

ΠΡΩΤΗ ΕΡΓΑΣΙΑ
ΣΤΟ ΜΑΘΗΜΑ «ΤΕΧΝΟΛΟΓΙΕΣ ΕΥΦΥΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ
ΡΟΜΠΟΤΙΚΗ»
2016-17

Στην πρώτη εργασία καλείστε να δημιουργήσετε ένα πρόγραμμα LISP. Το πρόγραμμα αυτό θα προσομοιώνει στον Η/Υ τις ενέργειες που επιτρέπεται να γίνονται για την επίλυση ενός προβλήματος και τις καταστάσεις που προκύπτουν από αυτές τις ενέργειες. Με βάση αυτές τις ενέργειες θα λύσετε το πρόβλημα χρησιμοποιώντας το πρόγραμμα σαν μέσο αποτύπωσης των ενεργειών σας και των καταστάσεων που προκύπτουν. Γι' αυτό, θα πρέπει να βρείτε ένα τρόπο να εκτυπώνετε το αποτέλεσμα κάθε ενέργειας. Θα δημιουργήσετε δηλαδή ένα από παιχνίδι.

Δίνονται οκτώ (8) προβλήματα. Οι περιγραφές των προβλημάτων και τα ζητούμενα βρίσκονται σ' ένα αρχείο. Κάθε πρόβλημα έχει ένα αριθμό (1-9). Η εργασία είναι ατομική. Σε καθένα ανατίθεται το πρόβλημα με αριθμό ίδιο με το τελευταίο ψηφίο του Α.Μ. του. Αν το τελευταίο ψηφίο είναι '0' (μηδέν) ή 9 (εννέα) τότε λαμβάνεται υπ' όψιν το προτελευταίο κ.ο.κ. Προφανώς κάποιοι θα έχετε κοινό πρόβλημα. Θα πρέπει όμως να εργαστείτε ανεξάρτητα, ώστε τα προγράμματά σας να μην είναι παρόμοια. Για οποιεσδήποτε δυσκολίες μπορείτε να έρχεστε στο γραφείο μου κάθε Τετάρτη 5.00-7.00 μμ. Η προτεινόμενη προθεσμία παράδοσης είναι η Δευτέρα 24 Απριλίου 2017.

Τα παραδοτέα σας είναι:

- (α) Μια αναφορά, όπου θα περιγράφετε τις απαντήσεις σας στις ερωτήσεις με τις κατάλληλες επεξηγήσεις και την απαραίτητη τεκμηρίωση.
- (β) Αρχείο με τον κώδικα LISP. Το πρόγραμμα θα υπάρχει και στην αναφορά.

Η εργασία αυτή θα συμβάλει κατά 40% στο βαθμό του μαθήματος.

Επίσης, θα γίνει προφορική παρουσίαση/εξέταση της εργασίας.

Η βαθμολογία σας θα βασιστεί σε όλα τα παραπάνω. Θα πρέπει να ξέρετε ότι η αξιολόγηση ενός προγράμματος δεν βασίζεται μόνο στο αν τρέχει σωστά, δηλ. αν βγάζει σωστά αποτελέσματα, αλλά και στο πώς λειτουργεί εσωτερικά, πόσο καλά γραμμένο είναι κλπ. Επίσης, όλα τα προβλήματα δεν είναι ακριβώς της ίδιας δυσκολίας, πράγμα που θα ληφθεί υπ' όψιν.

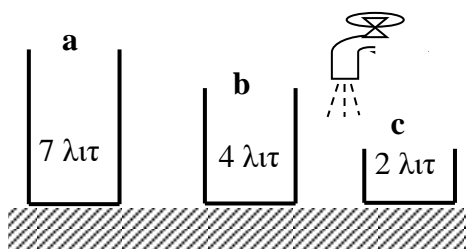
I. Χατζηλυγερούδης

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!

ΠΕΡΙΓΡΑΦΕΣ ΠΡΟΒΛΗΜΑΤΩΝ

1. Πρόβλημα τριών δοχείων

Διαθέτουμε τρία δοχεία που χωρούν 7, 4 και 2 λίτρα νερό αντίστοιχα και δεν διαθέτουν μετρητή. Επίσης διαθέτουμε και μια βρύση. Οι ενέργειες που επιτρέπονται είναι: γέμισμα ενός δοχείου από τη βρύση, άδειασμα ενός δοχείου στο έδαφος, μετάγγιση από ένα δοχείο σ' ένα άλλο μέχρις ότου είτε το πρώτο αδειάσει είτε το δεύτερο γεμίσει.

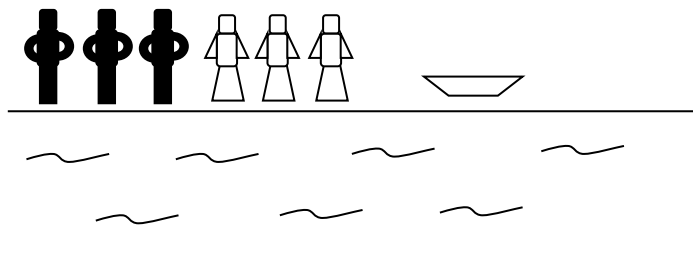


Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο κειμένου (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. γέμισμα ενός δοχείου από τη βρύση, άδειασμα ενός δοχείου στο έδαφος κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα με αρχικά άδεια τα δοχεία κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να απομονωθεί 1 λίτρο νερού στο δοχείο b. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί η όλη αλληλεπίδραση με το πρόγραμμα.

2. Πρόβλημα ιεραποστόλων-κανιβάλων

Στις όχθες ενός ποταμού υπάρχουν συνολικά τρεις ιεραπόστολοι και τρεις κανίβαλοι έτσι ώστε σε κάθε όχθη να υπάρχει ίσος αριθμός ιεραποστόλων και κανιβάλων. Επίσης, υπάρχει και μια βάρκα σε μια από τις όχθες. Θέλουμε όλοι να μεταφερθούν σε μια όχθη. Όμως, η βάρκα για να κινηθεί χρειάζεται τουλάχιστον ένα άτομο, χωρά μόνο μέχρι δύο άτομα και σε καμία περίπτωση δεν πρέπει ο αριθμός των κανιβάλων να είναι μεγαλύτερος από αυτόν των ιεραποστόλων σε κάποια όχθη.



Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετάβαση ενός ιεραποστόλου και ενός κανιβάλου στην απέναντι όχθη κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε), με αρχική κατάσταση αυτή του σχήματος, ώστε να μεταβούν όλοι οι ιεραπόστολοι και οι κανίβαλοι στην απέναντι όχθη. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

3. Πρόβλημα του ανελκυστήρα

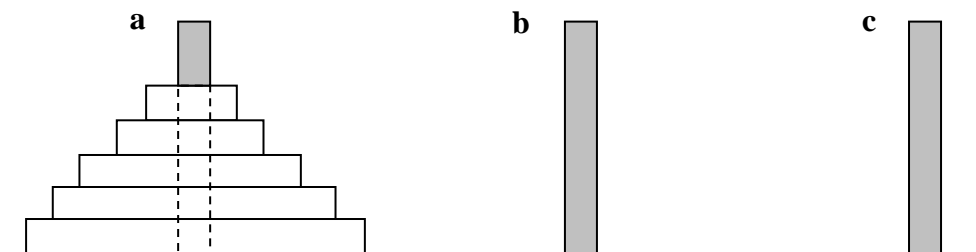
Διαθέτουμε ένα ανελκυστήρα χωρητικότητας 3 ατόμων σε μια πολυκατοικία 4 ορόφων (δηλ. ισόγειο, 1^{ος}, 2^{ος}, 3^{ος}). Ο ανελκυστήρας σε κάποια στιγμή δέχεται κλήση για να μεταβεί σε κάποιο όροφο (είτε από επιβάτη μέσα στον ανελκυστήρα είτε από επιβάτη σε όροφο). Τότε ο ανελκυστήρας μεταβαίνει στον όροφο (εκτός αν βρίσκεται ήδη σ' αυτόν, οπότε παραμένει), βγαίνουν όσοι επιβάτες ήθελαν να μεταβούν στον όροφο και μπαίνουν όσοι χωρούν και είναι πρώτοι στη σειρά από αυτούς που περιμένουν στον όροφο. Μετά ο ανελκυστήρας περιμένει για την επόμενη κλήση, κ.ο.κ. Κάθε επιβάτης που βρίσκεται σ' ένα όροφο έχει σαν τελικό προορισμό κάποιον άλλο όροφο. Όταν ένας επιβάτης βγει στον όροφο του προορισμού του διαγράφεται.

Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικόι (κατανοητοί) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. κλήση του ανελκυστήρα). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα με την εξής αρχική κατάσταση: ανελκυστήρας άδειος και βρίσκεται στο ισόγειο, υπάρχουν 2 επιβάτες στον 1^ο όροφο που θέλουν ο ένας να ανέβει στον 2^ο και ο άλλος στον 3^ο, 2 επιβάτες στον 2^ο όροφο, που ο ένας θέλει να κατέβει στο ισόγειο και ο άλλος να ανέβει στον 3^ο, και 1 επιβάτης στον 3^ο όροφο, που θέλει να κατέβει στο ισόγειο. Να γίνουν οι κατάλληλες ενέργειες, δηλ. κλήσεις του ανελκυστήρα, (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να μεταβούν όλοι οι επιβάτες στους προορισμούς τους. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

4. Πύργος του Ανόϊ

Πέντε δίσκοι διαφορετικού μεγέθους βρίσκονται σε τρεις πασσάλους, έτσι ώστε ο κάθε δίσκος να βρίσκεται πάνω από μεγαλύτερο δίσκο. Η ενέργεια που επιτρέπεται είναι μετακίνηση ενός δίσκου από την κορυφή ενός πασσάλου σ' έναν άλλο αρκεί να μη βρεθεί πάνω από μικρότερο δίσκο.

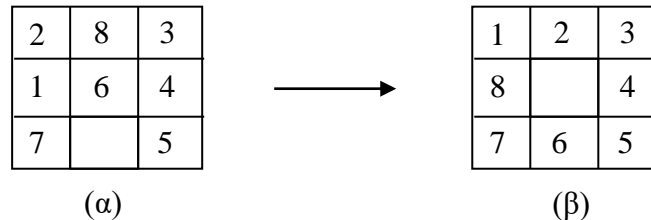


Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετακίνηση ενός δίσκου από τον πρώτο πάσσαλο στον δεύτερο κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα με αρχική κατάσταση αυτή του σχήματος κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να μετακινηθούν όλοι οι δίσκοι στον πάσσαλο c. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

5. Τετράγωνο παζλ 8 ψηφίδων

Έχουμε ένα τετράγωνο πλαίσιο $3 \times 3 = 9$ τετράγωνων θέσεων, που τις 8 θέσεις καταλαμβάνουν τετράγωνα πλακίδια αριθμημένα από το 1 ως το 8, ενώ η ένατη είναι κενή. Τα πλακίδια μπορούν να μετακινούνται πάνω, κάτω, δεξιά και αριστερά, εφ' όσον αυτό είναι δυνατόν, δηλ. δεν βγαίνουν εκτός του πλαισίου και υπάρχει κενό.

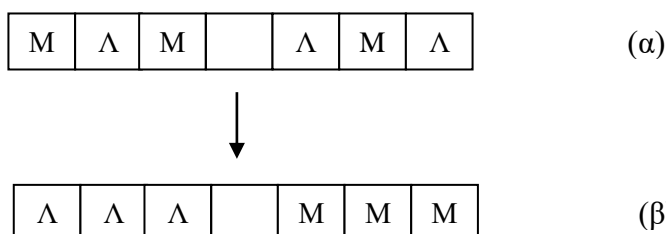


Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετακίνηση ενός πλακιδίου αριστερά, δεξιά κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα με αρχική κατάσταση αυτή του σχήματος (α) κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να καταλήξει στη διάταξη του σχήματος (β). Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

6. Ευθύγραμμο παζλ 6 ψηφίδων

Έχουμε ένα ορθογώνιο πλαίσιο 1X7 θέσεων, που τις 6 θέσεις καταλαμβάνουν τρία λευκά (Λ) και τρία μαύρα (Μ) τετράγωνα πλακίδια, ενώ η έβδομη είναι κενή. Οι κινήσεις που επιτρέπονται είναι: μετακίνηση δεξιά ή αριστερά ενός πλακιδίου είτε απ' ευθείας στην κενή θέση, εφ' όσον είναι δίπλα του, είτε πηδώντας πάνω από άλλα τετραγωνίδια.

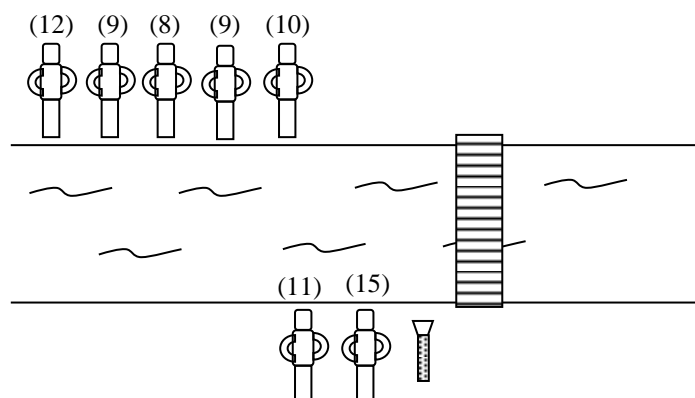


Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετακίνηση ενός πλακιδίου αριστερά κατά μία θέση, δεξιά κατά μία θέση κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα με αρχική κατάσταση αυτή του σχήματος (α) κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να καταλήξει στη διάταξη του σχήματος (β). Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

7. Τουρίστες και φακός

Μια ομάδα 7 τουριστών βρίσκεται νύχτα μοιρασμένη στις δύο όχθες ενός ποταμού. Θέλουν να βρεθούν όλοι μαζί στη μια όχθη. Υπάρχει μια γέφυρα και μόνο ένας φακός. Επιπλέον, η γέφυρα είναι παλιά και μπορούν να την περνάνε το πολύ ανά τρεις. Επίσης, σε καμία πλευρά της γέφυρας δεν πρέπει να βρεθεί μόνος τους ένας τουρίστας, επειδή είναι επικίνδυνο. Κάθε τουρίστας κάνει κάποιο χρόνο για να περάσει τη γέφυρα. Αν περισσότεροι από ένας τουρίστες περάσουν μαζί την γέφυρα, ο χρόνος που χρειάζονται είναι ο χρόνος του πιο αργού.

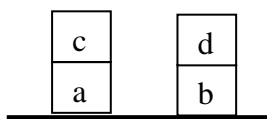


Ζητούνται:

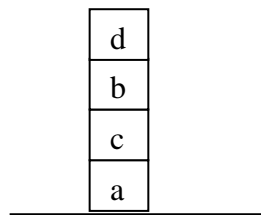
- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικόι (κατανοητοί) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετάβαση δύο τουριστών στην απέναντι όχθη, τριών τουριστών στην απέναντι όχθη κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα, με αρχική κατάσταση αυτή του σχήματος (σε παρένθεση είναι ο χρόνος που κάνει μόνος του ο κάθε τουρίστας), κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να μεταβούν όλοι οι τουρίστες στη μία όχθη. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο) (όπου θα περιλαμβάνεται και ο συνολικός χρόνος διελεύσεων από τη γέφυρα μέχρι τότε). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.

8. Τέσσερις κύβου

Δίνεται το παρακάτω πρόβλημα των τεσσάρων κύβων. Έχουμε τέσσερις κύβους πάνω σ' ένα τραπέζι, π.χ. όπως στη διάταξη του σχ.1α. Μπορούμε να μετακινούμε τους κύβους. Οι κινήσεις που επιτρέπονται είναι: μετακίνηση ενός κύβου από την κορυφή μιας στοίβας στο τραπέζι και μετακίνηση ενός κύβου από το τραπέζι στην κορυφή μιας στοίβας. Έτσι, μπορούμε να αλλάζουμε την κατάσταση των κύβων, π.χ. από αυτήν του σχ. 1α σε αυτήν του σχ. 1β.



Σχ.1α



Σχ.1β

Ζητούνται:

- (α) Να προσδιοριστεί ένας τρόπος απεικόνισης μιας τυχαίας κατάστασης του προβλήματος για την οθόνη και για αρχείο (μπορεί να είναι κοινός και για τα δύο, μπορεί και όχι).
- (β) Να σχεδιαστεί πρόγραμμα LISP που
 - (β1) Να παίρνει την αρχική κατάσταση του προβλήματος είτε από το πληκτρολόγιο είτε από ένα αρχείο (η επιλογή να επαφίεται στον χρήστη). Και στην πρώτη περίπτωση, τα δεδομένα θα καταγράφονται τελικά και σε αρχείο. Ο τρόπος (η μορφή) εισαγωγής των αρχικών δεδομένων του προβλήματος από το πληκτρολόγιο και ο τρόπος (η μορφή) καταγραφής τους στο αρχείο θα οριστούν από σας. Θα πρέπει να είναι φιλικό (κατανοητό) στον (από τον) χρήστη.
 - (β1) Να υλοποιεί τις διάφορες ενέργειες (τελεστές) που επιτρέπει το πρόβλημα (π.χ. μετακίνηση ενός κύβου από την κορυφή μιας στοίβας στο τραπέζι κλπ). Ο ορισμός των ενεργειών πρέπει να είναι τέτοιος ώστε να λαμβάνει υπ' όψιν τους περιορισμούς του προβλήματος και μετά την εκτέλεση κάθε ενέργειας να ενημερώνεται η κατάσταση του προβλήματος.
 - (β2) Να εκτυπώνει (στην οθόνη και στο αρχείο), με μορφές κατανοητές στον χρήστη, την κατάσταση του προβλήματος μετά από κάθε εφαρμογή μιας ενέργειας.
- (γ) Να λύσετε το πρόβλημα, με αρχική κατάσταση αυτή του σχήματος 1α, κάνοντας τις κατάλληλες ενέργειες (μέσω χρήσης των συναρτήσεων-τελεστών που υλοποιήσατε) ώστε να φθάσουμε στην κατάσταση του σχήματος 1β. Μετά από κάθε ενέργεια να εκτυπώνεται η νέα κατάσταση του προβλήματος (και στην οθόνη και στο αρχείο). Να καταγραφεί όλη η αλληλεπίδραση με το πρόγραμμα.