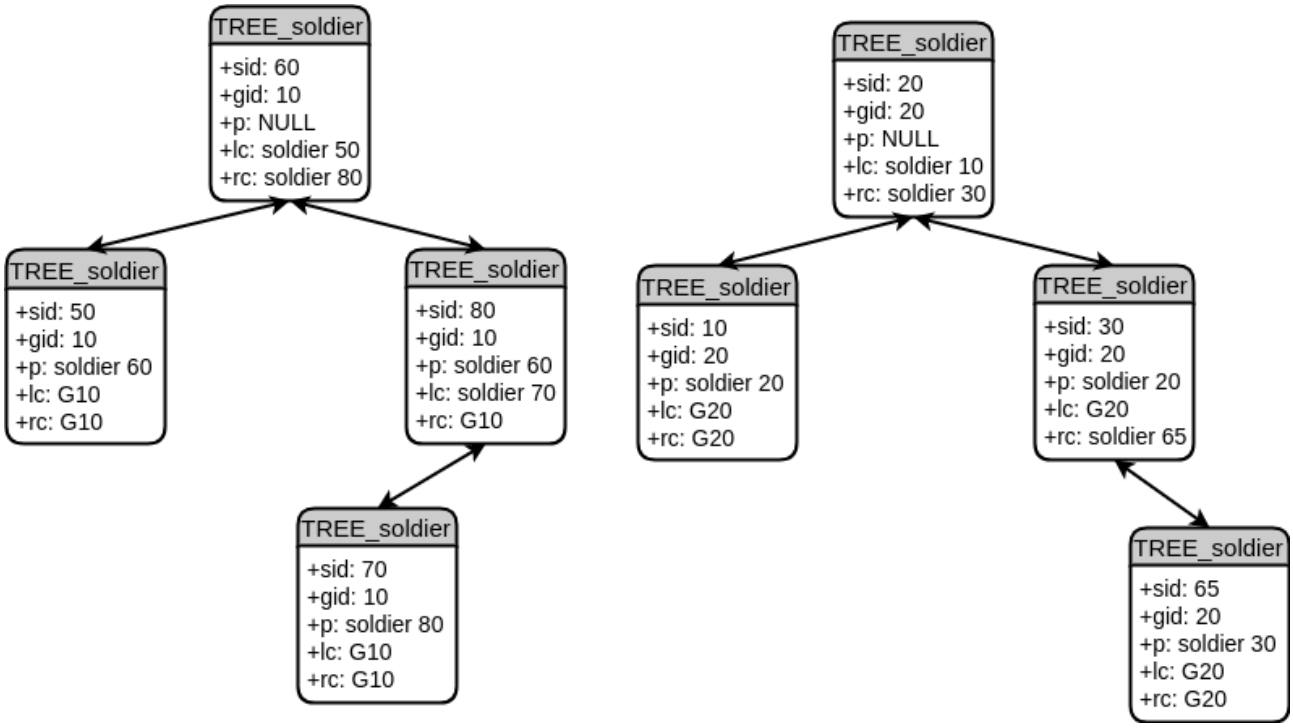
Οι στρατιώτες που θα πολεμήσουν, θα πρέπει να αποθηκεύονται στο δέντρο μάχης (combat_s) της εγγραφής combat. Κάθε στοιχείο του δέντρου αυτού περιέχει μια εγγραφή τύπου c_soldier. Κάθε φορά που εισάγεται ένας κόμβος σ' αυτό το δέντρο, πρέπει να ενημερώνετε κατάλληλα τον μετρητή soldiers_cnt της εγγραφής combat. Ο τρόπος με τον οποίο θα εισάγονται οι στρατιώτες του κάθε στρατηγού στο δέντρο μάχης, είναι ο ακόλουθος:

Ξεκινώντας από το στρατιώτη με το μικρότερο αναγνωριστικό, στο δέντρο στρατιωτών του στρατηγού με αναγνωριστικό $<gid_1>$ και από το στρατιώτη με το μεγαλύτερο αναγνωριστικό στο δένδρο στρατιωτών του στρατηγού με αναγνωριστικό $<gid_2>$, θα τοποθετείται κόμβους από τα δύο δένδρα εναλλάξ στο δένδρο της μάχης. Συγκεκριμένα, θα πρέπει να διασχίσετε ταυτόχρονα τα δύο δέντρα στρατιωτών, τοποθετώντας σε κάθε βήμα στο δέδνρο μάχης είτε τον επόμενο κόμβο στην ενδοδιατεταγμένη διάσχιση του δένδρου στρατιωτών του στρατηγού $<gid_1>$, είτε τον προηγούμενο κόμβο στην ενδοδιατεταγμένη διάσχιση του δένδρου στρατιωτών του στρατηγού $<gid_2>$. Για να υλοποιηθεί αυτό, θα χρειαστεί να δημιουργήσετε τις συνάρτησεις *inorder_successor() και inorder_predeccesor(),* οι οποίες θα σας οδηγούν στον επόμενο ή στον

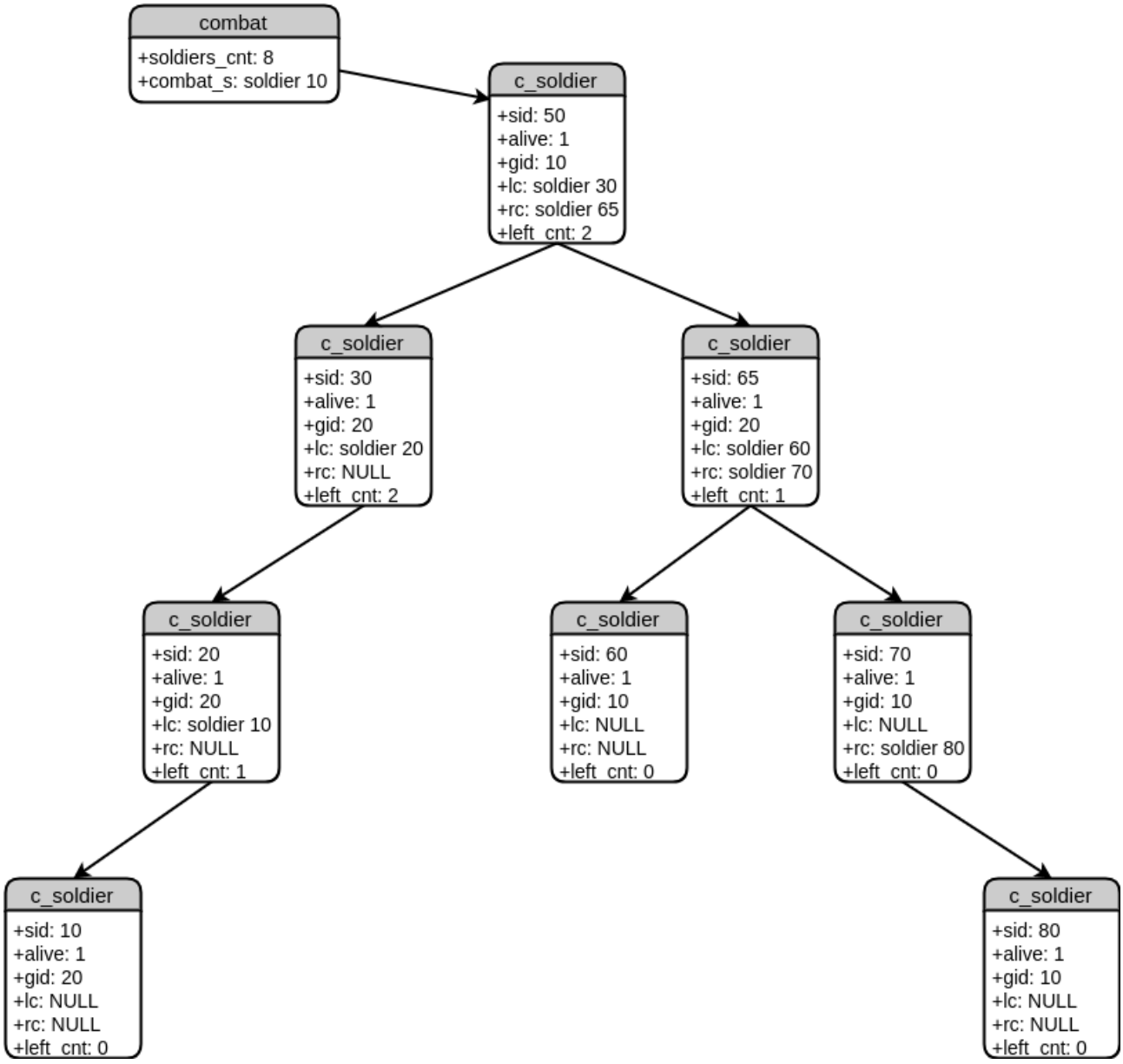
προηγούμενο κόμβο στην ενδοδιατεταγμένη διάσχιση σε κάθε δένδρο αντίστοιχα (δηλαδή η μετάβαση από τον ένα κόμβο στον άλλο γίνεται χρησιμοποιώντας είτε την inorder_successor() ή την inorder_predessecor(), ανάλογα με το δένδρο από όπου θα επισκεφτείτε τον επόμενο άνδρα που θα εισαχθεί στο δένδρο της μάχης). Για κάθε κόμβο που επισκέπτεσθε, θα εισάγετε ένα αντίστοιχο struct τύπου c_soldier στο δέντρο μάχης. Μετά από κάθε εισαγωγή στρατιώτη από το δέντρο στρατιωτών ενός στρατηγού, η επόμενη εισαγωγή θα γίνεται από το δέντρο στρατιωτών του άλλου στρατηγού. Με άλλα λόγια, οι εισαγωγές θα γίνονται εναλλάξ.

Για κάθε στρατιώτη που εισάγετε στο δέντρο μάχης της εγγραφής combat, θα πρέπει να θέτετε την τιμή alive = 1 αφού ο στρατιώτης είναι ζωντανός και να ανανεώνετε την τιμή του μετρητή *left_cnt* στους κόμβους που πρέπει.

Στην εικόνα 4 παρουσιάζεται ένα παράδειγμα με τα δέντρα στρατιωτών των στρατηγών με αναγνωριστικά 10 και 20, που θα συμμετέχουν στη μάχη. Η εικόνα 5 παρουσιάζει το αποτέλεσμα της εισαγωγής των στρατιωτών των παραπάνω στρατηγών στο δέντρο μάχης, σύμφωνα με τον αλγόριθμο που περιγράφεται παραπάνω.



Σχήμα 4 Τα δέντρα στρατιωτών των στρατηγών με αναγνωριστικά 10 και 20.



Σχήμα 5 Το δέντρο μάχης όπως προκύπτει απ' τη συνένωση των δέντρων των στρατιωτών που παρουσιάζονται στην εικόνα 4. Οι εισαγωγές στο δένδρο αυτό έγιναν με την ακόλουθη σειρά: 50, 65, 60, 30, 70, 20, 80, 10.

Επιπρόσθετα, θα πρέπει να αυξάνετε τους κατάλληλους μετρητές combats_no, που υποδηλώνουν τον αριθμό των μαχών στις οποίες κάθε ένας από τους δύο στρατηγούς έχει συμμετάσχει. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
P <gid1> <gid2>
    Combat soldiers: <sid1>, <sid2> . . . <sidn>
DONE
```

όπου n είναι το πλήθος των κόμβων του δέντρου μάχης και για κάθε i, $1 \leq i \leq n$, $<sid_i>$ είναι το αναγνωριστικό του i-οστού στρατιώτη στο δέντρο μάχης (βάση της ενδοδιατεταγμένης διάσχισης).

- B <god_favor> <bit_stream>

Γεγονός που σηματοδοτεί την προσομοίωση της μάχης (δείτε π.χ. Ραψωδίες Δ, Ε, Λ, Μ και Ν).

Στο γεγονός αυτό (που είναι τύπου *battle)* σηματοδοτείται η προσομοίωση της μάχης. Κατά τη διάρκεια της μάχης κάποιοι από τους μαχόμενους πεθαίνουν. Το μέγεθος απωλειών επηρεάζεται από την εύνοια των θεών προς την πλευρά των Ελλήνων όπως ορίζεται από την παράμετρο <god_favor>.

Στην περίπτωση που οι Θεοί δεν ευνοούν τα στρατεύματα των Ελλήνων (π.χ. στις μάχες της 25ης και 26ης ημέρας που συμπεριλαμβάνουν και τις μάχες γύρω από τα τείχη των Αχαιών, Ραψωδίες Λ,Μ και Ν), τότε οι απώλειες ανέρχονται στο 40% των στρατιωτών που συμμετέχουν στη μάχη. Χρησιμοποιώντας τον μετρητή soldiers_cnt της εγγραφής combat, θα υπολογίσετε τον αριθμό των στρατιωτών που θα χαθούν στη μάχη. Στη συνέχεια θα διασχίσετε το δέντρο μάχης (combat_s) της εγγραφής combat και θα ορίσετε το 40% των στρατιωτών του δέντρου ως νεκρούς (θα θέσετε στο πεδίο alive του c_soldier την τιμή 0). Αυτό θα πρέπει να επιτευχθεί σε χρόνο O(m), όπου m είναι ο αιρθμός των ανδρών που θα πεθάνουν, λαμβάνοντας υπ' όψιν την τιμή του πεδίου *left_no* σε κάθε κόμβο.

Στην περίπτωση που οι Θεοί ευνοούν τα στρατεύματα των Ελλήνων (π.χ. οι μάχες στις Ραψωδίες Δ, Ε, Υ και Φ), τότε οι απώλειες είναι λιγότερες. Σε αυτήν την περίπτωση ο τρόπος που θα επιλέξετε ποιοί στρατιώτες θα χάσουν τη ζωή τους οι οι στρατιώτες που αντιστοιχούν σε ένα μονοπάτι του δένδρου μάχης το οποίο θα επιλέγετε κάθε φορά ως εξής:

Θα χρησιμοποιήσετε την παράμετρο <bit_stream>, η οποία έχει τη μορφή μίας σειράς από bits. Τα bits αυτά εκφράζουν το μονοπάτι που θα ακολουθήσετε στο δέντρο μάχης, με σκοπό τον θάνατο του κάθε στρατιώτη που επισκέπτεστε. Θα διαβάζετε τα bits αυτά με τη σειρά και κάθε φορά που το τρέχον bit είναι 0, θα ακολουθείτε τον lc δείκτη του τρέχοντος κόμβου στο δέντρο μάχης, ενώ αν το bit είναι 1, θα ακολουθείτε τον rc δείκτη. Για κάθε κόμβο που επισκέπτεστε, θα αλλάζετε την τιμή της μεταβλητής alive σε 0, δηλώνοντας έτσι τον θάνατο του αντίστοιχου στρατιώτη.

Σε περίπτωση που φτάσετε σε φύλλο και περισσεύουν και άλλα bits στο bit_stream, ο αλγόριθμος τερματίζει.

Στην περίπτωση που έχετε χρησιμοποιήσει όλα τα bits του bit_stream και ακόμα δεν έχετε φτάσει σε κάποιο φύλλο, ο αλγόριθμος θα πρέπει να χρησιμοποιεί το ίδιο bit_stream από την αρχή και να συνεχίζει τη διάσχιση (επαναχρησιμοποιώντας δηλαδή το bit stream), όπως περιγράφτηκε παραπάνω.

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

```
B <god_favor> <bit_stream>

    Combat soldiers: <sid1:alive1>, <sid2:alive2>, . . . <sidn:aliven>
DONE
```

όπου n είναι το πλήθος των κόμβων του δέντρου μάχης και για κάθε i, $1 \leq i \leq n$, $<sid_i>$ και $<alive_i>$ είναι το αναγνωριστικό και η κατάσταση του στρατιώτη που αντιστοιχεί στον i-οστό κόμβο του δέντρου μάχης (βάσει της ενδοδιατεταγμένης διάσχισης).

- U

Ανακωχή και συλλογή νεκρών από το πεδίο της μάχης (23η ημέρα, Ραψωδία Η381-432). Γεγονός κατά το οποίο πραγματοποιείται συλλογή των νεκρών από το πεδίο της μάχης. Στο γεγονός αυτό θα πρέπει να διαγράφονται οι στρατιώτες που έχασαν τη ζωή τους στη μάχη από τα δέντρα των στρατιωτών των στρατηγών. Κατά το γεγονός αυτό, θα διατρέχετε το δέντρο μάχης (combat_s) και θα διαγράφετε όλους τους στρατιώτες που έχουν πεθάνει από αυτό (όλες οι διαγραφές θα πρέπει να πραγματοποιούνται με μια διάσχιση). Για να πραγματοποιηθεί αυτό θα πρέπει να επιλέξετε την κατάλληλη διάσχιση που επιτρέπει να γίνουν σωστά οι διαγραφές σε ένα πέρασμα. Επιπρόσθετα, για κάθε στρατιώτη που είναι νεκρός (πεδίο alive=0), θα εντοπίζετε σε ποιο στρατηγό ανήκει (εξετάζοντας το πεδίο gid του c_soldier) και στη συνέχεια θα διαγράφετε τον στρατιώτη αυτόν από το δέντρο του στρατηγού του. Τέλος, ο στρατιώτης θα διαγράφεται και από τον πίνακα κατακερματισμού απογραφής.

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

```
U

    GENERALS LIST:
    <gid1>: <sid1,1> . . . <sid1,n1>
    <gid2>: <sid2,1> . . . <sid2,n2>
    ...
    <gidk>: <sidk,1> . . . <sidk,nk>

    SOLDIERS HASH TABLE:
    <sid1,1>, <sid1,2>, ..., <sid1,m1>,
    <sid2,1>, <sid2,2>, ..., <sid2,m2>,

    ...

    <sidw,1>, <sidw,2>, ..., <sidw,mw>
DONE
```

όπου k είναι ο αριθμός κόμβων στη λίστα στρατηγών, για κάθε i, $1 \le i \le k$, n_i είναι το το πλήθος των κόμβων του δέντρου στρατιωτών του i-οστού κόμβου της λίστας στρατηγών, και για κάθε j, $1 \le j \le n_i$, και $<sid_{i,j}>$ είναι το αναγνωριστικό του j-οστού κόμβου στο δέντρο στρατιωτών του i-οστού στρατηγού (βάσει της ενδοδιατεταγμένης διάσχισης) και όπου w είναι ο αριθμός των θέσεων του πίνακα κατακερματισμού στρατιωτών, για κάθε i, $1 \le i \le w$, και για κάθε j, $1 \le j \le m_i$, $sid_{i,j}$ είναι το αναγνωριστικό του j-οστού κόμβου της i-οστής αλυσίδας του πίνακα κατακερματισμού.

- W <gid>

Γεγονός τύπου print general's soldiers το οποίο σηματοδοτεί την εκτύπωση όλων των στρατιωτών ενός στρατηγού. Για το στρατηγό με αναγνωριστικό <gid> θα πρέπει να τυπώνεται το δέντρο στρατιωτών του. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
W <gid>
    Soldier tree = <sid1>, <sid2>, ..., <sidn>
DONE
```

όπου n είναι ο αριθμός των κόμβων του δέντρου στρατιωτών του στρατηγού με αναγνωριστικό <gid> και για κάθε $i \in \{1, \ldots, n\}$, $<sid_i>$ είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στην i-οστό κόμβο του δέντρου στρατιωτών του στρατηγού (βάσει ενδοδιατεταγμένης διάσχισης).

- X

Γεγονός τύπου print generals το οποίο σηματοδοτεί την εκτύπωση όλων των στρατηγών. Για κάθε στρατηγό θα πρέπει να εκτυπώνονται όλα τα στοιχεία του, συμπεριλαμβανομένων του δέντρου στρατιωτών του. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
X
    GENERALS:
    <gid1>: <sid1,1> . . . <sid1,n1>
    <gid2>: <sid2,1> . . . <sid2,n2>
    ...
    <gidk>: <sidk,1> . . . <sidk,nk>
DONE
```

όπου k είναι ο αριθμός κόμβων στη λίστα στρατηγών, για κάθε i, $1 \le i \le k$, n_i είναι το τον πλήθος των κόμβων του δέντρου στρατιωτών του i-οστού κόμβου της λίστας στρατηγών, και για κάθε j, $1 \le j \le n_i$, και $<sid_{i,j}>$ είναι το αναγνωριστικό του j-οστού κόμβου στο δέντρο στρατιωτών του i-οστού στρατηγού (βάσει ενδοδιατεταγμένης διάσχισης).

- Y

Γεγονός το οποίο σηματοδοτεί την εκτύπωση όλων των στρατιωτών του πίνακα κατακερματισμού στρατιωτών. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την

ακόλουθη πληροφορία:

```
Y
    Registration list = <sid1:gid1>, <sid2:gid2>, ..., <sidn:gidn>
DONE
```

όπου n είναι ο αριθμός των εγγραφών του πίνακα κατακερματισμού και για κάθε $i \in \{1, \ldots, n\}$, $<sid_i>$ είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στην i-οστή εγγραφή του πίνακα αυτού και $<gid_i>$ το αναγνωριστικό του στρατηγού στον οποίο υπακούει ο στρατιώτης που αντιστιχεί στην i-οστή εγγραφή του πίνακα.

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 soldier {
      int sid;
      int gid;
      struct soldier *next;
};
struct TREE_soldier {
      int sid;
      struct TPEE_soldier *rc;
      struct TPEE_soldier *lc;
      struct TPEE_soldier *p;
};

struct general {
      int gid;
      int combats_no;
      struct TREE_soldier *soldiers_R;
      struct TREE_soldier *soldiers_S;
      struct general *next;
};

struct c_soldier {
      int sid;
      int alive;
      int gid;
      int left_cnt;
      struct c_soldier *lc;
      struct c_soldier *rc;
};
```

```
struct combat {
      int soldier_cnt;
      struct c_soldier *combat_s;
};

/* global, the maximum number of soldiers */
extern unsigned int max_soldiers_g;

|/* global, the registration hashtable */
struct soldier *registration_hashtable;

/* global, the generals list*/
struct general *generals_list;

/* global, variable holding the combat info */
```