

Τμήμα Εφαρμοσμένης Πληροφορικής
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
Εξάμηνο Β'
Φύλλο Ασκήσεων 6: ΚΑΤΑΚΕΡΜΑΤΙΣΜΟΣ
Μάγια Σατρατζέμη, Γεωργία Κολωνιάρη

- Θεωρήστε ότι έχουμε έναν πίνακα κατακερματισμού με έντεκα θέσεις και τη συνάρτηση κατακερματισμού $h(i) = i \% 11$.
Σχεδιάστε τον πίνακα κατακερματισμού που προκύπτει αν εισαχθούν οι ακόλουθοι ακέραιοι αριθμοί με τη σειρά που δίνονται: 26, 42, 5, 44, 92, 59, 40, 36, 12, 60, 80. Για το χειρισμό των συγκρούσεων χρησιμοποιείται η μέθοδος της αλυσιδωτής σύνδεσης.
- Θεωρώντας ότι η παρακάτω δομή δεδομένων δημιουργήθηκε με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων, να προσδιορίσετε ποιο είναι το περιεχόμενο της κάθε υπολίστας συνωνύμων:

HashTable

1	7
2	1
3	4
4	2
5	6

List

1	6	3
2	8	5
3	11	8
4	12	9
5	13	-1
6	9	12
7	10	11
8	21	-1
9	32	10
10	22	-1
11	5	-1
12	24	-1

- Γράψτε ένα πρόγραμμα για τη δημιουργία και επεξεργασία μιας δομής δεδομένων που δημιουργήθηκε με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων, στην οποία αποθηκεύονται τα στοιχεία των χρηστών μιας Πανεπιστημιακής βιβλιοθήκης. Κάθε εγγραφή περιλαμβάνει τα εξής στοιχεία της κάρτας που εκδίδεται από τη βιβλιοθήκη και χρησιμοποιούν οι χρήστες για το δανεισμό βιβλίων:
 - τον κωδικό του χρήστη (ακέραιος αριθμός – κλειδί κατακερματισμού)
 - το ονοματεπώνυμο του χρήστη (αλφαριθμητικό 20 θέσεων)
 - την ιδιότητα του χρήστη (ακέραιος αριθμός, 1 = postgraduate student, 2 = PhD student, 3 = Professor, 4 = External user)

Η συνάρτηση κατακερματισμού να είναι: $h(i) = (i \% 10) + 1$.

Το πρόγραμμα καθοδηγείται από το ακόλουθο μενού επιλογών:

- | | |
|----------------------------|---|
| 1. Create HashList | (Δημιουργία της δομής δεδομένων) |
| 2. Insert new user | (Εισαγωγή νέου χρήστη της βιβλιοθήκης) |
| 3. Delete a user | (Διαγραφή χρήστη) |
| 4. Search for a user | (Αναζήτηση χρήστη – αν υπάρχει χρήστης με το συγκεκριμένο κωδικό θα εμφανίζονται τα στοιχεία του, αλλιώς θα εμφανίζεται το μήνυμα 'There is no user with code x', όπου x ο κωδικός που δόθηκε προς αναζήτηση) |
| 5. Print list of all users | (Εκτύπωση των στοιχείων όλων των χρηστών – τα στοιχεία κάθε χρήστη θα εμφανίζονται σε διαφορετική γραμμή χωρισμένα με κόμμα) |
| 7. Quit | (Εξοδος) |

Για τις λειτουργίες 2, 3 και 4 ο χρήστης του προγράμματος θα έχει τη δυνατότητα εισαγωγής, διαγραφής ή αναζήτησης αντίστοιχα όσων χρηστών της βιβλιοθήκης επιθυμεί. Η συνέχιση ή όχι κάθε μιας από τις τρεις αυτές λειτουργίες θα ελέγχεται από σχετικό μήνυμα 'Continue Y/N?'.

Για τις λειτουργίες 4 και 5 θα εμφανίζεται η ιδιότητα του χρήστη (postgraduate student, PhD student, Professor, External user) και όχι ο κωδικός 1, 2, 3 ή 4 που αποθηκεύεται στη δομή.

Για τη λειτουργία 5 η εμφάνιση των στοιχείων ομαδοποιημένα σε συνώνυμα, δηλαδή θα γίνεται ως εξής:

Synonyms, collision at position: <θέση κατακερματισμού>

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

....

Synonyms, collision at position: <θέση κατακερματισμού>

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

.....

Το μήνυμα «Synonyms, collision at position: <θέση κατακερματισμού>» θα εμφανίζεται ακόμη και αν υπάρχει μόνο ένα κλειδί στη θέση <θέση κατακερματισμού>

Σας δίνεται ένα στιγμιότυπο, όπου τα στοιχεία καταχωρήθηκαν με την εξής σειρά: 4, 2, 7, 14, 12, 17 (δίνονται μόνο τα κλειδιά των εγγραφών):

LIBRARY USERS:

Synonyms, collision at position: 3

2: [2, VASILEIOY VASILIS, PhD student]

5: [12, KWNSTANTINOY KWSTAS, PhD student]

Synonyms, collision at position: 5

1: [4, ANASTASIOY ANASTASIS, External user]

4: [14, DHMHTRIOY DIMITRIS, External user]

Synonyms, collision at position: 8

3: [7, NIKOLAOY NIKOS, Professor]

6: [17, LIOLIOY ASPA, postgraduate student]

4. Σε έναν εκπαιδευτικό οργανισμό εργάζονται εκπαιδευτικοί διαφόρων ειδικοτήτων. Τα βασικά τους στοιχεία υπάρχουν σε ένα αρχείου κειμένου 'i4f6.txt' (σε διαφορετικές γραμμές για κάθε εκπαιδευτικό):

Όνομα	αλφαριθμητικό 9 χαρακτήρες
Επώνυμο	αλφαριθμητικό 20 χαρακτήρες
Τηλέφωνο	αλφαριθμητικό 10 θέσεων
Κωδικός ειδικότητας	byte (1=Θεολόγοι, 2=Φιλόλογοι, ...20=Πληροφορικοί)

Να γράψετε πρόγραμμα καθοδηγούμενο από μενού επιλογών για την εκτέλεση των παρακάτω λειτουργιών:

1. Create HashList	Διάβασμα των στοιχείων από το αρχείο κειμένου και δημιουργία Δομής Δεδομένων (ΔΔ) που αποθηκεύει και επεξεργάζεται τα στοιχεία της με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων. Το κλειδί σχηματίζεται από το όνομα+«κενό χαρακτήρα»+επώνυμο (αλφαριθμητικό 30 χαρακτήρων). Πχ αν το όνομα είναι «nikos» και το επώνυμο «dimitriou», τότε το κλειδί κατακερματισμού που θα σχηματιστεί είναι: «nikos dimitriou»
2. Insert new teacher	Εισαγωγή των στοιχείων ενός νέου εκπαιδευτικού στη ΔΔ κατακερματισμού με αλυσίδες συνωνύμων
3. Delete a teacher	Διαγραφή ενός εκπαιδευτικού από τη ΔΔ κατακερματισμού με αλυσίδες συνωνύμων
4. Search for a teacher	Αναζήτηση και εμφάνιση των στοιχείων ενός εκπαιδευτικού βάσει ονοματεπώνυμου στη ΔΔ κατακερματισμού με αλυσίδες συνωνύμων
5. Search by subject	Αναζήτηση και εμφάνιση των στοιχείων των εκπαιδευτικών μιας συγκεκριμένης ειδικότητας (ο κωδικός της ειδικότητας [1..20] αποτελεί παράμετρο της διαδικασίας) στη ΔΔ κατακερματισμού με αλυσίδες συνωνύμων
6. EMFANISH DOMHS	Όπως δίνεται στο πρόγραμμα πελάτη (Print HashList(...))
7. EMFANISH PINAKWN (BOH8HTIKH)	Όπως δίνεται στο πρόγραμμα πελάτη (PrintPinakes(...))
6. Quit	Έξοδος

Για τις λειτουργίες 1 και 5 θα πρέπει να υλοποιήσετε 2 νέες συναρτήσεις.

Ο πίνακας κατακερματισμού θα έχει 10 θέσεις και ως συνάρτηση κατακερματισμού θα χρησιμοποιηθεί η εξής:

$$h(i) = (\text{average} \% 10) + 1$$

όπου

$$\text{average} = (\text{κωδικός_πρώτου_χαρακτήρα} + \text{κωδικός_τελευταίου_χαρακτήρα}) / 2$$

Θεωρήστε ότι χρησιμοποιούνται οι ακόλουθοι κωδικοί για τους χαρακτήρες: 'Α' = 1, 'Β' = 2, ..., 'Ζ' = 26. Ο πρώτος και ο τελευταίος χαρακτήρας του ονοματεπώνυμου θα μετατρέπεται στον αντίστοιχο κεφαλαίο χαρακτήρα, εφόσον είναι πεζός. Ο μέσος όρος average θα υπολογίζεται με μια συνάρτηση findAverage, η οποία θα καλείται από τη συνάρτηση HashKey.

Δίνεται ένα στιγμιότυπο εκτέλεσης

MENOU	MENOU
-------	-------

<pre> 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 1 MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 2 Enter teacher's name: Marios Enter teacher's surname: Nikou Enter teacher's phone: 777777777 Enter teacher code: 2 MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 6 HASHLIST STRUCTURE with SYNONYM CHAINING ===== PINAKAS DEIKTWN STIS YPO-LISTES SYNWNMYMN EGGRAFWN: -1 2 5 6 -1 -1 7 1 -1 -1 OI YPO-LISTES TWN SYNWNMYMN EGGRAFWN: TELOS 1HS YPO-LISTAS 2: [maya satratzemi, 6933333332, 20] -> TELOS 2HS YPO-LISTAS 5: [dimitra totsika, 6911555111, 14] -> TELOS 3HS YPO-LISTAS 6: [giannis pappas, 6911111222, 16] -> TELOS 4HS YPO-LISTAS TELOS 5HS YPO-LISTAS TELOS 6HS YPO-LISTAS 7: [nikos ploskas, 6911111662, 20] -> TELOS 7HS YPO-LISTAS 1: [nikos dimitriou, 6911111112, 1] -> 3: [marios nikolaou, 6910001112, 15] -> 4: [maria giannou, 6914441112, 1] -> 8: [Marios Nikou, 777777777, 2] -> TELOS 8HS YPO-LISTAS TELOS 9HS YPO-LISTAS TELOS 10HS YPO-LISTAS H STOIBA TWN ELEY8ERWN 8ESEWN THS LISTAS: 9 <= 10 <= 11 <= 12 <= 13 <= 14 <= 15 <= 16 <= 17 <= 18 <= 19 <= 20 <= 21 <= 22 <= 23 <= 24 <= 25 <= 26 <= 27 <= 28 <= 29 <= 30 <= 31 <= 32 <= 33 <= 34 <= 35 <= 36 <= 37 <= 38 <= 39 <= 40 <= 41 <= 42 <= 43 <= 44 <= 45 <= 46 <= 47 <= 48 <= 4 9 <= 50 <= 51 <= 52 <= 53 <= 54 <= 55 <= 56 <= 57 <= 58 <= 59 <= 60 <= 61 <= 62 <= 63 <= 64 <= 65 <= 66 <= 67 <= 68 <= 69 <= 70 <= 71 <= 72 <= 73 <= 74 <= 75 <= 76 <= 77 <= 78 <= 79 <= 80 <= 81 <= 82 <= 83 <= 84 <= 85 <= 86 <= 87 <= 88 <= 8 9 <= 90 <= 91 <= 92 <= 93 <= 94 <= 95 <= 96 <= 97 <= 98 <= 99 <= 100 <= TELOS MEGE8OS THS LISTAS = 8 ===== </pre>	<pre> 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 7 Hash table -1, 2, 5, 6, -1, -1, 7, 1, -1, -1, Hash List 1) nikos dimitriou, 3 2) maya satratzemi, -1 3) marios nikolaou, 4 4) maria giannou, 8 5) dimitra totsika, -1 6) giannis pappas, -1 7) nikos ploskas, -1 8) Marios Nikou, -1 MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 4 Enter teacher's name for searching in the Hash: Marios Enter teacher's surname for searching in the Hash: Martinidis There is no teacher with name Marios Martinidis MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 4 Enter teacher's name for searching in the Hash: Marios Enter teacher's surname for searching in the Hash: Nikou [Marios Nikou, 777777777, 2] MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE: 5 Enter code: 1 List of teachers with subject code 1: Synonyms, collision at position: 2 Synonyms, collision at position: 3 Synonyms, collision at position: 4 Synonyms, collision at position: 7 Synonyms, collision at position: 8 1: [nikos dimitriou, 6911111112, 1] </pre>
--	---

(συνέχεια στην επόμενη στήλη)	4: [maria giannou, 6914441112, 1] MENOY ----- 1. Create HashList 2. Insert new teacher 3. Delete a teacher 4. Search for a teacher 5. Search by subject 6. EMFANISH DOMHS 7. EMFANISH PINAKWN (BOH8HTIKH) 8. Quit CHOICE:8
-------------------------------	---

5. Σε ένα υπολογιστικό σύστημα είναι επιτρεπτή η πρόσβαση (login) μόνον αν κάθε χρήστης διαθέτει αντίστοιχο λογαριασμό που αποτελείται από:

- μια ταυτότητα χρήστη *user_id* (αλφαριθμητικό 8 θέσεων), η οποία θα αποτελεί και το κλειδί κατακερματισμού και
- έναν κωδικό πρόσβασης *password* (αλφαριθμητικό 6 θέσεων).

Τα *user_id* και τα *password* των χρηστών είναι αποθηκευμένα στο αρχείο κειμένου "I5F6.txt" (σε διαφορετικές γραμμές).

Να γίνει πρόγραμμα που:

- θα διαβάσει τα περιεχόμενα του αρχείου κειμένου και θα δημιουργεί μία ΔΔ που αποθηκεύει και επεξεργάζεται τα στοιχεία της με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων
- θα εμφανίζει το μηνύματα 'USERNAME: ' και 'PASSWORD:', και θα δέχεται από το πληκτρολόγιο την ταυτότητα χρήστη και τον κωδικό πρόσβασης αντίστοιχα. Η εισαγωγή των στοιχείων των χρηστών θα ελέγχεται από το μήνυμα 'New entry Y/N (Y=Yes, N=No)?'.
- Για κάθε χρήστη θα ελέγχεται αν τα στοιχεία του λογαριασμού είναι σωστά και θα εμφανίζονται τα κατάλληλα μηνύματα:
 - i) Αν υπάρχει στη δομή ΔΔ λογαριασμός με τα στοιχεία που έδωσε ο χρήστης θα εμφανίζεται το μήνυμα 'You have logged in to the system'
 - ii) Αν η ταυτότητα χρήστη δεν είναι σωστή θα εμφανίζεται το μήνυμα 'Access is forbidden: Wrong user ID'.
 - iii) Αν η ταυτότητα χρήστη είναι σωστή, αλλά όχι και ο κωδικός πρόσβασης τότε θα εμφανίζεται το μήνυμα 'Access is forbidden: Wrong password'.

Ο πίνακας κατακερματισμού θα έχει 10 θέσεις και ως συνάρτηση κατακερματισμού θα χρησιμοποιηθεί η εξής:

$$h(i) = \text{average} \% 10$$

όπου

$$\text{average} = (\text{κωδικός_ASCII_πρώτου_χαρακτήρα} + \text{κωδικός_ASCII_τελευταίου_χαρακτήρα}) / 2$$

Ο μέσος όρος *average* θα υπολογίζεται με μια συνάρτηση *findAverage*, η οποία θα καλείται από τη συνάρτηση *HashKey*. Η υλοποίηση του ΑΤΔ θα πρέπει να τροποποιηθεί κατάλληλα, ώστε το κλειδί να είναι ένα αλφαριθμητικό 8 χαρακτήρων.

6. Μια συνάρτηση κατακερματισμού θεωρείται καλή όταν αποτιμάται εύκολα και κατανέμει τα στοιχεία σε όλο το εύρος του πίνακα κατακερματισμού, ελαχιστοποιώντας έτσι την πιθανότητα συγκρούσεων. Αν και δεν έχει εντοπιστεί μια μέθοδος κατακερματισμού που να λειτουργεί τέλεια σε όλες τις καταστάσεις, μία δημοφιλής μέθοδος, γνωστή ως *τυχαίος κατακερματισμός*, χρησιμοποιεί μια απλή τεχνική δημιουργίας αριθμών για την 'τυχαία' κατανομή των στοιχείων στον πίνακα κατακερματισμού. Το στοιχείο που αποτελεί το κλειδί *key* μετατρέπεται πρώτα σε ένα μεγάλο τυχαίο ακέραιο χρησιμοποιώντας μια εντολή της μορφής

$$\text{RandomInt} = ((\text{Multiplier} * \text{key}) + \text{Addend}) \% \text{Modulus}$$

και έπειτα υπολογίζεται η θέση στον πίνακα κατακερματισμού με τη γνωστή συνάρτηση

$$\text{Location} = (\text{RandomInt} \% \text{HMax}) + 1$$

Κατάλληλες τιμές για τις σταθερές *Multiplier*, *Addend*, και *Modulus* είναι οι εξής:

$$\text{Modulus} = 65536$$

$$\text{Multiplier} = 25173$$

$$\text{Addend} = 13849$$

Η σταθερά *Modulus* έχει την τιμή 65536 εφόσον ο υπολογιστής έχει λέξεις των 32 bit. Για έναν υπολογιστή με λέξεις των *M*-bit, πρέπει να αντικατασταθεί με την τιμή του $2^{M/2}$.

- Προκειμένου το πρόγραμμα σας να λειτουργεί σε έναν οποιοδήποτε υπολογιστή, κατά την έναρξη της εκτέλεσης θα ζητάτε από τον χρήστη τον αριθμό των bits της λέξης του υπολογιστή και θα υπολογίζετε την τιμή της (μεταβλητής του κυρίως προγράμματος) *Modulus*.
- Στη συνέχεια, χρησιμοποιώντας την παραπάνω συνάρτηση κατακερματισμού δημιουργήστε μια δομή ΔΔ που αποθηκεύει και επεξεργάζεται τα στοιχεία της με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων με ακέραια κλειδιά. Για λόγους απλότητας αποθηκεύστε στη δομή ακέραια κλειδιά με τιμές [1..100]. Θεωρήστε ότι ο πίνακας κατακερματισμού έχει μέγεθος 10 και η λίστα αποθήκευσης των κλειδιών 100. Τέλος, εμφανίστε τα περιεχόμενα του πίνακα κατακερματισμού και των υπολίστων συνωνύμων. Η εμφάνιση θα γίνεται όπως στην άσκηση 3, λειτουργία 5.

Παρατήρηση: αυτή η συνάρτηση κατακερματισμού μπορεί να χρησιμοποιηθεί και με κλειδιά που δεν είναι ακέραιοι αριθμοί, αν πρώτα τα κωδικοποιήσουμε ως ακέραιους. Για παράδειγμα, ένα όνομα μπορεί να κωδικοποιηθεί ως το άθροισμα των κωδικών ASCII ορισμένων ή όλων των χαρακτήρων του.

7. Γράψτε ένα πρόγραμμα για τη δημιουργία και επεξεργασία μιας ΔΔ που αποθηκεύει και επεξεργάζεται τα στοιχεία της με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων, στην οποία αποθηκεύονται τα στοιχεία των συνδρομητών ενός γυμναστηρίου. Κάθε εγγραφή περιλαμβάνει τα εξής στοιχεία της κάρτας μέλους που δίνεται σε κάθε συνδρομητή όταν εγγράφεται στο γυμναστήριο:

- τον κωδικό (ακέραιος αριθμός – κλειδί κατακερματισμού)
- το ονοματεπώνυμο του συνδρομητή (*αλφαριθμητικό 20 θέσεων*)
- το ποσό που χρωστάει ο πελάτης στο γυμναστήριο (*ακέραιος*)

Η συνάρτηση κατακερματισμού να είναι: $h(i) = i \% 11$.

Το πρόγραμμα καθοδηγείται από το ακόλουθο μενού επιλογών:

- | | | |
|----|-------------------------|---|
| 1. | Create HashList | (Δημιουργία της δομής δεδομένων) |
| 2. | Insert new member | (Εισαγωγή νέου μέλους) |
| 3. | Delete a member | (Διαγραφή μέλους – η διαγραφή δεν μπορεί να πραγματοποιηθεί αν ο λογαριασμός του μέλους δεν έχει εξοφληθεί. Σε αυτή την περίπτωση η διαγραφή δεν πραγματοποιείται και εμφανίζεται το μήνυμα ‘Member must arrange his/her bill first’) |
| 4. | Search for a member | (Αναζήτηση χρήστη – αν υπάρχει μέλος με το συγκεκριμένο κωδικό θα εμφανίζονται τα στοιχεία του, αλλιώς θα εμφανίζεται το μήνυμα ‘There is no member with code x’, όπου x ο κωδικός που δόθηκε προς αναζήτηση) |
| 5. | Update member’s account | (Ενημέρωση του λογαριασμού του μέλους – ο χρήστης δίνει τον κωδικό του μέλους και το ποσό και ενημερώνεται κατάλληλα ο λογαριασμός του. Όταν το μέλος πληρώνει στο γυμναστήριο κάποιο ποσό ο χρήστης του προγράμματος το πληκτρολογεί με αρνητικό πρόσημο.) |
| 6. | Print list of synonyms | (Ο χρήστης δίνει το κλειδί ενός μέλους του γυμναστηρίου και εμφανίζονται τα περιεχόμενα της υπολίστας των συνωνύμων στην οποία ανήκει). Η εμφάνιση θα γίνεται όπως στην άσκηση 3, λειτουργία 5. |
| 7. | Quit | (Εξοδος) |

Για τις λειτουργίες 2, 3, 4 και 5 ο χρήστης του προγράμματος θα έχει τη δυνατότητα εισαγωγής, διαγραφής, αναζήτησης ή ενημέρωσης αντίστοιχα όσων χρηστών της βιβλιοθήκης επιθυμεί. Η συνέχιση ή όχι κάθε μιας από τις λειτουργίες θα ελέγχεται από σχετικό μήνυμα ‘Continue Y/N?’.

8. Γράψτε ένα πρόγραμμα για τη δημιουργία και επεξεργασία μιας δομής δεδομένων που δημιουργήθηκε με την τεχνική του κατακερματισμού με αλυσίδες συνωνύμων, στην οποία αποθηκεύονται τα στοιχεία των απόμων που σιτίζονται στο εστιατόριο του πανεπιστημίου. Κάθε εγγραφή περιλαμβάνει τα εξής στοιχεία της κάρτας που εκδίδεται από το εστιατόριο και χρησιμοποιούν οι χρήστες για τη σίτισή τους::

- τον κωδικό του χρήστη (ακέραιος αριθμός – κλειδί κατακερματισμού)
- το ονοματεπώνυμο του χρήστη (*αλφαριθμητικό 20 θέσεων*)
- την ιδιότητα του χρήστη (ακέραιοι αριθμοί, 1 = student, 2 = postgraduate student, 3 = teacher, 4 = visitor)

Η συνάρτηση κατακερματισμού να είναι: $h(i) = (i \% 10)$.

Το πρόγραμμα καθοδηγείται από το ακόλουθο μενού επιλογών:

- | | | |
|----|-------------------------|--|
| 1. | Create HashList | (Δημιουργία της δομής δεδομένων) |
| 2. | Insert new user | (Εισαγωγή νέου χρήστη στο εστιατόριο) |
| 3. | Delete a user | (Διαγραφή χρήστη) |
| 4. | Search for a user | (Αναζήτηση χρήστη – αν υπάρχει χρήστης με το συγκεκριμένο κωδικό θα εμφανίζονται τα στοιχεία του, αλλιώς θα εμφανίζεται το μήνυμα ‘There is no user with code x’, όπου x ο κωδικός που δόθηκε προς αναζήτηση) |
| 5. | Print list of all users | (Εκτύπωση των στοιχείων όλων των χρηστών – τα στοιχεία κάθε χρήστη θα εμφανίζονται σε διαφορετική γραμμή χωρισμένα με κόμμα) |
| 7. | Quit | (Εξοδος) |

Για τις λειτουργίες 2, 3 και 4 ο χρήστης του προγράμματος θα έχει τη δυνατότητα εισαγωγής, διαγραφής ή αναζήτησης αντίστοιχα όσων χρηστών του εστιατορίου επιθυμεί. Η συνέχιση ή όχι κάθε μιας από τις τρεις αυτές λειτουργίες θα ελέγχεται από σχετικό μήνυμα ‘Continue Y/N?’.

Για τις λειτουργίες 4 και 5 θα εμφανίζεται η ιδιότητα του χρήστη (προπτυχιακός φοιτητής, Μεταπτυχιακός φοιτητής, Διδάσκων, Επισκέπτης) και όχι ο κωδικός 1, 2, 3 ή 4 που αποθηκεύεται στη δομή.

Για τη λειτουργία 5 η εμφάνιση των στοιχείων ομαδοποιημένα σε συνώνυμα, δηλαδή θα γίνεται ως εξής;

Synonyms, collision at :position <θέση κατακερματισμού>

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

....

Synonyms, collision at :position <θέση κατακερματισμού>

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

Θέση στη λίστα εγγραφών: [<κλειδί>, <ονοματεπώνυμο>, <ιδιότητα>]

.....

Σας δίνεται ένα στιγμιότυπο, όπου τα στοιχεία καταχωρήθηκαν με την εξής σειρά: 2, 4, 7, 12, 14, 17 (δίνονται μόνο τα κλειδιά των εγγραφών):

USERS:

Synonyms, collision at position: 2

0: [2, ANDREOY, student]

3: [12, KWSTAKIS, teacher]

Synonyms, collision at position: 4

1: [4, NINOY, postgraduate student]

4: [14, LEONIDOY, visitor]

Synonyms, collision at position: 7

2: [7, APOSTOLOY, postgraduate student]

5: [17, TOTHS, postgraduate student]

9. Σε ένα οδοντιατρείο, διατηρείται ένας κατάλογος με τις επισκέψεις των ασθενών για κάθε ημέρα, όπου κάθε εγγραφή έχει τα εξής στοιχεία:

Όνομα (Αλφαριθμητικό 20 θέσεων, αποθηκεύεται η τιμή κατακερματισμού)

Είδος υπηρεσίας (int: 1-Λεύκανση, 2-Καθαρισμός, 3-Εξαγωγή)

Ποσό πληρωμής(double)

Προκειμένου να προστατεύονται τα ιδιωτικά δεδομένα των πελατών, το όνομα αποθηκεύεται ως ένας αριθμός. Γράψτε ένα πρόγραμμα το οποίο διαχειρίζεται τη βάση δεδομένων του οδοντιατρείου, και το οποίο διαθέτει μενού για:

1. Δημιουργία της βάσης.
2. Εισαγωγή εγγραφής επίσκεψης.
3. Εμφάνιση επίσκεψης πελάτη, διαβάζοντας το όνομά του.
4. Έξοδο

Η συνάρτηση κατακερματισμού υπολογίζεται ως

$$H(x)=x\%p+1, p=1543$$

Όπου x είναι το άθροισμα των γραμμάτων του ονόματος, κωδικοποιημένα ως

$$A \rightarrow 1, B \rightarrow 2 \dots Z \rightarrow 26$$

Και πολλαπλασιασμένα με τη θέση τους στο string. Π.Χ. το αλφαριθμητικό NICK έχει ως άθροισμα το

$$1 \cdot 14 + 2 \cdot 9 + 3 \cdot 3 + 4 \cdot 11$$

ΠΡΟΣΟΧΗ Το όνομα δεν αποθηκεύεται ως αλφαριθμητικό, αλλά ως το άθροισμα των γραμμάτων του!

<pre>----- MENOU 1. CREATE DATABASE 2. INSERT APPOINTMENT 3. PRINT CLIENT'S APPOINTMENTS 4. EXIT CHOICE: 1 ----- MENOU 1. CREATE DATABASE 2. INSERT APPOINTMENT 3. PRINT CLIENT'S APPOINTMENTS 4. EXIT CHOICE: 2 Enter the client's Name:GEORGE Enter the service: 1-Whitening 2-Cleaning 3-Extraction 1 Enter the amount paid:50 Continue Y/N:Y Enter the client's Name:NICK Enter the service: 1-Whitening 2-Cleaning 3-Extraction 2 Enter the amount paid:20.50</pre>	<pre>Enter the client's Name:GREGEO Enter the service: 1-Whitening 2-Cleaning 3-Extraction 3 Enter the amount paid:100 Continue Y/N:N ----- MENOU 1. CREATE DATABASE 2. INSERT APPOINTMENT 3. PRINT CLIENT'S APPOINTMENTS 4. EXIT CHOICE: 3 Enter the client's Name:GEORGE Service: 1 Amount Paid: 50.00 ----- MENOU 1. CREATE DATABASE 2. INSERT APPOINTMENT 3. PRINT CLIENT'S APPOINTMENTS 4. EXIT CHOICE: 3 Enter the client's Name:GREGEO Service: 3 Amount Paid: 100.00 ----- MENOU</pre>
--	--

Continue Y/N:Y	1. CREATE DATABASE 2. INSERT APPOINTMENT 3. PRINT CLIENT'S APPOINTMENTS 4. EXIT CHOICE:
----------------	---

10. Προσομοιώστε τη λειτουργία μίας απλής γλώσσας προγραμματισμού η οποία διαθέτει τις εξής δυνατότητες, οι οποίες καλούνται μέσω μενού:

1. Δήλωση μεταβλητής μήκους το πολύ 5 χαρακτήρων και ανάθεση τιμής στη μεταβλητή.
2. Πρόσθεση δύο μεταβλητών
3. Πολλαπλασιασμός δύο μεταβλητών.

Η υλοποίηση να γίνει χρησιμοποιώντας λίστα κατακερματισμού με συνώνυμα και η συνάρτηση κατακερματισμού να είναι η εξής:

$$H(x)=x\%p+1, p=31$$

Όπου x είναι το άθροισμα των γραμμάτων του ονόματος της μεταβλητής, κωδικοποιημένα ως

$$A \rightarrow 1, B \rightarrow 2 \dots Z \rightarrow 26$$

Και πολλαπλασιασμένα με τη θέση τους στο string. Π.Χ. το αλφαριθμητικό NICK έχει ως άθροισμα το $1 \cdot 14 + 2 \cdot 9 + 3 \cdot 3 + 4 \cdot 11$

```

      MENΟΥ
-----
1. DECLARE VARIABLES
2. MULTIPLY VARIABLES
3. ADD VARIABLES
4. EXIT

CHOICE: 1
Enter the variable name: A
Enter variable value: 20

Continue Y/N: Y
Enter the variable name: B
Enter variable value: 15

Continue Y/N: N

      MENΟΥ
-----
1. DECLARE VARIABLES
2. MULTIPLY VARIABLES
3. ADD VARIABLES
4. EXIT

CHOICE: 2
Enter the first variable: A
Enter the second variable: B
A*B=300

      MENΟΥ
-----
1. DECLARE VARIABLES
2. MULTIPLY VARIABLES
3. ADD VARIABLES
4. EXIT

CHOICE: 3
Enter the first variable: A
Enter the second variable: B
A+B=35

      MENΟΥ
-----
1. DECLARE VARIABLES
2. MULTIPLY VARIABLES
3. ADD VARIABLES
4. EXIT

```