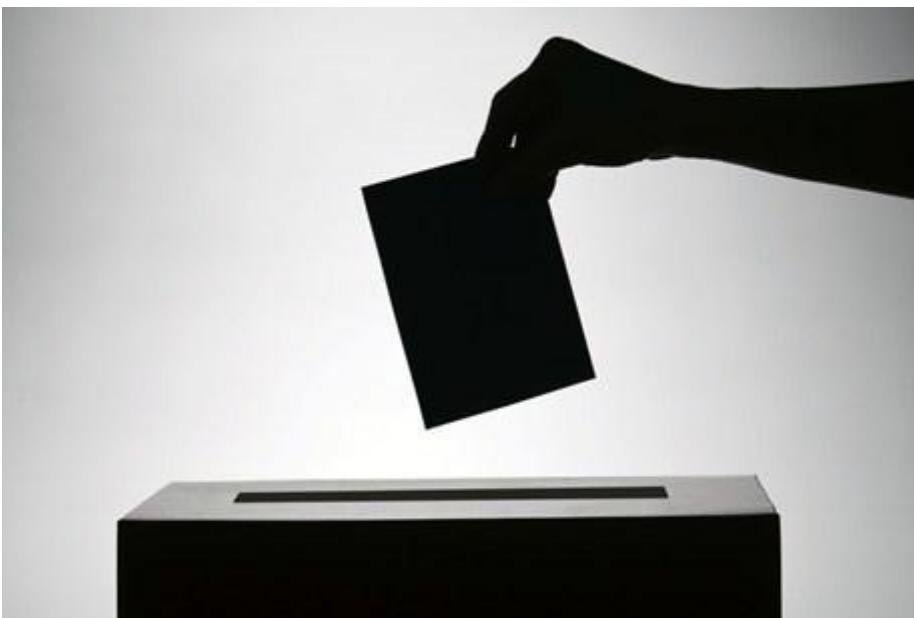


ΗΥ240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2024-25
Διδάσκουσα: Παναγιώτα Φατούρου

Προγραμματιστική Εργασία - 1^ο Μέρος

Ημερομηνία Παράδοσης: Δευτέρα, 18 Νοεμβρίου 2024, ώρα 23:59.

Τρόπος Παράδοσης: Μέσω της σελίδας elearn του μαθήματος. Πληροφορίες σχετικά με το elearn παρέχονται στην ιστοσελίδα του μαθήματος.



Γενική Περιγραφή

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που προσομοιώνει το εκλογικό σύστημα της Ελλάδας. Το **ελληνικό εκλογικό σύστημα** είναι το σύστημα με το οποίο κατανέμονται οι βουλευτικές έδρες στα κόμματα με βάση τις ψήφους που έλαβαν στις βουλευτικές εκλογές. Ρυθμίζεται από ειδικό **νόμο**, που ονομάζεται εκλογικός νόμος.

Σημείωση: Προκειμένου να πραγματωθούν οι εκπαιδευτικοί στόχοι της εργασίας, δεν ακολουθείται πιστά ο εκλογικός νόμος και έχουν γίνει σημαντικές απλοποιήσεις σχετικά με τον τρόπο με τον οποίο κατανέμονται οι έδρες. Συνεπώς, όσα αναγράφονται στην παρούσα εκφώνηση δεν αντιστοιχούν με ακρίβεια στην ισχύουσα εκλογική νομοθεσία.

Οι υποψήφιοι βουλευτές χωρίζονται σε εκλογικές περιφέρειες, και αποθηκεύονται στη διπλά συνδεδεμένη ταξινομημένη λίστα **candidates** της δομής **district**. Είναι εγγραφές τύπου **candidate**, αποτελούμενες από τα ακόλουθα πεδία:

1. **cid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει τον υποψήφιο. Το cid μπορεί να έχει οποιαδήποτε τιμή εκτός από τις δεσμευμένες τιμές **0** και **1**. Αυτές τις τιμές τις δεσμεύουμε για τις λευκές και άκυρες ψήφους αντίστοιχα, όπως θα δούμε παρακάτω.
2. **pid**: Αναγνωριστικό (τύπου int) που αντιστοιχεί στο κόμμα το οποίο ο υποψήφιος εκπροσωπεί.
3. **votes**: Ο αριθμός (τύπου int) των ψήφων που συγκέντρωσε ο υποψήφιος απ' όλα τα εκλογικά τμήματα της εκλογικής περιφέρειας με αναγνωριστικό did.
4. **elected**: Μεταβλητή (τύπου int) που δηλώνει αν ο υποψήφιος εκλέγεται. Η μεταβλητή αυτή αποθηκεύει μόνο τις τιμές **0** (FALSE) και **1** (TRUE).
5. **prev**: Δείκτης (τύπου candidate) στον προηγούμενο κόμβο της λίστας υποψηφίων.
6. **next**: Δείκτης (τύπου candidate) στον επόμενο κόμβο της λίστας υποψηφίων.

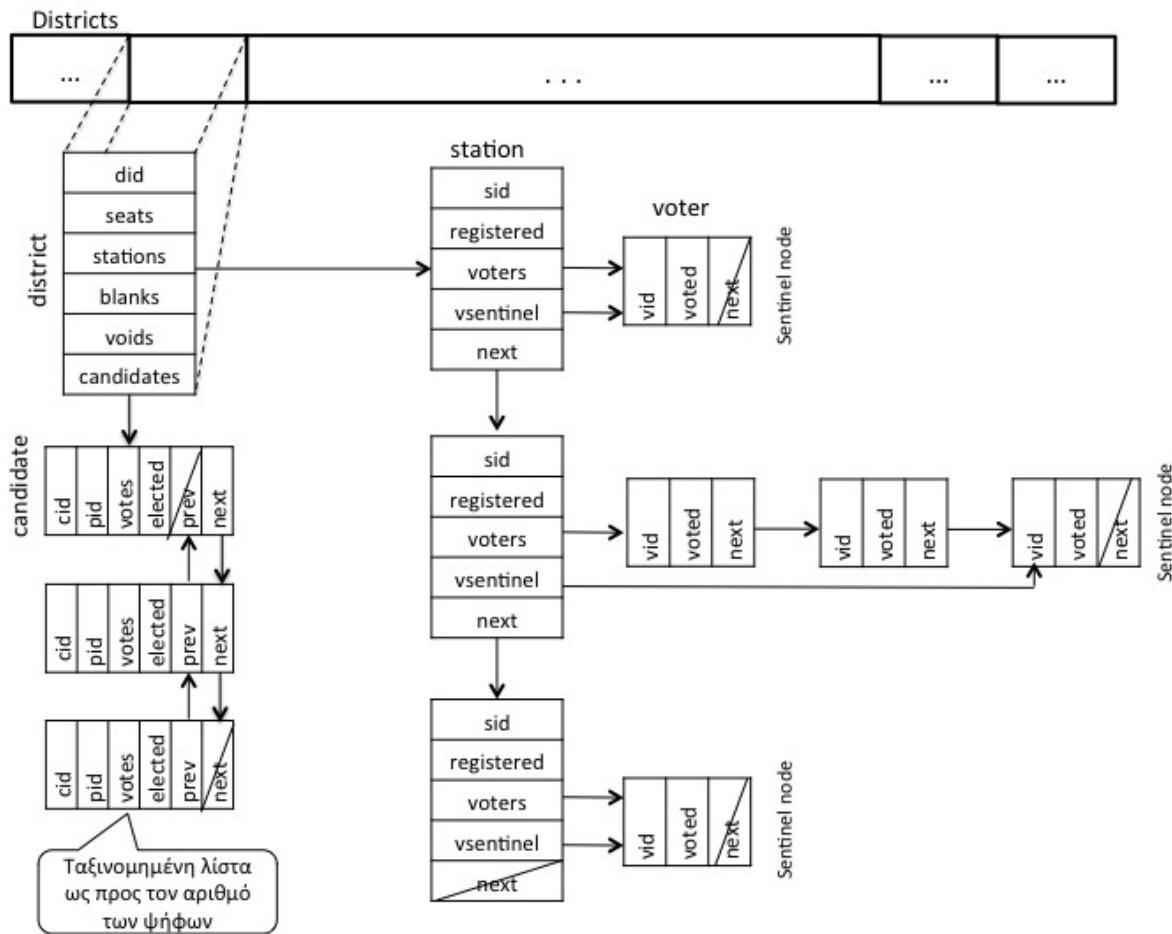
Για τις ανάγκες της εργασίας αυτής, θεωρούμε ότι υπάρχουν πέντε διαφορετικά κόμματα που συμμετέχουν στις εκλογές. Θα πρέπει να χρησιμοποιήσετε έναν πίνακα σταθερού μεγέθους **5 θέσεων**, ο οποίος ονομάζεται **πίνακας κομμάτων** (Parties). Το κάθε στοιχείο του **πίνακα κομμάτων** είναι μια εγγραφή τύπου party με τα ακόλουθα πεδία:

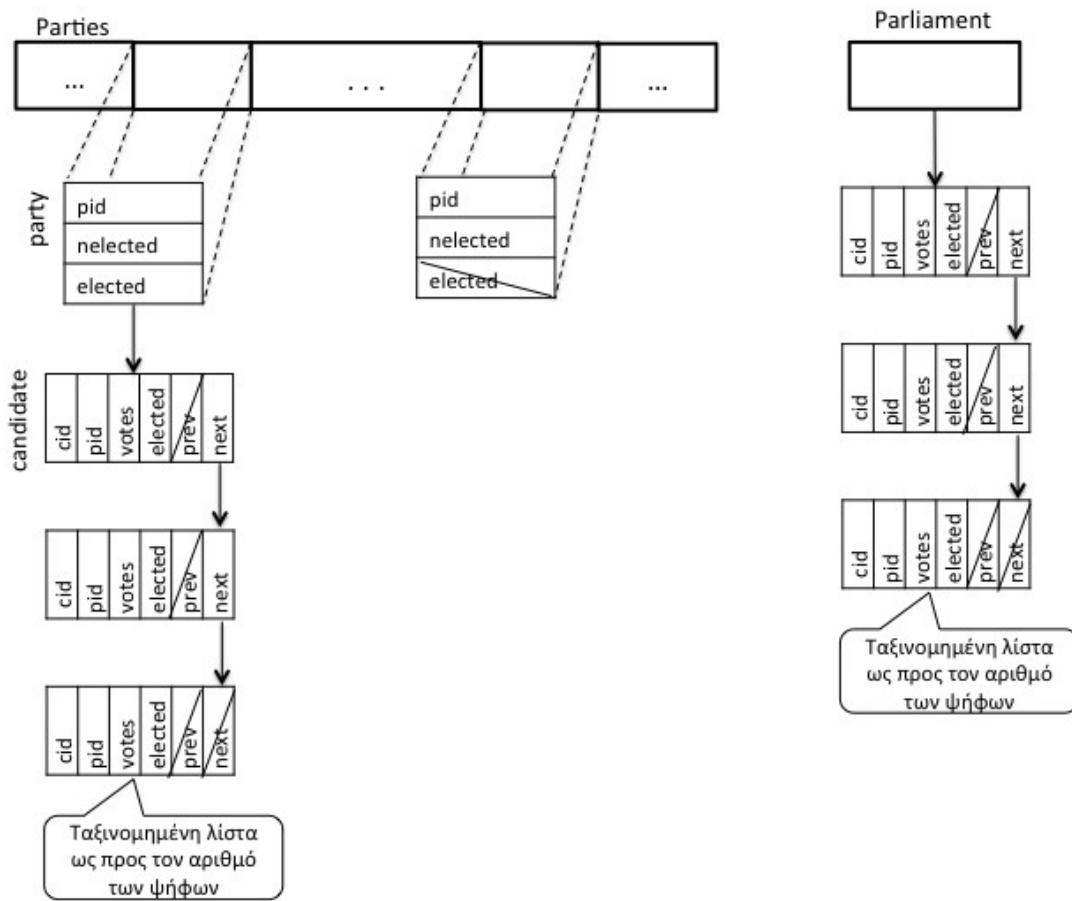
1. **pid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει το κόμμα.
2. **selected**: Ο αριθμός (τύπου int) των βουλευτών που εκλέγει το κόμμα με αναγνωριστικό j.
3. **elected**: Δείκτης (τύπου candidate) στο πρώτο στοιχείο μιας **ταξινομημένης**, απλά συνδεδεμένης λίστας που περιέχει εγγραφές τύπου candidate. Η λίστα αυτή ονομάζεται **λίστα εκλεγμένων μελών** του κόμματος και είναι **ταξινομημένη σε φθίνουσα διάταξη ως προς το πεδίο votes** κάθε στοιχείου της. Το κάθε στοιχείο της λίστας αντιστοιχεί σε έναν υποψήφιο που εκπροσωπεί το κόμμα και έχει εκλεγεί σε κάποια εκλογική περιφέρεια. Τα πεδία της εγγραφής candidate περιγράφονται παραπάνω. Εφόσον η λίστα βουλευτών του κόμματος ζητείται να είναι απλά συνδεδεμένη, η τιμή του πεδίου prev κάθε τέτοιας εγγραφής θα πρέπει να έχει την τιμή null.

Τέλος δημιουργείται η δομή του **κοινοβουλίου** (Parliament). Ουσιαστικά αποτελείται από έναν δείκτη (τύπου candidate) στο πρώτο στοιχείο μιας **ταξινομημένης**, απλά συνδεδεμένης λίστας. Η λίστα αυτή ονομάζεται **λίστα βουλευτών** και είναι ταξινομημένη σε φθίνουσα διάταξη ως προς τον αριθμό των ψήφων που έλαβε ο κάθε βουλευτής (δηλαδή ως προς το πεδίο votes κάθε στοιχείου της).

Κάθε στοιχείο του πίνακα Districts αρχικοποιείται έτσι ώστε το πεδίο did του να είναι ίσο με την τιμή **-1**. Κάθε φορά που εισάγετε ένα νέο στοιχείο στον πίνακα, θα πρέπει να το τοποθετείτε στο πρώτο στοιχείο του πίνακα του οποίου το πεδίο did είναι ίσο με την τιμή **-1**. Τα έγκυρα στοιχεία του πίνακα, δηλαδή εκείνα με αναγνωριστικό διαφορετικό από **-1** θα πρέπει να βρίσκονται σε συνεχόμενες θέσεις του πίνακα, ξεκινώντας από τη θέση **0**.

Παρακάτω παρουσιάζονται οι δομές δεδομένων σε σχηματική μορφή.





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:

– **A**

Γεγονός τύπου *announce elections* το οποίο υποδηλώνει την αναγγελία των εκλογών. Κατά το γεγονός αυτό αρχικοποιείται η δομή του **πίνακα εκλογικών περιφερειών** (Districts). Συγκεκριμένα το κάθε πεδίο του κάθε στοιχείου του πίνακα αρχικοποιείται με την τιμή `NULL` αν είναι δείκτης και την τιμή `-1` αν είναι τύπου `int`. Με τον ίδιο τρόπο αρχικοποιούνται ο **πίνακας κομμάτων** (Parties) και η δομή του **κοινοβουλίου** (Parliament). Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
A  
DONE
```

– **D <did> <seats>**

Γεγονός τύπου *create district* το οποίο σηματοδοτεί τη δημιουργία μιας νέας εκλογικής περιφέρειας (district) με αναγνωριστικό `<did>` και αριθμό εκλεγόμενων εδρών `<seats>`. Ο αριθμός των κατανεμημένων εδρών allotted αντίστοιχα αρχικοποιείται σε `0`. Η νέα εκλογική περιφέρεια εισάγεται στον **πίνακα εκλογικών περιφερειών**. Η νέα εκλογική περιφέρεια περιέχει κενή **λίστα εκλογικών τμημάτων** (stations) και κενή **λίστα υποψηφίων** (candidates). **Χρησιμοποιήστε κατάλληλες μεταβλητές ώστε η χρονική πολυπλοκότητα της εισαγωγής να είναι O(1)**. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
D <did> <seats>  
Districts = <did1>, <did2>, ..., <didn>  
DONE
```

όπου `n` είναι ο αριθμός των εκλογικών περιφερειών που έχουν δημιουργηθεί μέχι τώρα και για κάθε `i ∈ {1, ..., n}`, `<did>` είναι το αναγνωριστικό της `i`-οστής εκλογικής περιφέρειας.

– **S <sid> <did>**

Γεγονός τύπου *create station* το οποίο σηματοδοτεί τη δημιουργία ενός νέου εκλογικού τμήματος με αναγνωριστικό `<sid>` στην εκλογική περιφέρεια με αναγνωριστικό `<did>`. Στην εγγραφή τύπου *station* που αντιστοιχεί στο νέο εκλογικό τμήμα, ο αριθμός των εγγεγραμμένων ψηφοφόρων (πεδίο registered), αρχικοποιείται με την τιμή `0`, ενώ η **λίστα ψηφοφόρων (voters)** είναι κενή (δηλαδή η λίστα ψηφοφόρων περιέχει μόνο τον κόμβο φρουρό). Το γεγονός αυτό εισάγει το νέο εκλογικό τμήμα στη **λίστα εκλογικών τμημάτων** (stations) της εκλογικής περιφέρειας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
S <sid> <did>  
Stations = <sid1>, <sid2>, ..., <sidn>  
DONE
```

όπου n είναι ο αριθμός των εκλογικών τμημάτων που υπάρχουν στην εκλογική περιφέρεια με αναγνωριστικό $\langle did \rangle$ και για κάθε $i \in \{1, \dots, n\}$, $\langle sid_i \rangle$ είναι το αναγνωριστικό του i -οστού εκλογικού τμήματος στην εκλογική περιφέρεια.

– **P $\langle pid \rangle$**

Γεγονός τύπου *create party* το οποίο σηματοδοτεί την ίδρυση ενός νέου κόμματος (party) με αναγνωριστικό $\langle pid \rangle$. Το νέο κόμμα εισάγεται στον **πίνακα κομμάτων**. Το νέο κόμμα περιέχει κενή **λίστα εκλεγμένων μελών** (elected). Μετά της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
P <pid>
  Parties = <pid1>, <pid2>, ..., <pidn>
DONE
```

όπου n είναι το ο αριθμός των ιδρυθέντων κομμάτων και για κάθε $i \in \{1, \dots, n\}$, $\langle pid_i \rangle$ είναι το αναγνωριστικό του i -οστού ιδρυθέντος κόμματος.

– **C $\langle cid \rangle \langle did \rangle \langle pid \rangle$**

Γεγονός τύπου *register candidate* το οποίο σηματοδοτεί την εγγραφή ενός νέου υποψηφίου (candidate) με αναγνωριστικό $\langle cid \rangle$ του κόμματος με αναγνωριστικό $\langle pid \rangle$ στην εκλογική περιφέρεια με αναγνωριστικό $\langle did \rangle$. Το γεγονός αυτό προσθέτει το νέο υποψήφιο στη **λίστα υποψηφίων** της συγκεκριμένης εκλογικής περιφέρειας και το πεδίο votes αρχικοποιείται με την τιμή 0. Η πληροφορία ότι ο υποψήφιος ανήκει στο συγκεκριμένο κόμμα, δεν αποθηκεύεται στην παρούσα φάση στον πίνακα Parties (αποθηκεύεται απλά στο πεδίο pid της δομής του υποψηφίου στη λίστα υποψηφίων της περιφέρειας με αναγνωριστικό $\langle did \rangle$). Μετά της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
C <cid> <did> <pid>
  Candidates = <cid1>, <cid2>, ..., <cidn>
DONE
```

όπου n είναι ο αριθμός των υποψηφίων της εκλογικής περιφέρειας και για κάθε $i \in \{1, \dots, n\}$, $\langle cid_i \rangle$ το αναγνωριστικό του i -οστού υποψηφίου στη λίστα των υποψηφίων της εκλογικής περιφέρειας.

– **R $\langle vid \rangle \langle did \rangle \langle sid \rangle$**

Γεγονός τύπου *register voter* το οποίο σηματοδοτεί την εγγραφή ενός νέου ψηφοφόρου (voter) με αναγνωριστικό $\langle vid \rangle$ στο εκλογικό τμήμα με αναγνωριστικό $\langle sid \rangle$ της εκλογικής περιφέρειας με αναγνωριστικό $\langle did \rangle$. Το γεγονός αυτό προσθέτει το νέο ψηφοφόρο στη **λίστα ψηφοφόρων** του εκλογικού τμήματος. Σε αυτό το σημείο αυξάνεται ο αριθμός εγγεγραμμένων ψηφοφόρων του εκλογικού τμήματος κατά ένα. Μετά της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
R <vid> <did> <sid>  
Voters = <vid1>, <vid2>, ..., <vidn>  
DONE
```

όπου n είναι ο αριθμός των ψηφοφόρων του εκλογικού τμήματος με αναγνωριστικό $<sid>$ και για κάθε $i \in \{1, \dots, n\}$, $<vid_i>$ είναι το αναγνωριστικό του i -οστού ψηφοφόρου στη λίστα των ψηφοφόρων του εκλογικού τμήματος με αναγνωριστικό $<sid>$.

– **U <vid>**

Γεγονός τύπου *unregister voter* το οποίο σηματοδοτεί τη διαγραφή του ψηφοφόρου (voter) με αναγνωριστικό $<vid>$. Το γεγονός αυτό εντοπίζει τον ψηφοφόρο ψάχνοντας όλα τα εκλογικά τμήματα όλων των εκλογικών περιφερειών της επικράτειας και τον αφαιρεί από τη **λίστα ψηφοφόρων** του εκλογικού του τμήματος. Σε αυτό το σημείο μειώνεται ο αριθμός εγγεγραμμένων ψηφοφόρων του συγκεκριμένου εκλογικού τμήματος κατά ένα. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
U <vid>  
<did> <sid>  
Voters = <vid1>, <vid2>, ..., <vidn>  
DONE
```

όπου $<sid>$ είναι το αναγνωριστικό του εκλογικού τμήματος στο οποίο ανήκει ο ψηφοφόρος, $<did>$ είναι το αναγνωριστικό της εκλογικής περιφέρειας στην οποία ανήκει το εκλογικό τμήμα με αναγνωριστικό $<sid>$, n είναι ο αριθμός των ψηφοφόρων του εκλογικού τμήματος με αναγνωριστικό $<sid>$ και για κάθε $i \in \{1, \dots, n\}$, $<vid_i>$ είναι το αναγνωριστικό του i -οστού ψηφοφόρου στη λίστα των ψηφοφόρων του συγκεκριμένου εκλογικού τμήματος.

– **E**

Γεγονός τύπου *delete empty stations* το οποίο σηματοδοτεί την κατάργηση όλων των εκλογικών τμημάτων στα οποία δεν υπάρχουν εγγεγραμμένοι ψηφοφόροι. Τα εκλογικά τμήματα που καταργούνται αφαιρούνται από τη λίστα εκλογικών τμημάτων της εκλογικής περιφέρειας στην οποία ανήκουν. **Το γεγονός αυτό πρέπει να εκτελείται σε χρόνο O(n)**, όπου n είναι ο συνολικός αριθμός εκλογικών τμημάτων σε όλες τις εκλογικές περιφέρειες (δηλαδή σε ολόκληρη την επικράτεια). Κατά την εκτέλεση ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
E  
<sid1> <did1>  
<sid2> <did2>  
...  
<sidn> <didn>  
DONE
```

όπου n είναι ο αριθμός των εκλογικών τμημάτων που καταργήθηκαν και για κάθε $i \in [1 .. n]$, $<sid_i>$ είναι το αναγνωριστικό του i -οστού εκλογικού τμήματος που καταργήθηκε και $<did>$ είναι η εκλογική

περιφέρεια στην οποία ανήκε το εκλογικό τμήμα με αναγνωριστικό <sid>.

– **V <vid> <sid> <cid>**

Γεγονός τύπου *vote* το οποίο σηματοδοτεί τη διαδικασία ψήφου. Ο ψηφοφόρος με αναγνωριστικό <vid> που ανήκει στο εκλογικό τμήμα με αναγνωριστικό <sid> ψηφίζει τον υποψήφιο με αναγνωριστικό <cid> (ή λευκό, ή άκυρο). Το γεγονός αυτό εντοπίζει πρώτα το εκλογικό τμήμα ψάχνοντας όλες τις εκλογικές περιφέρειες. Στη συνέχεια εντοπίζει τον ψηφοφόρο στη **λίστα ψηφοφόρων** του εκλογικού του τμήματος και σημειώνει ότι συμμετείχε στις εκλογές (voted). Σε αυτό το σημείο προσμετράται και η ψήφος. Αν το <cid> είναι ίσο με το δεσμευμένο αναγνωριστικό 0, τότε αυξάνεται ο αριθμός των λευκών κατά ένα. Ομοίως, αν το <cid> είναι ίσο με το δεσμευμένο αναγνωριστικό 1, τότε αυξάνεται ο αριθμός των άκυρων κατά ένα. Σε κάθε άλλη περίπτωση εντοπίζεται ο υποψήφιος με αναγνωριστικό <cid> στη λίστα υποψηφίων της συγκεκριμένης εκλογικής περιφέρειας και αυξάνεται ο αριθμός των ψήφων του. Επίσης, αν ο υποψήφιος ξεπερνά τώρα σε αριθμό ψήφων που έχει λάβει τον προηγούμενό του στη λίστα πραγματοποιείται μια ανταλλαγή θέσεων μεταξύ τους για να διατηρηθεί η λίστα υποψηφίων ταξινομημένη σε φθίνουσα διάταξη. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
V <vid> <sid> <cid>
  District = <did>
  Candidate votes = (<cid1>, <votes1>), (<cid2>, <votes2>), ..., (<cidn>, <votesn>)
  Blanks = <blanks>
  Voids = <voids>
  DONE
```

όπου <did> είναι το αναγνωριστικό της εκλογικής περιφέρειας στην οποία ανήκει το εκλογικό τμήμα με αναγνωριστικό <sid>, η είναι ο αριθμός των υποψηφίων της συγκεκριμένης εκλογικής περιφέρειας και για κάθε $i \in \{1, \dots, n\}$, <cid_i> και <votes_i> είναι το αναγνωριστικό και ο αριθμός ψήφων του i -οστού υποψηφίου στη λίστα των υποψηφίων της εκλογικής περιφέρειας, αντίστοιχα. Επίσης, <blanks> είναι ο αριθμός των λευκών και <voids> ο αριθμός των άκυρων ψήφων στη συγκεκριμένη εκλογική περιφέρεια.

– **M <did>**

Γεγονός τύπου *count votes* το οποίο σηματοδοτεί την καταμέτρηση ψήφων στην εκλογική περιφέρεια με αναγνωριστικό <did>. Το γεγονός αυτό τρέχει τον ακόλουθο αλγόριθμο για τη συγκεκριμένη εκλογική περιφέρεια:

1. Διασχίζεται η λίστα υποψηφίων και για κάθε κόμμα, προσμετρούνται οι ψήφοι που έλαβε το κόμμα συνολικά στην περιφέρεια αυτή και το αποτέλεσμα αποθηκεύεται στο κατάλληλο στοιχείο ενός βοηθητικού πίνακα.

2. Υπολογίζεται το εκλογικό μέτρο το οποίο ορίζεται ως ο λόγος του συνόλου των έγκυρων ψήφων στην εκλογική περιφέρεια (χωρίς λευκά και άκυρα) προς τον αριθμό των εδρών (seats) που αναλογούν στην περιφέρεια αυτή.
3. Υπολογίζεται το πόσες έδρες βγάζει το κάθε κόμμα στην περιφέρεια διαιρώντας το σύνολο έγκυρων ψήφων που έλαβε το κόμμα με το εκλογικό μέτρο, με στρογγυλοποίηση προς τα κάτω. Το νούμερο αυτό αποθηκεύεται επίσης στον βιοηθητικό πίνακα.
4. Για κάθε κόμμα και για κάθε έδρα που δικαιούται το κόμμα εντοπίζεται ο πρώτος μη εκλεγμένος υποψήφιος του κόμματος αυτού στη **λίστα υποψηφίων της εκλογικής περιφέρειας** και εκλέγεται, δηλαδή τίθεται το πεδίο elected του υποψηφίου σε TRUE, οι κατανεμημένες έδρες της περιφέρειας (πεδίο allotted) αυξάνονται κατά 1, και γίνονται οι κατάλληλες εισαγωγές στη **λίστα εκλεγμένων μελών** του κόμματος αυτού στον **πίνακα κομμάτων**. Προσοχή: η **λίστα εκλεγμένων μελών** του κόμματος είναι ταξινομημένη ως προς τον αριθμό ψήφων. Επίσης, ενημερώνεται το πεδίο nelected το οποίο δηλώνει τον αριθμό των εκλεγμένων μελών του κόμματος.

Για την εκτέλεση αυτού του γεγονότος, ο αλγόριθμος δεν επιτρέπεται να διασχίσει τη λίστα υποψηφίων της περιφέρειας, περισσότερες από δύο φορές, μια για την καταμέτρηση των ψήφων κάθε κόμματος και μια για την εκλογή των κατάλληλων μελών του κόμματος.

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

```
M <did>
Seats =
<cid1> <pid1> <votes1>
<cid2> <pid2> <votes2>
...
<cidn> <pidn> <votesn>
DONE
```

όπου n είναι ο αριθμός των εδρών στην εκλογική περιφέρεια και για κάθε i ∈ {1, ..., n}, <cid_i>, <pid_i>, <votes_i> είναι το αναγνωριστικό, το κόμμα, και ο αριθμός ψήφων του i-οστού υποψηφίου που εκλέγεται στην εκλογική περιφέρεια, αντίστοιχα.

- G

Γεγονός τύπου *form government* το οποίο σηματοδοτεί την απόδοση των λοιπών αδιάθετων εδρών των εκλογικών περιφερειών. Σε κάθε εκλογική περιφέρεια αυτές θα ισούνται με seats (συνολικές έδρες) - allotted (κατανεμημένες έδρες). Βρίσκουμε το πρώτο κόμμα, το οποίο χάριν απλοποίησης της εργασίας, είναι το κόμμα που έχει συγκεντρώσει τις περισσότερες έδρες σε ολόκληρη την επικράτεια. Διανύουμε τον **πίνακα εκλογικών περιφερειών** και σε κάθε περιφέρεια, όσες έδρες περισσεύουν προσμετρούνται στο συγκεκριμένο κόμμα. Έστω ότι αυτές είναι x για το κόμμα y. Τότε, οι x μη εκλεγμένοι υποψήφιοι

του κόμματος υ στη συγκεκριμένη περιφέρεια, οι οποίοι έλαβαν το μεγαλύτερο αριθμό ψήφων, εκλέγονται, δηλαδή τίθεται το elected πεδίο της εγγραφής που τους αντιστοιχεί σε TRUE και ενημερώνεται κατάλληλα η λίστα εκλεγμένων μελών του κόμματος υ στον πίνακα κομμάτων. Εάν το πρώτο κόμμα δεν έχει αρκετούς μη εκλεγμένους υποψηφίους στην εκλογική περιφέρεια ώστε να παραλάβει όλες τις αδιάθετες έδρες, τότε συνεχίζουμε εκλέγοντας τους υποψηφίους της περιφέρειας κατά φθίνουσα σειρά βάσει ψήφων, ανεξαρτήτως κόμματος, μέχρις ότου να κατανεμηθούν όλες οι διαθέσιμες έδρες της περιφέρειας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
G
Seats =
<did1> <cid1> <votes1>
<did2> <cid2> <votes2>
...
<didn> <cidn> <votesn>
DONE
```

όπου n είναι ο συνολικός αριθμός αδιάθετων εδρών σε όλες τις εκλογικές περιφέρειες και για κάθε i ∈ {1, ..., n}, <did_i>, <cid_i>, <votes_i> είναι το αναγνωριστικό της εκλογικής περιφέρειας, το αναγνωριστικό του i-οστού υποψηφίου που εκλέγεται στη συγκεκριμένη εκλογική περιφέρεια και ο αριθμός ψήφων που έλαβε αντίστοιχα.

- **N**

Γεγονός τύπου *form parliament* το οποίο σηματοδοτεί την εκλογή των μελών της Βουλής. Το γεγονός αυτό συνενώνει τη λίστα εκλεγμένων μελών του πίνακα κομμάτων (Parties) στη λίστα βουλευτών (members), η οποία είναι ταξινομημένη ως προς τον αριθμό των ψήφων που έλαβε ο κάθε βουλευτής. Το γεγονός αυτό πρέπει να εκτελείται σε χρόνο O(n), όπου n είναι ο συνολικός αριθμός των βουλευτών. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
N
Members =
<cid1> <pid1> <votes1>
<cid2> <pid2> <votes2>
...
<cidn> <pidn> <votesn>
DONE
```

όπου n είναι ο αριθμός των εκλεγμένων βουλευτών και για κάθε i ∈ {1, ..., n}, <cid_i>, <pid_i>, <votes_i> είναι το αναγνωριστικό, το κόμμα, και ο αριθμός ψήφων του i-οστού βουλευτή αντίστοιχα.

- **I <did>**

Γεγονός τύπου *print district* το οποίο σηματοδοτεί την εκτύπωση στοιχείων από την εκλογική περιφέρεια με αναγνωριστικό <did>. Το γεγονός απαιτεί την αναζήτηση της εκλογικής περιφέρειας στον πίνακα εκλογικών περιφερειών. Κατά την εκτέλεση ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
I <did>
  Seats = <seats>
  Blanks = <blanks>
  Voids = <voids>
  Candidates =
    <cid1> <pid1> <votes1>
    <cid2> <pid2> <votes2>
    ...
    <cidn> <pidn> <votesn>
  Stations = <sid1>, <sid2>, ..., <sidm>
DONE
```

όπου n είναι ο αριθμός των υποψηφίων της εκλογικής περιφέρειας και για κάθε i ∈ {1, ..., n}, <cid_i>, <pid_i>, <votes_i> είναι το αναγνωριστικό, το κόμμα, και ο αριθμός ψήφων του i-οστού υποψηφίου αντίστοιχα. Επιπρόσθετα, m είναι ο αριθμός των εκλογικών τμημάτων που υπάρχουν στην εκλογική περιφέρεια και για κάθε j ∈ {1, ..., m}, <sid_j> είναι το αναγνωριστικό του i-οστού εκλογικού τμήματος στην εκλογική περιφέρεια. Επίσης, <seats> είναι ο αριθμός των εδρών που εκλέγει αυτή η εκλογική περιφέρεια, <blanks> είναι ο αριθμός των λευκών και <voids> είναι ο αριθμός των άκυρων ψήφων στην εκλογική περιφέρεια.

- **J <sid> <did>**

Γεγονός τύπου *print station* το οποίο σηματοδοτεί την εκτύπωση στοιχείων από το εκλογικό τμήμα με αναγνωριστικό <sid> το οποίο βρίσκεται στην εκλογική περιφέρεια με αναγνωριστικό <did>. Κατά την εκτέλεση ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
J <sid>
  Registered = <registered>
  Voters =
    <vid1> <voted1>
    <vid2> <voted2>
    ...
    <vidn> <votedn>
DONE
```

όπου n είναι ο αριθμός των εγγεγραμμένων ψηφοφόρων στο εκλογικό τμήμα με αναγνωριστικό <sid> και για κάθε i $\in \{1, \dots, n\}$, <vid_i>, <voted_i> είναι το αναγνωριστικό του i-οστού ψηφοφόρου και το αν ο ψηφοφόρος άσκησε το εκλογικό του δικαίωμα αντίστοιχα. Επίσης, <registered> είναι ο αριθμός των εγγεγραμμένων ψηφοφόρων στο εκλογικό τμήμα με αναγνωριστικό <sid>.

- **K <pid>**

Γεγονός τύπου *print party* το οποίο σηματοδοτεί την εκτύπωση στοιχείων από το κόμμα με αναγνωριστικό <pid>. Κατά την εκτέλεση ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
K <pid>
  Elected =
    <cid1> <votes1>
    <cid2> <votes2>
    ...
    <cidn> <votesn>
DONE
```

όπου n είναι ο αριθμός των εκλεγμένων μελών του κόμματος και για κάθε i $\in \{1, \dots, n\}$, <cid_i>, <votes_i> το αναγνωριστικό του i-οστού εκλεγμένου μέλους και ο αριθμός των ψήφων που έλαβε, αντίστοιχα.

- **L**

Γεγονός τύπου *print parliament* το οποίο σηματοδοτεί την εκτύπωση των εκλεγμένων βουλευτών επικράτειας. Κατά την εκτέλεση ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L
  Members =
    <cid1> <pid1> <votes1>
    <cid2> <pid2> <votes2>
    ...
    <cidn> <pidn> <votesn>
DONE
```

όπου n είναι ο αριθμός των εκλεγμένων βουλευτών και για κάθε i $\in \{1, \dots, n\}$, <cid_i>, <pid_i>, <votes_i> είναι το αναγνωριστικό, το κόμμα, και ο αριθμός των ψήφων που έλαβε ο i-οστός βουλευτής, αντίστοιχα.

Δομές Δεδομένων

Στη συνέχεια παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας.

```
struct candidate {
    int cid; /* 0 reserved for blanks, 1 for voids */
    int pid;
    int votes;
    int elected;
    struct candidate *next;
    struct candidate *prev;
};

struct party {
    int pid;
    int nelected;
    struct candidate *elected; /* Singly-linked, sorted by votes */
};

struct parliament {
    struct candidate *members; /* Singly-linked, sorted by votes */
};

struct voter {
    int vid;
    int voted;
    struct voter *next;
};

struct station {
    int sid;
    int registered; /* Increment on new voter */
    struct voter *voters; /* Singly-linked, unsorted, sentinel */
    struct voter *vsentinel;
    struct station *next;
};

struct district {
    int did;
    int seats;
    int allotted;
    int blanks;
    int voids;
    struct station *stations; /* Singly-linked, unsorted */
    struct candidate *candidates; /* Doubly-linked, sorted by votes */
};

struct district Districts[56];
struct party Parties[5];
struct parliament Parliament;
```