

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Χειμερινό εξάμηνο ακ. έτους 2013-2014

Α' ομάδα ασκήσεων

1. Η παρούσα άσκηση σκοπό έχει να σας εξασκήσει στις συναρτήσεις κατασκευής-καταστροφής στην περίπτωση που έχουμε σύνθεση κλάσεων και στην προσβασιμότητα και εμβέλεια μεταξύ των μελών των κλάσεων.

Υλοποιήστε σε C++ μια προσομοίωση βιβλιοθήκης. Η βιβλιοθήκη αποτελείται από μια βάση, ένα ντουλάπι και τρία ράφια (με αυτή τη σειρά). Το ντουλάπι εσωτερικά έχει δυό ράφια. (Σημ: όπου “ράφι” εννοούμε χώρο που τοποθετούμε βιβλία).

Στη βιβλιοθήκη, ανά πάσα στιγμή βρίσκονται N βιβλία ($N \geq 0$). Το κάθε ράφι (είτε εξωτερικό είτε εσωτερικό) φιλοξενεί κάποια από αυτά. Υπάρχει μια μέγιστη χωρητικότητα ραφιών, L , ως προς τα βιβλία που το κάθε ράφι μπορεί να φιλοξενήσει.

Κατά την κατασκευή μιας βιβλιοθήκης, εκτυπώνεται ένα μήνυμα ότι μια βιβλιοθήκη κατασκευάζεται. Κατά την κατασκευή ενός ραφιού, εκτυπώνεται ένα μήνυμα ότι ένα ράφι κατασκευάζεται. Κατά την κατασκευή ενός ντουλαπιού, εκτυπώνεται ένα μήνυμα ότι ένα ντουλάπι κατασκευάζεται. Κατά την κατασκευή μιας βάσης, εκτυπώνεται ένα μήνυμα ότι μια βάση κατασκευάζεται. Κατά την κατασκευή τους, τόσο η βιβλιοθήκη, όσο και τα ράφια δεν περιέχουν κανένα βιβλίο.

Κατά την καταστροφή της βιβλιοθήκης, εκτυπώνεται σχετικό μήνυμα. Σχετικά μηνύματα εκτυπώνονται και στις περιπτώσεις καταστροφής ραφιών, ντουλαπιών και βάσης βιβλιοθήκης.

Στη βιβλιοθήκη τοποθετούμε ένα βιβλίο (`place_book`) προσδιορίζοντας με ένα αριθμό το χώρο στον οποίο θα τοποθετηθεί και τοποθετώντας το σε αυτό. Δηλαδή, αντιστοιχώντας

- 1: στο πάνω ράφι της βιβλιοθήκης
- 2: στο μεσαίο ράφι της βιβλιοθήκης
- 3: στο κάτω ράφι της βιβλιοθήκης
- 4: στο πάνω ράφι του ντουλαπιού της βιβλιοθήκης
- 5: στο κάτω ράφι του ντουλαπιού της βιβλιοθήκης

Σε αυτή την περίπτωση, αυξάνεται κατά 1 ο αριθμός των βιβλίων που είναι τοποθετημένα στη βιβλιοθήκη.

Μπορούμε επίσης να πάρουμε ένα βιβλίο από τη βιβλιοθήκη (`take_book`) προσδιορίζοντας το χώρο από το οποίο το αφαιρούμε και αφαιρώντας το από το χώρο αυτό. Ο προσδιορισμός του χώρου γίνεται με τρόπο ανάλογο με την τοποθέτηση (Σημείωση: δεν αναφερόμαστε σε συγκεκριμένο βιβλίο). Σε αυτή την περίπτωση, ο αριθμός των βιβλίων που βρίσκονται στη βιβλιοθήκη μειώνεται κατά 1.

Η τοποθέτηση ενός βιβλίου στο ντουλάπι (`place_book`), πραγματοποιείται προσδιορίζοντας το ράφι στο οποίο θα γίνει η τοποθέτηση και τοποθετώντας το στο ράφι αυτό. Για τον προσδιορισμό του ραφιού, με 1 προσδιορίζουμε το πάνω ράφι και με 2 το κάτω ράφι του ντουλαπιού της βιβλιοθήκης. Η αφαίρεση ενός βιβλίου από το ντουλάπι (`take_book`), πραγματοποιείται προσδιορίζοντας το ράφι από οποίο θα γίνει η αφαίρεση και αφαιρώντας το από το ράφι αυτό.

Η τοποθέτηση ενός βιβλίου σε ένα ράφι (`place_book`), πραγματοποιείται αυξάνοντας τον αριθμό των βιβλίων που είναι τοποθετημένα σε αυτό κατά 1. Αντίστοιχα, η αφαίρεση ενός βιβλίου από ένα ράφι (`take_book`), πραγματοποιείται μειώνοντας τον αριθμό των βιβλίων που είναι τοποθετημένα σε αυτό κατά 1 και αν το ράφι είναι άδειο, επιστρέφει false αλλιώς επιστρέφει true.

Στην περίπτωση που η ενδεχόμενη τοποθέτηση βιβλίου σε κάποιο ράφι ξεπερνά τη χωρητικότητα ραφίου, L , η ενέργεια να μην ολοκληρώνεται και να εκτυπώνεται μήνυμα λάθους. Στις περιπτώσεις αφαιρέσεων βιβλίων από κενούς χώρους, να εκτυπώνεται μήνυμα λάθους.

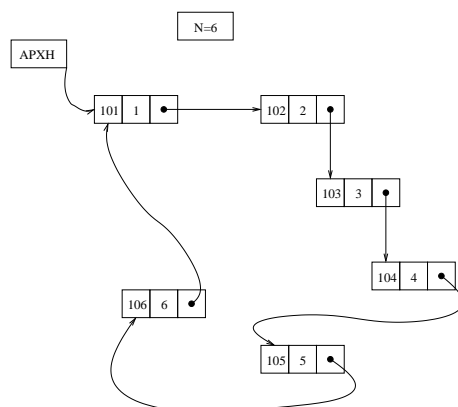
Υλοποιήστε τα παραπάνω, μέσω των κατάλληλων κλάσεων, ορίζοντας τα μέλη-δεδομένα που χρειάζονται και τις συναρτήσεις-μέλη που υλοποιούν την παραπάνω συμπεριφορά.

Για να αναδείξετε την προσομοίωσή σας, να υλοποιήσετε μια συνάρτηση `main` η οποία να δημιουργεί μια βιβλιοθήκη και κατόπιν να πραγματοποιεί K τυχαίες τοποθετήσεις βιβλίων και αφαιρέσεις.

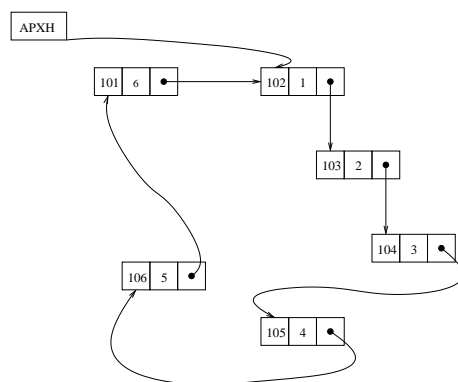
Τα K και L είναι παράμετροι της προσομοίωσης και να δίδονται από τη γραμμή εντολής με τη σειρά αυτή.

2. Να υλοποιηθεί σε C++ μια προσομοίωση (μιας εκδοχής) του παιχνιδιού με τις “μουσικές καρέκλες”. Στο παιχνίδι αυτό, έχουμε N συμμετέχοντες και $N - 1$ καρέκλες. Οι συμμετέχοντες σχηματίζουν αλυσίδα και γυρνούν γύρω από τις καρέκλες, ενόσω χτυπά ένα τύμπανο (ένα βήμα για κάθε κρούση).

Για το κάθε χτύπημα, θεωρούμε ότι αυτό που αλλάζει είναι η θέση του καθενός μέσα στην αλυσίδα (ο πρώτος γίνεται N -οστός, ο δεύτερος πρώτος, ο i -οστός πάει στη θέση $i - 1$ και ο N -οστός πάει στη θέση $N - 1$). Δηλαδή, αρχικά:



μετά από ένα χτύπημα:



Το τύμπανο χτυπά για τυχαίο αριθμό φορές. Μόλις το τύμπανο σταματήσει, $N - 1$ άτομα καταλαμβάνουν τις καρέκλες κι ο j -οστός (τυχαίος) που περισσεύει διαγράφεται από την αλυσίδα. Η αλυσίδα ανασυντάσσεται με τους υπόλοιπους και αφαιρούμε μια καρέκλα. Για λόγους ευκολίας, θεωρούμε ότι διατηρείται η σχετική διάταξη μεταξύ των συμμετεχόντων.

Η διαδικασία τύμπανο-διαγραφή-ανασύνταξη επαναλαμβάνεται έως ότου μείνει ένας μόνο στην αλυσίδα ο οποίος είναι και ο νικητής.

Κάθε συμμετέχων στην αλυσίδα έχει μια (αριθμητική) ταυτότητα και καταχωρεί και τη θέση του στη διάταξη της αλυσίδας. Επίσης, συνδέεται (“δείχνει”) στον επόμενό του. Ο τελευταίος (N -οστός) συνδέεται (“δείχνει”) με τον πρώτο συμμετέχοντα.

Η αλυσίδα να αναπαρασταθεί σαν κλάση με την εξής συμπεριφορά:

- Αρχικοποιείται με βάση το μήκος της N (ακέραιος). Κατά τη διαδικασία αρχικοποίησης δημιουργούνται N συμμετέχοντες και συνδέονται μεταξύ τους. Οι ταυτότητές τους είναι τυχαίοι αριθμοί.
- Ανταποκρίνεται σε ένα χτύπημα του τυμπάνου (**hit**), αλλάζοντας κατά ένα τη θέση κάθε συμμετέχοντα (όπως περιγράφηκε παραπάνω) και, κατά συνέπεια, την αρχή της αλυσίδας.
- Ανταποκρίνεται στο σταμάτημα του τυμπάνου (**stop_tune**). Στην περίπτωση αυτή, γίνεται διαγραφή ενός (τυχαίου) συμμετέχοντα, “διορθώνοντας” τη σύνδεση των υπολοίπων και ενημερώνοντας το μήκος αλυσίδας με το πλήθος των συμμετεχόντων που απέμειναν. Επίσης, γίνεται ενημέρωση της αρχής της αλυσίδας, αν είναι απαραίτητο. Τέλος, εκτυπώνεται η ταυτότητα αυτού που διαγράφηκε.
- Η αλυσίδα “ανασυντάσσεται” (**resume**), διορθώνοντας τις θέσεις των συμμετεχόντων ώστε να είναι πάλι διαδοχικές.
- Εκτυπώνονται οι ταυτότητες των συμμετεχόντων της αλυσίδας με τη σειρά που αυτοί εμφανίζονται στην αλυσίδα (**print_chain**).
- Γίνεται έλεγχος αν το παιχνίδι τελείωσε, δηλαδή αν έχει απομείνει μόνο ένας συμμετέχων στην αλυσίδα (**game_over**).
- Εμφανίζει το νικητή, αν το παιχνίδι έχει τελειώσει (**winner**).

Η κάθε αλυσίδα περιέχει:

- (δείκτη προς) τον πρώτο συμμετέχοντα της αλυσίδας.
- το τρέχον μέγεθος της αλυσίδας.

Προκειμένου να αναδείξετε την προσομοίωσή σας, να υλοποιήσετε μια συνάρτηση **main** η οποία:

- Αρχικά, να διαβάσει από το standard input το πλήθος των συμμετεχόντων N .
- Κατόπιν, να δημιουργεί μια αλυσίδα μήκους N .
- Στη συνέχεια, να επαναλαμβάνει $N - 1$ φορές τη διαδικασία:
 - (α') χρούση τυμπάνου (K φορές, όπου K τυχαίος αριθμός από το 1 έως το 10)
 - (β') σταμάτημα τυμπάνου
 - (γ') ανασύνταξη της αλυσίδας
 - (δ') εκτύπωση της αλυσίδας
- Τέλος, να εκτυπώνεται ο νικητής του παιχνιδιού.

(Σημείωση): οι καρέκλες δεν χρειάζεται να αναπαρασταθούν.