

ΗΥ240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2017-2018

Διδάσκουσα: Παναγιώτα Φατούρου

Προγραμματιστική Εργασία - 1^ο Μέρος

Ημερομηνία Παράδοσης: Δευτέρα, 20 Νοεμβρίου 2017, ώρα 23:59.

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



Γενική Περιγραφή

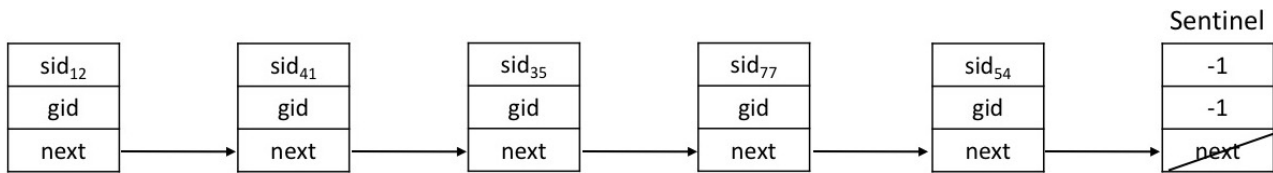
Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που θα προσομοιώνει τις τελευταίες ημέρες του τρωικού πολέμου, μεταξύ των Ελλήνων (Αχαιών κατά τον Όμηρο) και των Τρώων, όπως αυτός περιγράφεται στην Ιλιάδα. Ο πόλεμος αυτός είναι από τα κύρια γεγονότα της Ελληνικής Μυθολογίας και αποτέλεσε πηγή αστείρευτης έμπνευσης για την αρχαία ελληνική λογοτεχνία, συμπεριλαμβανομένων και των έργων του Ομήρου: της Ιλιάδας και της Οδύσσειας.

Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Ο κατάλογος των εμπόλεμων (Ραψωδία Β, 22^η 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 μια απλά συνδεδεμένη, μη ταξινομημένη λίστα, με κόμβο φρουρό. Η λίστα αυτή ονομάζεται **λίστα απογραφής** (registration list). Ο κάθε κόμβος της λίστας αυτής 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 λίστας απογραφής.

Ο κόμβος φρουρός είναι τύπου soldier με τιμή -1 στα πεδία sid και gid. Στο Σχήμα 1 παρουσιάζεται η λίστα απογραφής (registration list).



Σχήμα 1: Η απλά συνδεδεμένη λίστα απογραφής, η οποία είναι μη-ταξινομημένη, με κόμβο φρουρό.

Τα στρατεύματα των Αχαιών είναι οργανωμένα σε παρατάξεις των οποίων ηγούνται γενναίοι βασιλιάδες και στρατηγοί (όπως ο Αγαμέμνωνας, ο Μενέλαος, ο Νέστορας, ο Οδυσσεύς, ο Αχιλλεύς, ο Διομήδης κι άλλοι). 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, με κόμβο φρουρό*. 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_head:** Δείκτης τύπου soldier που δείχνει στον πρώτο κόμβο της λίστας ανδρών που υπακούουν στον στρατηγό. Η λίστα αυτή είναι *διπλά συνδεδεμένη, ταξινομημένη με βάση το αναγνωριστικό του στατιώτη* και ονομάζεται **λίστα στρατιωτών**. Κάθε στοιχείο της λίστας αυτής αποτελεί μια εγγραφή τύπου DDL_soldier με τα ακόλουθα πεδία:
 - **sid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον εμπόλεμο άνδρα.
 - **next:** Δείκτης (τύπου soldier) στον επόμενο κόμβο της λίστας απογραφής.
 - **prev:** Δείκτης (τύπου soldier) στον προηγούμενο κόμβο της λίστας απογραφής.
- **soldiers_tail:** Δείκτης τύπου DDL_soldier στον τελευταίο κόμβο της **λίστας στρατιωτών** που υπακούν στον στρατηγό.
- **next:** Δείκτης (τύπου general) στον επόμενο κόμβο της λίστας στρατηγών.

Ο κόμβος φρουρός είναι τύπου general με τιμή -1 στα πεδία gid και combats_no, ενώ τα πεδία soldiers_head και soldiers_tail έχουν τιμή NULL.

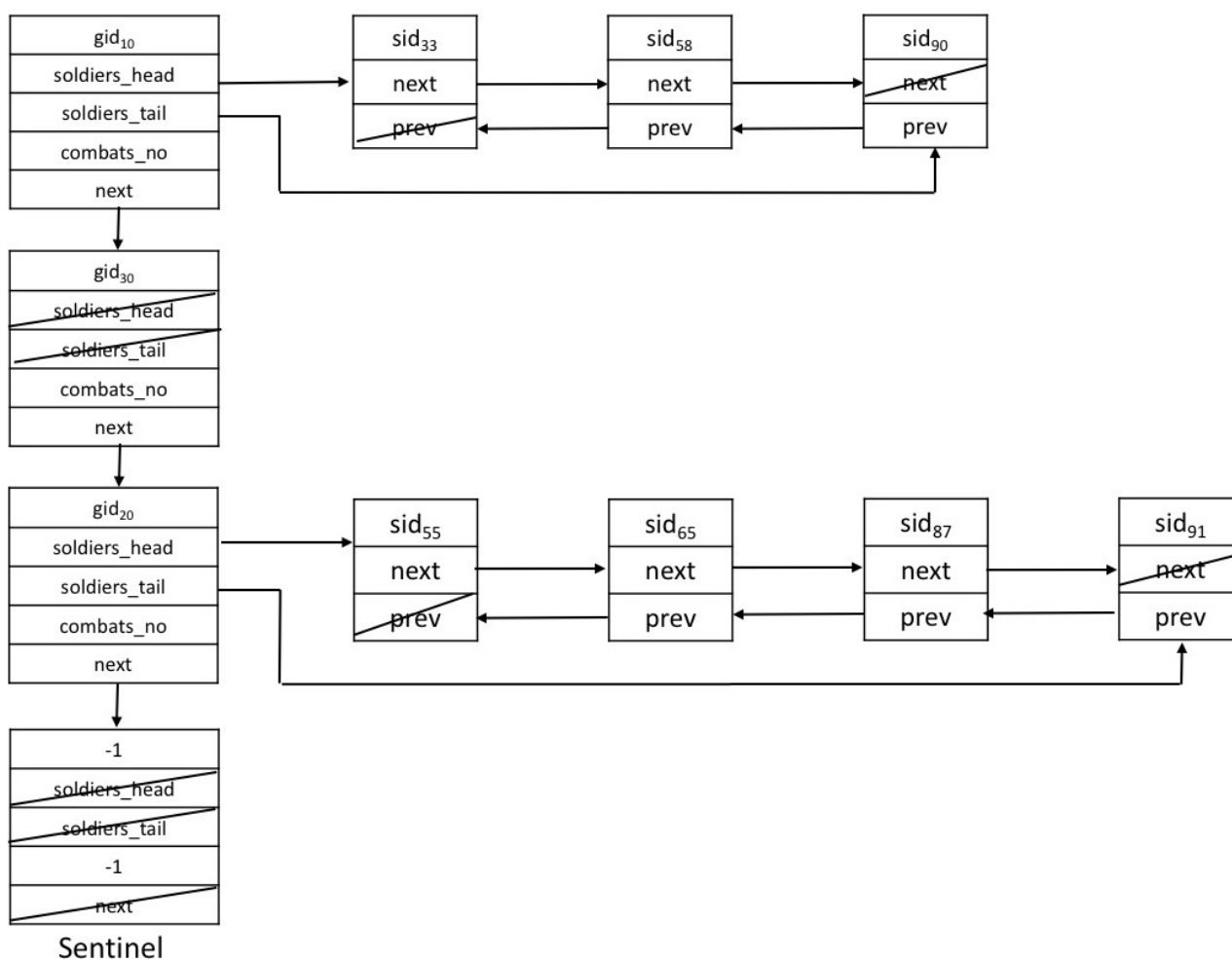
Μάχες στις Ραφωδίες Δ και Ε (22^η ημέρα), Θ (25^η ημέρα), Λ, Μ και Ν (26^η ημέρα, οι μάχες γύρω από το τείχος των Αχαιών), Π και Ρ (27^η ημέρα, ο Πάτροκλος οδηγεί τους Μυρμηδόνες στη μάχη), Φ (ο Αχιλλεύς επιστρέφει στη μάχη και αντιμετωπίζει εκτός από τους Τρώες και τον ποταμό Σκάμανδρο που ορμά εναντίον του).

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) στην αρχή μιας *απλά συνδεδεμένης, μη ταξινομημένης λίστας* που ονομάζεται **λίστα μάχης**. Το κάθε στοιχείο της λίστας αποτελεί μια εγγραφή τύπου c_soldier με τα ακόλουθα πεδία:

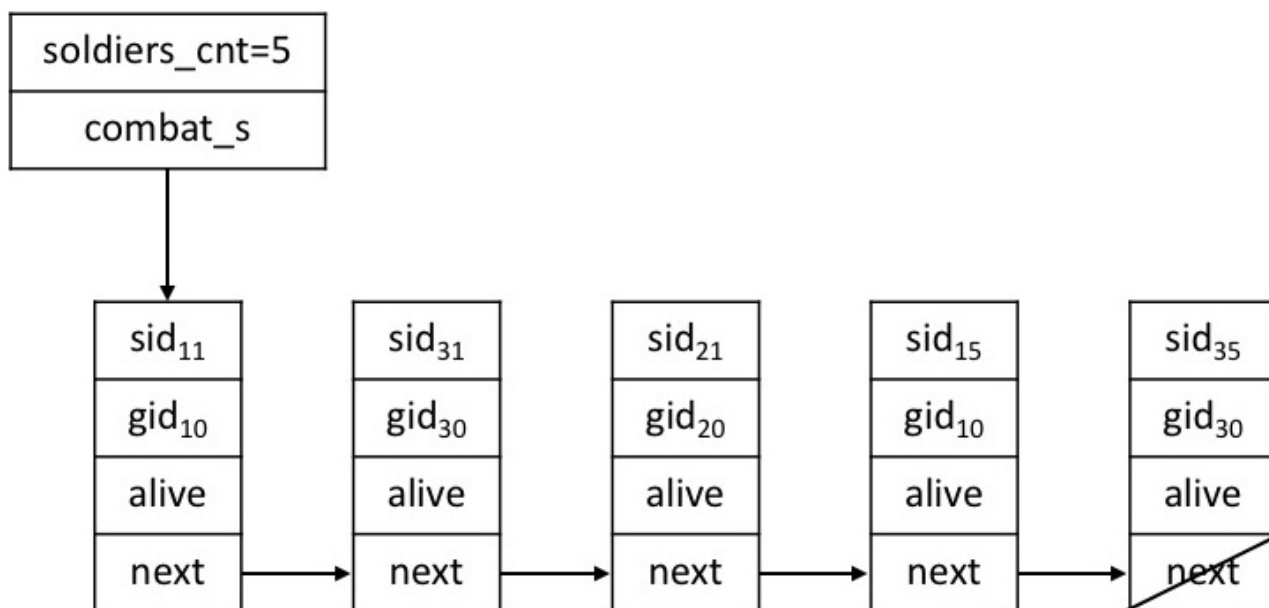
- **sid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον μαχόμενο άνδρα.
- **alive**: Δυαδική μεταβλητή που αντιπροσωπεύει την κατάσταση του μαχόμενου άνδρα. Η τιμή 1 σηματοδοτεί ότι ο άνδρας είναι ζωντανός, ενώ η τιμή 0 το αντίθετο.
- **gid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το βασιλιά/στρατηγό στον οποίο υπακούει ο μαχόμενος άνδρας.
- **next**: Δείκτης (τύπου c_soldier) στον επόμενο κόμβο της λίστας μάχης.

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



Σχήμα 2: Η μη ταξινομημένη, απλά συνδεδεμένη λίστα στρατηγών. Ο κάθε κόμβος (general) περιέχει έναν δείκτη στο πρώτο στοιχείο μιας διπλά συνδεδεμένης λίστας στατιωτών. Η λίστα στρατιωτών του κάθε στρατηγού είναι ταξινομημένη με βάση το αναγνωριστικό των στρατιωτών.

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



Σχήμα 3: Η δομή `combat` που περιέχει τη λίστα 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], 22^η ημέρα**). Κατά το γεγονός αυτό γίνεται εισαγωγή ενός νέου στρατιώτη στη λίστα απογραφής. Ο στρατιώτης θα έχει αναγνωριστικό <sid> και θα δρά υπό τις διαταγές του στρατηγού με αναγνωριστικό <gid>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

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

όπου n είναι ο αριθμός των κόμβων της λίστας απογραφής και για κάθε $i \in \{1, \dots, n\}$, <sid_i> είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής και <gid_i> το αναγνωριστικό του στρατηγού που υπακούει ο στρατιώτης του i-οστό κόμβου.

– **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. Η εισαγωγή πρέπει να γίνεται με τη μικρότερη χρονική πολυπλοκότητα. 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 $i \in \{1, \dots, n\}$, <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 τη λίστα απογραφής, να δημιουργηθούν οι λίστες ανδρών που υπάγονται σε κάθε στρατηγό. Αυτό μπορεί να γίνει εξετάζοντας το πεδίο gid του κάθε κόμβου της λίστας απογραφής και στη συνέχεια να γίνεται αναζήτηση του στρατηγού με αναγνωριστικό gid στη λίστα στρατηγών. Στη συνέχεια θα γίνεται εισαγωγή ενός struct τύπου DLL_soldier το οποίο θα αντιστοιχεί στον στρατιώτη στη λίστα στρατιωτών του στρατηγού. Για παράδειγμα αν το πεδίο gid ενός κόμβου της λίστας απογραφής έχει τιμή 32, τότε θα γίνει αναζήτηση του στρατηγού με αναγνωριστικό 32 στη λίστα στρατηγών. Στη συνέχεια θα δημιουργηθεί ένα νέο struct DDL_soldier για τον στρατιώτη στη λίστα στρατιωτών του στρατηγού που

εντοπίσαμε προηγουμένως. Επιπλέον μετά από κάθε εισαγωγή η λίστα στρατιωτών του κάθε στρατηγού πρέπει να είναι ταξινομημένη με βάση το αναγνωριστικό των στρατιωτών. Στο τέλος του γεγονότος, για όλους τους στρατιώτες της λίστας απογραφής πρέπει να υπάρχουν αντίστοιχες εγγραφές στις κατάλληλες λίστες 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 μέγεθος της λίστας στρατιωτών του i -οστού κόμβου της λίστας στρατηγών, και για κάθε j , $1 \leq j \leq n_i$, και $\langle \text{sid}_{i,j} \rangle$ είναι το αναγνωριστικό του j -οστού κόμβου στη λίστα στρατιωτών του i -οστού στρατηγού.

– M $\langle \text{gid}_1 \rangle \langle \text{gid}_2 \rangle$

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 $\langle \text{gid}_1 \rangle$ και ο Πάτροκλος έχει αναγνωριστικό $\langle \text{gid}_2 \rangle$. Κατά το γεγονός αυτό πρέπει να διαγράψετε τον κόμβο με αναγνωριστικό $\langle \text{gid}_1 \rangle$ από τη λίστα στρατηγών. Επιπλέον οι άνδρες του Αχιλλέα (οι Μυρμηδόνες) θα μεταφερθούν στη λίστα στρατιωτών του Πάτροκλου (ο οποίος έχει αναγνωριστικό $\langle \text{gid}_2 \rangle$)¹. **Η συνένωση των δύο λιστών πρέπει να γίνει με πολυπλοκότητα $O(n_1+n_2)$, όπου n_1 και n_2 το μέγεθος της λίστας στρατιωτών του Αχιλλέα και του Πάτροκλου, αντίστοιχα.** Με άλλα λόγια πρέπει να διασχίσετε την καθεμιά από τις δύο λίστες στρατιωτών μόνο μία φορά. Μετά το τέλος της διαδικασίας, η λίστα στρατιωτών του Πάτροκλου πρέπει να είναι ταξινομημένη.

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

¹ Στην Ιλιάδα, ο Πάτροκλος οδηγεί απλά 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.

M

GENERALS:

<gid₁>: <sid_{1,1}> . . . <sid_{1,n1}><gid₂>: <sid_{2,1}> . . . <sid_{2,n2}>

...

<gid_k>: <sid_{k,1}> . . . <sid_{k,nk}>

DONE

όπου για κάθε i , $1 \leq i \leq k$, n_i είναι το μέγεθος της λίστας στρατιωτών του i -οστού κόμβου της λίστας στρατηγών, και για κάθε j , $1 \leq j \leq n_i$, και $\langle \text{sid}_{i,j} \rangle$ είναι το αναγνωριστικό του j -οστού κόμβου στη λίστα στρατιωτών του i -οστού στρατηγού.

– P <gid₁> <gid₂> <gid₃>

Γεγονός τύπου *prepare for battle* το οποίο σηματοδοτεί την επιλογή των ανδρών που θα συμμετέχουν στην επόμενη μάχη. Στη μάχη θα συμμετέχουν οι στρατηγοί με αναγνωριστικά <gid₁> <gid₂> και <gid₃> με τους άνδρες τους. Για κάθε έναν από τους στρατηγούς <gid₁>, <gid₂>, και <gid₃>, θα πρέπει να αυξήσετε τον μετρητή *combats_no*, που υποδηλώνει τον αριθμό των μαχών στις οποίες ο στρατηγός αυτός έχει συμμετάσχει.

Οι στρατιώτες που θα πολεμήσουν, θα πρέπει να αποθηκεύονται στη λίστα μάχης (*combat_s*) της εγγραφής *combat*. Κάθε στοιχείο της λίστας αυτής περιέχει μια εγγραφή τύπου *c_soldier*. Κάθε φορά που εισάγεται ένας κόμβος σ' αυτή τη λίστα, θα πρέπει να ενημερώνετε κατάλληλα τον μετρητή *soldiers_cnt* της εγγραφής *combat*. Ο τρόπος με τον οποίο θα γίνεται η τοποθέτηση στρατιωτών στη λίστα είναι επιλέγοντας στρατιώτες από τις λίστες των τριών στρατηγών, εναλλάξ, όπως περιγράφεται στη συνέχεια:

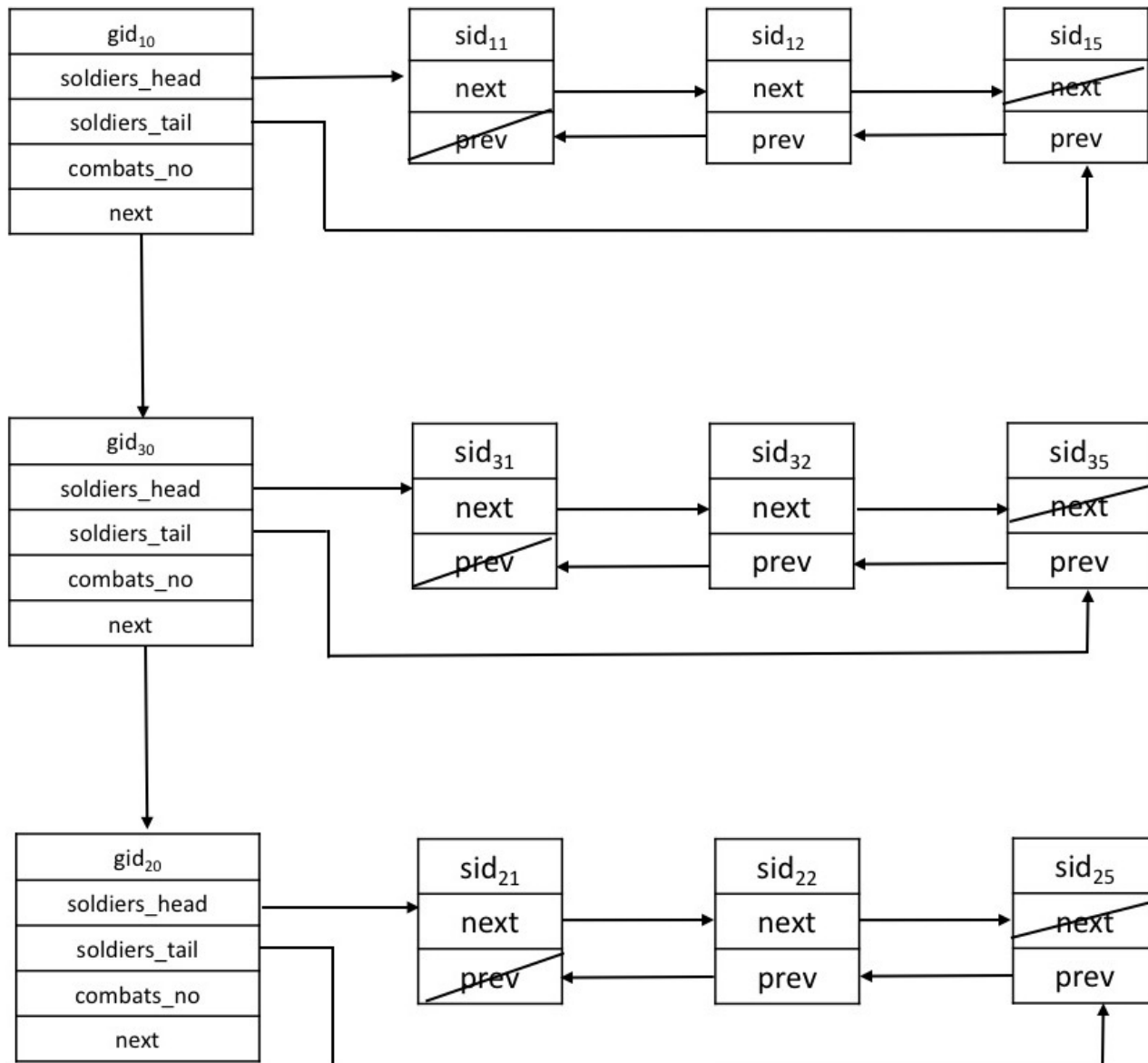
Βήμα 1: Αρχικά θα εισάγεται στη λίστα μάχης τον πρώτο στρατιώτη του πρώτου στρατηγού, στη συνέχεια τον πρώτο στρατιώτη του δεύτερου στρατηγού και μετά τον πρώτο στρατιώτη του τρίτου στρατηγού.

Βήμα 2: Στη συνέχεια θα επιλέξετε στρατιώτες από το τέλος της λίστας στρατιωτών του κάθε στρατηγού. Συνεπώς στην τέταρτη θέση της λίστας μάχης θα εισαχθεί ο τελευταίος στρατιώτης του πρώτου στρατηγού, έπειτα ο τελευταίος στρατιώτης του δεύτερου στρατηγού και στη συνέχεια ο τελευταίος στρατιώτης του τρίτου στρατηγού.

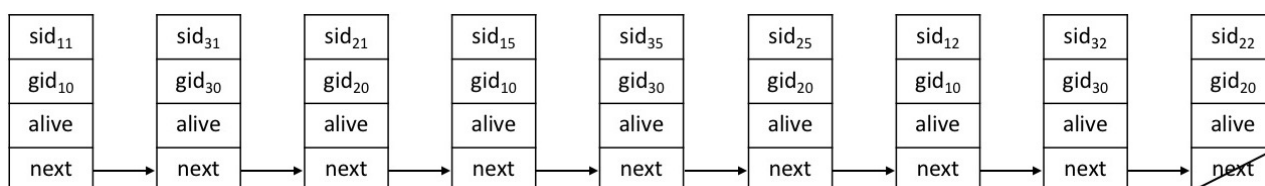
Ολοκληρώνοντας την επιλογή στρατιωτών από το τέλος της λίστας στρατιωτών του κάθε στρατηγού, θα συνεχίσετε με την επιλογή στρατιωτών από την αρχή της κάθε λίστας όπως περιγράφεται στο βήμα 1, συνεχίζοντας και πάλι με το βήμα 2, κ.ο.κ. Η διαδικασία αυτή θα συνεχίζεται μέχρι να έχετε εισάγει όλους τους στρατιώτες της λίστας του κάθε στρατηγού <gid₁>, <gid₂>, και <gid₃> στη λίστα μάχης. Με άλλα λόγια θα πρέπει να διατρέξετε την κάθε λίστα στρατιωτών των στρατηγών <gid₁>, <gid₂>, και <gid₃> μόνο μία φορά. Προσέξτε ότι από κάποια στοιχεία θα περάσετε καθώς θα κινείστε από την αρχή προς το τέλος της κάθε λίστας, ενώ από κάποια άλλα καθώς διασχίζετε τις λίστες από το τέλος προς την αρχή. Αυτό είναι εφικτό δεδομένου ότι οι λίστες είναι διπλά συνδεδεμένες. Κατά την εκτέλεση του γεγονότος αυτού, οι διπλά-συνδεδεμένες λίστες στρατιωτών των στρατηγών <gid1>, <gid2> και <gid3> μένουν αναλοίωτες.

Για κάθε στρατιώτη που εισάγετε στη λίστα μάχης της εγγραφής *combat*, θα πρέπει να θέτετε το πεδίο *alive* στην τιμή 1, αφού ο στρατιώτης είναι ζωντανός πριν τη μάχη.

Στην εικόνα 4 παρουσιάζεται ένα παράδειγμα με τις λίστες στρατιωτών των στρατηγών με αναγνωριστικά 10, 30 και 20 που θα συμμετέχουν στη μάχη. Η εικόνα 5 παρουσιάζει το αποτέλεσμα της συνένωσης των στρατιωτών των παραπάνω στρατηγών στη λίστα μάχης σύμφωνα με τον αλγόριθμο που περιγράφεται παραπάνω.



Σχήμα 4 Η λίστα στρατηγών που περιέχει του στρατηγούς με αναγνωριστικά 10, 30 και 20 όπως επίσης και οι αντίστοιχες λίστες στρατιωτών που περιέχουν.



Σχήμα 5 Η λίστα μάχης όπως προκύπτει απ' τη συνένωση των λιστών των στρατιωτών που παρουσιάζονται στην εικόνα 4.

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

```
P <gid1> <gid2> <gid3>
    Combat soldiers: <sid1>, <sid2> . . . <sidn>
DONE
```

όπου n είναι το μέγεθος της λίστας μάχης και για κάθε i , $1 \leq i \leq n$, $\langle \text{sid}_i \rangle$ είναι το αναγνωριστικό του i -οστού στρατιώτη στη λίστα μάχης.

– B <god_favor>

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 $\langle \text{god_favor} \rangle$.

Στην περίπτωση που οι Θεοί δεν ευνοούν τα στρατεύματα των Ελλήνων (π.χ. στις μάχες της 25^{ης} και 26^{ης} ημέρας που συμπεριλαμβάνουν και τις μάχες γύρω από τα τείχη των Αχαιώνόπως περογράφονται στις 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 διατρέχετε τη λίστα μάχης (`combat_soldiers`) της εγγραφής `combat` και θα ορίσετε το 40% των πρώτων στρατιωτών της λίστας ως νεκρούς (δηλαδή θα θέσετε απλά το πεδίο `alive` του αντίστοιχου `combat_soldier` στην τιμή 0).

Στην περίπτωση που οι Θεοί ευνοούν τα στρατεύματα των Ελλήνων (όπως π.χ. στις μάχες που περιγράφονται στις Ραψωδίες Δ, Ε, Υ και Φ), τότε οι απώλειες είναι λιγότερες και αντιστοιχούν στο 10% των στρατιωτών που συμμετέχουν στη μάχη. Σε αυτήν την περίπτωση ο τρόπος που θα επιλέξετε ποιοι στρατιώτες θα χάσουν τη ζωή τους είναι ο εξής:

Οι στρατιώτες που συμμετέχουν στη μάχη είναι παραταγμένοι σε σειρές. Η κάθε σειρά αποτελείται από 10 στρατιώτες. Οι απώλειες ορίζονται στο 10% της κάθε σειράς. Με άλλα λόγια θα διατρέξετε τη λίστα μάχης (`combat_soldiers`) της δομής `combat` και για κάθε 10άδα στρατιωτών (σ.σ. μια σειρά) θα μαρκάρετε ως νεκρό τον πρώτο στρατιώτη της σειράς.

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

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

όπου n είναι το μέγεθος της λίστας μάχης και για κάθε i , $1 \leq i \leq n$, $\langle \text{sid}_i \rangle$ και $\langle \text{alive}_i \rangle$ είναι το αναγνωριστικό και η κατάσταση του στρατιώτη που αντιστοιχεί στον i -οστό κόμβο στη λίστα μάχης.

– U

Ανακωχή και ταφή νεκρών (23^η ημέρα, Ραψωδία H381-432). Γεγονός κατά το οποίο πραγματοποιείται ταφή των νεκρών. Στο γεγονός αυτό θα πρέπει να διαγράφονται οι στρατιώτες που έχασαν τη ζωή τους στη μάχη από τις λίστες των στρατιωτών των στρατηγών. Κατά το γεγονός αυτό θα διατρέχετε τη λίστα μάχης (combat_s) της δομής combat και για κάθε στρατιώτη που είναι νεκρός (δηλαδή έχει στο πεδίο alive την τιμή 0), θα εντοπίζετε σε ποιο στρατηγό ανήκει (εξετάζοντας το πεδίο gid του c_soldier) και στη συνέχεια θα τον διαγράφετε από τη λίστα του στρατηγού του. Κατά τη διαδικασία αυτή θα πρέπει να γίνει μόνο μια διάσχιση της λίστας μάχης και της λίστας στρατιωτών του κάθε στρατηγού. Μετά το τέλος της διαδικασίας η λίστα μάχης της εγγραφής combat πρέπει να μείνει άδεια και ο μετρητής soldiers_cnt να μηδενιστεί.

Επιπρόσθετα, θα πρέπει να διατρέχετε τη λίστα απογραφής και να διαγράφετε την εγγραφή που αντιστοιχεί στον άνδρα που πέθανε και από εκεί.

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

```

U

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 μέγεθος της λίστας στρατιωτών του i -οστού κόμβου της λίστας στρατηγών, και για κάθε j , $1 \leq j \leq n_i$, και $\langle \text{sid}_{i,j} \rangle$ είναι το αναγνωριστικό του j -οστού κόμβου στη λίστα στρατιωτών του i -οστού στρατηγού.

– T

Ο δούρειος ίππος. Το γεγονός αυτό, που είναι τύπου trojan horse, προσομοιώνει την ιστορία του Δουρείου Ίππου. Στο Δούρειο Ίππο εισχωρούν οι πέντε στρατηγοί με τις περισσότερες μάχες. Για το σκοπό αυτό θα χρησιμοποιήσετε έναν βοηθητικό πίνακα 5 θέσεων στον οποίο θα αποθηκεύονται δείκτες προς τους κόμβους της λίστας στρατηγών.

Πιο συγκεκριμένα θα εφαρμόσετε τις εξής ενέργειες:

- i. Ο βοηθητικός πίνακας αρχικοποιείται με τους 5 πρώτους στρατηγούς της λίστας στρατηγών.
- ii. Καθώς εκτελείται η διάσχιση της λίστας στρατηγών, για κάθε κόμβο που επισκέπτεστε, εξετάζετε αν ο μετρητής combats για τον στρατηγό αυτό έχει τιμή μεγαλύτερη από τον στρατηγό με τις λιγότερες μάχες στον βοηθητικό πίνακα. Αν συμβαίνει αυτό, τότε αντικαθιστούμε τον στρατηγό με τις λιγότερες μάχες στον πίνακα με τον στρατηγό που αντιστοιχεί στον τρέχοντα κόμβο της λίστας. Στη συνέχεια, θα πρέπει να υπολογίζετε εκ νέου ποιος είναι ο στρατηγός του πίνακα με τις λιγότερες μάχες και να επαναλαμβάνετε το βήμα.
- iii. Μετά το τέλος της διάσχισης της λίστας στρατηγών, ο βοηθητικός πίνακας θα περιέχει δείκτες στους κόμβους της λίστας που αντιστοιχούν στους 5 στρατηγούς με τις περισσότερες μάχες.

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

```
T
    General = <gid1:combats1>, <gid2:combats2>, ..., <gid5:combats5>,
DONE
```

και για κάθε $i \in \{1, \dots, 5\}$, $\langle \text{gid}_i \rangle$ και $\langle \text{combats}_i \rangle$ είναι το αναγνωριστικό και ο αριθμός των μαχών αντίστοιχα του στρατηγού που αντιστοιχεί στην i -οστή θέση του βοηθητικού πίνακα.

– 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 της λίστας 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 για κάθε i , $1 \leq i \leq k$, n_i είναι το μέγεθος της λίστας στρατιωτών του i -οστού κόμβου της λίστας στρατηγών, και για κάθε j , $1 \leq j \leq n_i$, και $\langle \text{sid}_{i,j} \rangle$ είναι το αναγνωριστικό του j -οστού κόμβου στη λίστα στρατιωτών του i -οστού στρατηγού.

– Y

Γεγονός το οποίο σηματοδοτεί την εκτύπωση όλων των στρατιωτών της λίστας απογραφών. 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 κόμβων της λίστας απογραφής και για κάθε $i \in \{1, \dots, n\}$, $\langle \text{sid}_i \rangle$ είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον i -οστό κόμβο της λίστας αυτής και $\langle \text{gid}_i \rangle$ το αναγνωριστικό του στρατηγού στον οποίο υπακούει ο στρατιώτης που αντιστοιχεί στον i -οστό κόμβο της λίστας.

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 DDL_soldier {  
    int sid;  
    struct soldier *next;  
    struct soldier *prev;  
};
```

```
struct general {  
    int gid;  
    int combats_no;  
    struct DDL_soldier *soldiers_head;  
    struct DDL_soldier *soldiers_tail;  
    struct general *next;  
};
```

```
struct c_soldier {  
    int sid;  
    int alive;  
    int gid;  
    struct c_soldier *next;  
};
```

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

```
/* global variable pointing to the beginning of the registration list */
```

```
struct soldier * registration_list;

/* global variable for the sentinel of registration list */
struct soldier *registration_sentinel;

/* global variable, pointer to the beginning of the generals list*/
struct general *generals_list;

/* global variable for the sentinel of generals list */
struct general *generals_sentinel;

/* global variable holding the combat info */
struct combat my_combat;
```