Τεκμηρίωση Εργασίας στους Πίνακες

Μητουσούδης Βασίλειο ς Μηνάγιας Δημήτριος

Αποφασίσαμε να ασχοληθούμε με την δεύτερη προαιρετική εργασία που αναθέσατε, που έχει να κάνει με Πράξεις μεταξύ Πινάκων.

Το πρόγραμμα το οποίο γράψαμε αρχικά δίνει την δυνατότητα στον χρήστη να επιτελέσει μια από τις ακόλουθες διαδικασίες:

* Πράξεις μεταξύ πινάκων, με τιμές που βάζει ο χρήστης
* Πράξεις μεταξύ διανυσμάτων, με τιμές που βάζει ο χρήστης
* Δημιουργία αρχείου που περιέχει ένα διάνυσμα
* Δημιουργία αρχείου που περιέχει έναν πίνακα
* Πράξεις μεταξύ πινάκων που φορτώνονται από ήδη υπάρχοντα αρχεία
* Πράξεις μεταξύ διανυσμάτων που φορτώνονται από ήδη υπάρχοντα αρχεία

Οι πράξεις που εκτελούνται μεταξύ πινάκων, όταν θέτει τις τιμές ο χρήστης είναι η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, η εύρεση του αναστρόφου, η εύρεση του αντιστρόφου πίνακα και η διακρίνουσα πίνακα.

Οι πράξεις που εκτελούνται μεταξύ διανυσμάτων, όταν θέτει ο χρήστης τις τιμές είναι η πρόσθεση, η αφαίρεση, το εσωτερικό γινόμενο, το εξωτερικό γινόμενο (για 3x3) πίνακες, το μέτρο διανύσματος, η γωνία μεταξύ δύο διανυσμάτων και το μικτό γινόμενο.

Οι πράξεις που εκτελούνται μεταξύ πινάκων, όταν οι τιμές φορτώνονται από κάποιο αρχείο είναι η αφαίρεση, ο πολλαπλασιασμός, η εύρεση του αντιστρόφου πίνακα και η διακρίνουσα πίνακα.

Οι πράξεις που εκτελούνται μεταξύ διανυσμάτων, όταν οι τιμές φορτώνονται από κάποιο αρχείο είναι η πρόσθεση, η αφαίρεση, το εσωτερικό γινόμενο, το εξωτερικό γινόμενο (για 3x3) πίνακες, το μέτρο διανύσματος, η γωνία μεταξύ δύο διανυσμάτων και το μικτό γινόμενο.

ΒΙΒΛΙΟΘΗΚΕΣ

Χρησιμοποιήσαμε τις βιβλιοθήκες <stdio.h>, <stdlib.h>, <string.h>, <math.h>, έτσι ώστε να φορτώσουμε στο αρχείο πηγαίου κώδικα έτοιμες συναρτήσεις όπως η printf, scanf, fscanf, fprintf, strcat, pow, rand τις οποίες χρησιμοποιήσαμε με κατάλληλο τρόπο για να δημιουργήσουμε το πρόγραμμά μας.

ΣΥΝΑΡΤΗΣΕΙΣ

Προφανώς, για να περατωθεί το εγχείρημά μας, έτσι ώστε να είναι αφενός λειτουργικό και αφετέρου ευανάγνωστο στον κάθε τρίτο που θα ήθελε να το διαβάσει χρησιμοποιήσαμε ένα σύνολο συναρτήσεων που επιτελούν τις προαναφερθείσες πράξεις.

Σε κάθε συνάρτηση, για να προσπελάσουμε τους πίνακες χρησιμοποιήσαμε την μέθοδο της φυσαλίδας.

Πιο συγκεκριμένα, έχουμε δημιουργήσει τις παρακάτω συναρτήσεις:

* void vectormagnitude()

Συνάρτηση μέτρου διανύσματος που φορτώθηκε από αρχείο.

* void createvector()

Συνάρτηση η οποία αποθηκεύει σε αρχείο ένα διάνυσμα τις συντεταγμένες του οποίου τις θέτει ο χρήστης.

* void createMatrix()

Συνάρτηση η οποία αποθηκεύει σε αρχείο έναν πίνακα τα στοιχεία του οποίου τα θέτει ο χρήστης.

* void vectorsubtraction()

Συνάρτηση που υπολογίζει την διαφορά διανυσμάτων που φορτώθηκαν από αρχείο.

* void vectoraddition()

Συνάρτηση που υπολογίζει το άθροισμα διανυσμάτων που φορτώθηκαν από αρχείο.

* void matrixsubtraction()

Συνάρτηση που υπολογίζει την διαφορά πινάκων που φορτώθηκαν από αρχείο

* void matrixaddition()

Συνάρτηση που υπολογίζει το άθροισμα πινάκων που φορτώθηκαν από αρχείο.

* void matrixMultiplication()

Συνάρτηση που υπολογίζει το γινόμενο πινάκων που φορτώθηκαν από αρχείο.

* void generateMatrix()

Συνάρτηση που δημιουργεί στον ίδιο φάκελο που βρίσκεται ο πηγαίος κώδικας και το εκτελέσιμο αρχείο ένα αρχείο τύπου .txt το οποίο περιέχει τις τιμές του πίνακα που δόθηκαν με ψευδοτυχαίους rand() αριθμούς.

* void generatevector()

Συνάρτηση που δημιουργεί στον ίδιο φάκελο που βρίσκεται ο πηγαίος κώδικας και το εκτελέσιμο αρχείο ένα αρχείο τύπου .txt το οποίο περιέχει τις τιμές του διανύσματος που δόθηκαν με ψευδοτυχαίους rand() αριθμούς.

* float determinant(float [][25], float)

Συνάρτηση που υπολογίζει την διακρίνουσα πίνακα που διαβάστηκε από αρχείο. Η συνάρτηση αυτή υπολογίζει την διακρίνουσα πίνακα, ακολουθώντας τον τύπο , όπου , με την διακρίνουσα που δημιουργείται αν σβηστεί η i-γραμμή και j-στήλη.

* void cofactor(float [][25], float)

Συνάρτηση που υπολογίζει τον αντίστροφο πίνακα και στο εσωτερικό της υπάρχουν οι συναρτήσεις που δίνουν την διακρίνουσα και τον ανάστροφο πίνακα, απαραίτητες συναρτήσεις για την εύρεση του αντιστρόφου.

* void transpos(float [][25], float [][25], float)
* void add(int m[10][10], int n[10][10], int sum[10][10],int q)

Συνάρτηση που προσθέτει δύο πίνακες με τιμές που έβαλε ο χρήστης.

* void subtract(int m[10][10], int n[10][10], int result[10][10],int q)

Συνάρτηση που αφαιρεί δύο πίνακες με τιμές που έβαλε ο χρήστης.

* void multiply(int m[10][10], int n[10][10], int result[10][10],int q)

Συνάρτηση που πολλαπλασιάζει δύο πίνακες με τιμές που έβαλε ο χρήστης.

* void transpose(int matrix[10][10], int trans[10][10],int q)

Συνάρτηση που υπολογίζει τον ανάστροφο πίνακα που έχει θέσει ο χρήστης.

* void display(int matrix[10][10],int q)

Συνάρτηση που προβάλλει πίνακα

* float magnitude(int a[10],int s)

Συνάρτηση που υπολογίζει το μέτρο διανύσματος που τέθηκε από τον χρήστη.

* float dotprod(int a[10],int b[10],int s)

Συνάρτηση που υπολογίζει εσωτερικό γινόμενο δύο διανυσμάτων που τέθηκαν από τον χρήστη.

* void cofactor(float num[25][25], float f)

Η συνάρτηση αυτή υπολογίζει τον αντίστροφο πίνακα και στο εσωτερικό της υπάρχουν οι συναρτήσεις που δίνουν την διακρίνουσα και τον ανάστροφο πίνακα, απαραίτητες συναρτήσεις για την εύρεση του αντιστρόφου.

* float determinant(float g[25][25], float k)

Συνάρτηση που υπολογίζει την διακρίνουσα πίνακα, ακολουθώντας το , όπου , με την διακρίνουσα που δημιουργείται αν σβηστεί η i-γραμμή και j-στήλη.

* void transpos(float num[25][25], float fac[25][25], float r)

Συνάρτηση που υπολογίζει τον ανάστροφο πίνακα με τιμές που έθεσε ο χρήστης στην κονσόλα (και εδώ χρησιμοποιείται η πολύ σημαντική συνάρτηση determinant).

* int main(void) (Η κύρια συνάρτηση του προγράμματος)

Στην κύρια συνάρτηση, χρησιμοποιώντας την switch μπορέσαμε να δώσουμε τις διαφορετικές επιλογές στον χρήστη, ανάλογα με το τι θέλει να πετύχει. Αρχικά, φορτώνονται τα δεδομένα από τις συναρτήσεις πινάκων με τιμές που έθεσε ο χρήστες, έπειτα φορτώνονται οι συναρτήσεις των διανυσμάτων (διαχείριση αρχείων) και τέλος η φόρτωση συναρτήσεων που κάνουν πράξεις πινάκων με δεδομένα από πίνακες.

Το αρχείο του εκτελέσιμου κώδικα, το αρχείο του πηγαίου κώδικα και όλα τα αρχεία που περιέχουν αποθηκευμένους πίνακες και διανύσματα πρέπει να βρίσκονται στον ίδιο φάκελο.

Επίσης, κατά την ονοματοδοσία αρχείων θα πρέπει να αποφεύγονται τα κενά, ενώ δεν χρειάζεται η πρόσθεση του επιθέματος .txt, εφόσον έχει χρησιμοποιηθεί η συνάρτηση strcat(scanned\_name\_from\_user, “.txt”).

ΒΙΒΛΙΟΓΡΑΦΙΑ (MLA 9 CITATION)

* “R/c\_programming.” *Reddit*, https://www.reddit.com/r/C\_Programming/.
* Stack Overflow. “Stack Overflow - Where Developers Learn, Share, & Build Careers.” Stack Overflow, 2019, stackoverflow.com/.
* “C Program to Find Inverse of a Matrix.” Sanfoundry, 10 June 2014, www.sanfoundry.com/c-program-find-inverse-matrix/. Accessed 24 Jan. 2022.
* “Matrix Multiplication Using Text File in C» DrukLearn.” DrukLearn, 11 Feb. 2021, learn.drukinfotech.com/matrix-multiplication-using-text-file-in-c/. Accessed 24 Jan. 2022.
* “Mathematics Stack Exchange.” Mathematics Stack Exchange, 2019, math.stackexchange.com/.
* Cheng, Harry H. *C for Engineers and Scientists*. New York, Mcgraw-Hill Education Create, 2014.
* “Matrix Calculator.” CodeProject, 26 May 2004, www.codeproject.com/Articles/7171/Matrix-Calculator. Accessed 24 Jan. 2022.