

Εργασία 1ου Εξαμήνου – Δομημένος Προγραμματισμός.

– Ιωάννης Δημουλιός – ΑΕΜ: 10641

– Αθανάσιος Γεωργούσης – ΑΕΜ: 10634

– Διδάσκοντες: Χρήστος Σεβαστιάδης, Αθανάσιος Σιαχούδης.

– Θεσσαλονίκη, 30/01/22.

***Σκοπός – Γενικές Πληροφορίες.***

Το να δουλεύει κανείς με πίνακες μπορεί να είναι αρκετά κουραστικό και εκνευριστικό αφού απαιτούνται πολλές αριθμητικές πράξεις χωρίς ιδιαίτερα σύνθετη λογική. Παρατηρούμε λοιπόν, πως μπορούμε να αξιοποιήσουμε τον υπολογιστή για τις αριθμητικές πράξεις, το οποίο απασχολούν για ελάχιστο χρόνο σε σχέση με τον άνθρωπο και να τυποποιήσουμε την λογική τους με την χρήση κώδικα. Γι’ αυτό επιλέξαμε να δημιουργήσουμε μια εφαρμογή που θα μας βοηθήσει στις πράξεις μεταξύ τους. Μπορούμε να δημιουργούμε πίνακες είτε άμεσα μέσα από το πρόγραμμα, είτε διαβάζοντας τους μέσα από αρχεία. Το πρόγραμμα γράφτηκε στην προγραμματιστική γλώσσα c, όπως διδαχθήκαμε στο 1ο εξάμηνο στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του ΑΠΘ.

***Βασικές έννοιες.***

Ακολουθούν μερικές πολύ βασικές συναρτήσεις και δομές που φτιάξαμε, στις οποίες δομούνται πολλές λειτουργίες του προγράμματος.

1. Στο πρόγραμμά μας οι πίνακες αποθηκεύονται ως δομή “matrix” με στοιχεία:

* char id[10], είναι η (μοναδική) ταυτότητα του πίνακα.
* int rows, πλήθος γραμμών.
* int cols, πλήθος στηλών.
* float pin[], περιέχει τα στοιχεία του πίνακα με την μορφή pin[i \* cols +j], όπου i οι γραμμή του στοιχείου και j η στήλη (ξεκινώντας από το 0 x 0).

Παρατηρείστε πως τα στοιχεία αποθηκεύονται ως float.

1. Ο κάθε πίνακας αποθηκεύεται σε δυναμική συστοιχία με την μορφή struct matrix\_list και στοιχεία:

* size\_t size, μέγεθος συστοιχίας.
* struct matrix \*e[], πίνακας δεικτών σε struct matrix.

1. Η αναζήτηση μέσα στο διάνυσμα μνήμης γίνεται με την συνάρτηση search\_id με ορίσματα:

* char name [10], η ταυτότητα του πίνακα.
* matrix\_list \*\*v

Η συνάρτηση επιστρέφει τον δείκτη του πίνακα με ταυτότητα name μέσα από το διάνυσμα μνήμης αν υπάρχει. Αλλιώς επιστρέφει την προκαθορισμένη τιμή -1.

3 .1 Σαν «παιδί» της συνάρτησης search\_id, υπάρχει και η συνάρτηση indexing() η οποία ουσιαστικά συνδέει το γνωστό όνομα ενός πίνακα, με το index του στο διάνυσμα μνήμης. Μόλις εντοπίσει τον πίνακα με το όνομα που έδωσε ο χρήστης, επιστρέφει το index του. Παίρνει ορίσματα:

* matrix\_list \*v
* char message [200]

Η συστοιχία χαρακτήρων είναι το μήνυμα που θα δεχτεί ο χρήστης ώστε να ξέρει να δώσει το όνομα του πίνακα. Πάντα θα έχει την μορφή “Give the name of…”.

1. Για να την αρχικοποίηση των πινάκων καλούμε την συνάρτηση init\_matrix η οποία παίρνει τα εξής ορίσματα:

* struct matrix \*\*ptr, δείκτη σε δείκτη σε πίνακα
* int r, πλήθος γραμμών νέου πίνακα
* int c, πλήθος στηλών νέου πίνακα
* char name[10], ταυτότητα νέου πίνακα

η οποία δεσμεύει την κατάλληλη μνήμη, δεδομένων των σειρών, των στηλών και του id. Γίνεται έλεγχος για την ορθή δέσμευση μνήμης-στην περίπτωση σφάλματος τερματίζεται το πρόγραμμα ακαριαία με το μήνυμα «ERROR. Abort ship! »

1. Επίσης πολύ βασική συνάρτηση είναι η insert, η οποία αποθηκεύει τους πίνακες στο διάνυσμα μνήμης του οποίου διπλασιάζει τον διαθέσιμο χώρο σε περίπτωση που δεν επαρκεί για την εισαγωγή του νέου πίνακα και παίρνει ορίσματα:

* struct matrix \*m, νέος πίνακας
* struct matrix\_list \*\*v, διάνυσμα μνήμης.

Όπως και στην συνάρτηση init\_matrix, γίνεται έλεγχος για την ορθή δέσμευση μνήμης και στην περίπτωση σφάλματος το πρόγραμμα τερματίζεται ακαριαία με το ίδιο μήνυμα.

1. Τέλος, για την διαγραφή πινάκων χρησιμοποιείται η συνάρτηση delete\_matrix η οποία αποδεσμεύει την μνήμη του διαγραφόμενου πίνακα.

Αυτές ήταν οι πιο βασικές συναρτήσεις στις οποίες δομείται το πρόγραμμα. Εκτός από αυτές όμως, χρησιμοποιείται πληθώρα άλλων συναρτήσεων για τις πιο ειδικές εργασίες (π.χ. τον πολλαπλασιασμό δύο πινάκων), αν ο χρήστης το επιλέξει. Όλες οι συναρτήσεις φαίνονται στο διάγραμμα που περιγράφει σχηματικά και την δομή του προγράμματος.

***Δομή-Τρόπος χρήσης.***

Το πρόγραμμα έχει τη μορφή ένθετων διαδραστικών μενού. Ο χρήστης μόλις το τρέξει καλείται να πληκτρολογήσει τον κωδικό της λειτουργίας που επιθυμεί να χρησιμοποιήσει (π.χ. για να προχωρήσει ο χρήστης στο 1ο μενού «Create Matrix.(1)», πληκτρολογεί τον κωδικό που βρίσκεται στην παρένθεση, εδώ το ‘1’ ). Κάθε φορά που εμφανίζεται ένα ένθετο μενού, ο χρήστης ενημερώνεται κατάλληλα για το πώς μπορεί να συνεχίσει. Σαφώς, καθ΄ όλη την διάρκεια που τρέχει το πρόγραμμα, μπορεί να πληκτρολογήσει ‘m’ για να του εμφανιστεί το Main Menu (όπως του εξηγεί άλλωστε το μήνυμα μόλις τρέξει το πρόγραμμα) στο οποίο εμπεριέχονται όλοι οι διαθέσιμοι κωδικοί μαζί με την λειτουργία τους. Για τον τερματισμό του προγράμματος ο χρήστης πληκτρολογεί ‘q’ (Quit).

Η δομή καθώς και οι συναρτήσεις φαίνονται καλύτερα με την βοήθεια του δενδροδιαγράμματος της τελευταίας σελίδας.

***Χρήση αρχείων.***

1. Εισαγωγή

Ο χρήστης καλείται να δώσει προσοχή όταν εισάγει αρχεία με πίνακες. Καταρχάς, οι πίνακες πρέπει να έχουν αυστηρά τυποποιημένη μορφοποίηση, η οποία είναι η εξής :

# of matrices in the file.<ENTER>

Name<ENTER>

(# of rows)<SPACE>(# of cols)<ENTER>

a(1x1)<SPACE> a(1x2) ... A(1xN)<ENTER>

a(2x1) … …

… …

a(Mx1)<SPACE>a(Mx2) ... a(MxN)<ENTER>

Όπου Μ και Ν είναι τα προαναφερόμενα (# of rows) και (# of cols), αντίστοιχα. Name θα γίνει η ταυτότητα του εισαγόμενου πίνακα. Αν πρόκειται για πολλούς πίνακες, οι κενές γραμμές μεταξύ τους δεν επηρεάζουν την εφαρμογή.

Προσοχή! Εάν σε ένα αρχείο εισαγωγής υπάρχει πίνακας με το ίδιο όνομα με έναν υφιστάμενο πίνακα, τότε ο «καινούργιος» παραλείπεται!

1. Εξαγωγή

Να σημειωθεί πως η παραπάνω θα είναι η μορφή των πινάκων και κατά την εξαγωγή τους σε αρχεία. Όλη η πληροφορία που υπήρχε στο αρχείο που έδωσε ο χρήστης θα αντικατασταθεί από τον πίνακα τον οποίο εξάγει.

***Δενδροδιάγραμμα.***

Στο δενδροδιάγραμμα φαίνονται σχηματικά όλες οι λειτουργίες του προγράμματος. Παράλληλα μπορείτε να δείτε όλες τις συναρτήσεις που φτιαχτήκαν και χρησιμοποιήθηκαν για την λειτουργία του. Να σημειωθεί πως οι βασικές συναρτήσεις, στις οποίες έγινε αναφορά στην αρχή της εργασίας, έχουν παραληφθεί από το σχεδιάγραμμα αφού στόχος του είναι η κατανόηση των λειτουργιών ξεχωριστά και η προσθήκη τους θα την δυσκόλευε. (Βρίσκεται στη τελευταία σελίδα)

