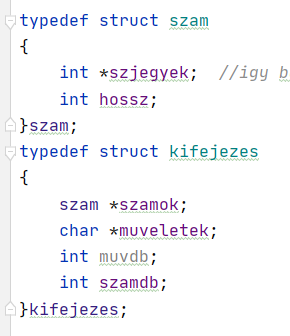
# Egész Aritmetika félkész dokumentáció

## main.c

A program 3 modulból fog állni. A main.c irányítja a programot, összefogja a másik 2 modult. Jelenleg még csak pár ciklus van benne, melyek a beolvasást ellenőrzik, illetve felszabadítják a dinamikusan lefoglalt helyeket.

Az adatok tárolásához ilyen struktúrát találtam ki:

Van egy a számok részére és egy a kifejezéseknek. A számokat dinamikus tömbben fogja a program eltárolni és ez azért hasznos, mert elméletileg bárhány számjegyű számot el lehetne így tárolni és a tömb elemeihez gyorsan lehet hozzáférni, ami a számjegyenkénti számolásoknál nagyon hasznos lesz. Egy kifejezés struktúrája tartalmazza a számok tömbjét, melyben sorban szerepelnek a megkapott számok, a számok darabszámát, ami a számolásoknál lesz hasznos, a műveletek tömbjét, melyben karakterekként szerepelnek az egyes műveletek és a műveletek darabszámát, ami a számolásoknál lesz hasznos.

## file\_muvelet.c

Ez a modul végzi el a file műveleteket, a beolvasást, a beolvasott adatok struktúrába töltését és a kiírást. A kiíró függvény kivételével a másik kettő készen van. A program használati útmutatását a beolvasó függvény írja ki:

Nyomjon egy "e" gombot, majd írja be a kifejezéseket, vagy egy "f" gombot, majd adja meg a beolvasandó file (például: "C:/Users/admin/Desktop/vmi.txt") elérési útját!

Ha Euklideszi algoritmust szeretne végrehajtani, akkor a 2 szám után egy szóközzel elválasztva ezt írja be: "e.a."!

Ha egy lineáris kongruenciát szeretne megoldani, akkor először az x együtthatóját adja meg, majd szóközzel elválasztva azt, hogy mivel legyen kongruens, majd szóközzel elválasztva azt, hogy mi a modulú-ja az egyenletnek, majd szóközzel elválasztva írjon be egy "m" betűt!

Ha befejezte a kifejezések beírását a programablakba, akkor nyomjon egy "x" gombot!

(file-ból csak egyszer lehet beolvasni, de abba bármennyi kifejezést beírhat)

Ezután a megfelelő helyről a függvény beolvassa egy sztringbe a szöveget, majd a kifejezes \*kiftomb(void)függvény eltárolja egy tömbben a beírt kifejezéseket. A kiíró függvény még nincs megírva. Az eredményeket kifejezésenként fogja kiírni.

A file\_muvelet.h tartalmazza a szükséges könyvtárakat, amiknek a függvényeit használják a modulok.

## szamolasok.c

Ebben a modulban lesznek a függvények, amikkel a számolásokat lehet majd megoldani, összeadás, kivonás, szorzás, osztás, az algoritmusok. Az összeadást számjegyenként összeadva, majd, ha az összeg túllépné a számrendszer alapját, akkor a következő számjegyösszeghez 1-et továbbadva. A kivonást szintén számjegyenként elvégezve kell csinálnia a programnak, ha kisebb lenne a különbség 0-nál, akkor a következő helyiértéknél eggyel többet levonva kell majd számolnia. A szorzást is majd számjegyenként kell elvégeznie, hasonlóan az összeadáshoz, ha több lenne a helyiértéki szorzat a számrendszer alapjánál, akkor a megfelelő mennyiséget hozzá kell majd adni a következő szorzathoz. A szorzó minden egyes számjegyével végig kell majd szorozni a szorzandó számot, helyiérték szerint növekvő sorrendben, az egyes rész-szorzatokat pedig a szorzó aktuális helyiértéke szerint kell majd eltolva összeadni (a nagyobb helyiértékkel szorzott számhoz a helyiérték indexével megegyező 0-át kell hozzáadni). Például:

31329 \* 653565437 =

1. lépés: 31329 \* 7 = 219303 (7 a szorzó tényezőben 10^0-on helyiértéken van, a 0 megegyezik a helyiérték indexével)

2. lépés: 31329 \* 3 = 93987 (3 a szorzó tényezőben 10^1-on helyiértéken van, itt az index az 1)

3. lépés: 219303 + 939870 = 1 159 173

4. …