Felhasználói dokumentáció

Balog Ádám Márk (ELAO0E)

December 2019

1. Függvénykönyvtár célja

A könyvtár lehetőséget nyújt a fejlesztőnek mátrixokkal kapcsolatos függvények használatára, ezáltal segítve, motiválva ezen területen való tevékenységét.

2. Függvénykönyvtár használata

Fejlécben meghívandó:

#include "matrix.h"

Ezután lehetőség nyílik a következő függvények használatára:

- **double**** beolvas(char* hely);
- **double*** soronkent(char* sor);
- **void** cout(double** matrix);
- **double**** sum(double** M1, double** M2);
- **double**** sub(double** M1, double** M2);
- **double**** mult(double** M1, double** M2);
- **double**** fmem(double** M, int oszlop);
- **double**** almatrix(double** M, int oszlop, int sor);
- **double** det(double** A);
- **void** freem(double** M);
- **void** mulc(double** M, double lambda);
- **double**** kulonsorra(double** M, int sor);
- **double**** transponalt(double** M);
- **void** sorcsere(double** M, int i, int j);

- **double**** adj(double** M);
- **double**** inverse(double** M);
- **void** letisztaz(double** M);
- **double**** deepcpy(double** M);
- **double**** GJE(double** M);
- **void** fkiir(char* hely, double** mit);
- **void** Mmalloc(double** M,int i, int j);

Ha *.txt fájlból olvasunk be mátrixot, annak formai követelményei:
Első sor 2 egész szám, szóközzel elválasztva. Ez írja le a mátrix méretét, első a sorok, második az oszlopok számát. tartalmazza.
Ezután következnek a mátrix elemei. Sorban az új elemet szóközzel, új oszlopot sortöréssel választjuk el. Tizedestörteket '.'-tal jelöljük. A függvények kivétel nélkül olyan pointerre mutató pointerrel operálnak, melyek a mátrix elemein kívül tartalmazzák annak méretét is a képen látható elrendezéshez hasonlóan. Így tehát az indexelés a valódi értékek elérése esetében 'oszlop'=1 és 'sor'=0 -val kezdődik.

3. Fontos tudnivalók

3.1.

A könyvtár '**double**** beolvas(char* hely)' függvényében egy soronkénti 1000 karakteres korlát található. Hosszabb sorral rendelkező mátrix esetén a memóriafoglalás nem lesz megfelelő.

3.2.

A mátrixokat reprezentáló pointerek struktúrája a következő:

- M[0]: két valós értéket tárol, a mátrix dimenzióit (valósként tárolt egész)
- M[0][0]: sorszám, nem beleszámítva a méret tárolására használt sort
- M[0][1]: oszlopszám

Ebből kifolyólag a mátrix valós értékeinek elérése a M[1][0] indexeléssel kezdődik.

$$M[1][0] = 1 \tag{1}$$

Példa: A lenti mátrix esetében a matematikailag (1,1) helyen álló elem elérése.

```
3 3
1 2 3
4 4.5 5
1 2 0
```

1. ábra. Egy 3x3-as mátrix bemenet

3.3.

Tekintve, hogy valós számokkal operálunk, és a számítási kapacitásunk korlátolt, így előfordulhat hogy a számolások során (pl. mátrix*mátrix.inverz) ott, ahol egész értéknek kéne kijönni, egy az adott számtól kicsit, de egyesek számára mégis potenciálisan zavaró mértékben eltérhet. Ekkor ajánlott a
void letisztaz(double** M) függvény használata.