

Programozás alapjai I.- Nagy házi

Rövid leírás: A programom a Programozás alapjai 1. tantárgy nyilvántartásaként szolgál. Tartalmazza az oktatók és hallgatók adatait, a különböző számonkérések eredményeit és csoportbeosztásokat. A nyilvántartás módosítható, így új személyeket lehet létrehozni vagy meglévőket törölni, adataikat/eredményeiket átírni. Ezentúl az adatok lekérhetők, fájlba menthetők. Akár egy-egy hallgató vagy oktató adatai, akár teljes listák: Nagy házi pótleadás, pót ZH lista, továbbá pontszám, osztályzat vagy hallgatók száma szerinti rangsor. A programom egy speciális funkciója, hogy képes megjeleníteni a hallgatók eredményének eloszlását oszlopdiagrammal, grafikus felületen.

Tartalomjegyzék

Tartalomjegyzék	1
1: Feladatválasztás	1
2: Specifikáció:.....	1
A program célja	1
A program használata és a futás eredménye	2
3: Programozói dokumentáció	2
alapok.h	3
main.c	3
fajlmuveletek (.c és .h)	4
hallgmuveletek (.c és .h).....	6
oktmuveletek (.c és .h)	8
szabalyosbemenet (.c és .h)	9
lista (.c és .h).....	10
diagram (.c és .h)	12
egyebek (.c és .h).....	13

1: Feladatválasztás

Prog1 nyilvántartás *

Készíts programot, amely képes hallgatók adatait nyilvántartani, és Prog1 eredményeiket számolni! A megjegyzendő adatok a következők:

- A hallgatók neve, neptun kódja, előadás- és gyakcsoportja.
- Az oktatók neve, csoportjaik.
- Hiányzások, kis ZH eredmények, NZH pontok stb.

„Tanítsd meg” a programnak a tárgykövetelményeket! Legyen képes a program listát készíteni azokról, akiknek PZH-t kell írniuk vagy nagy házi pótlást kell leadniuk! Legyen képes rangsorokat készíteni (pontszámok alapján), és ehhez hasonló listákat kiírni a képernyőre!

2: Specifikáció:

A program célja

A feladat célja egy adatbázis készítése, majd bővítése és szerkesztése, amiben a Programozás alapjai

I. című tárgyat hallgatóknak adatait tartja számon. A program interaktív, a felhasználó hozzáadhat, módosíthat és törölhet adatokat. Az adatokat lekérheti, azok alapján kereshet, vagy megtekintheti a tárgykövetelmény alapján elkészített listákat, statisztikákat.

A program használata és a futás eredménye

Kezelését felhasználóbarát menü segíti. A programfájllal egy mappába vannak elmentve a kezelendő fájlok, melyekkel a felhasználó műveleteket végezhet. A fájlokból információt kereshet ki, új adatokat adhat meg rajta vagy korábbi adatokat módosíthat. Ezeket a program bezárás előtt elmenti, így a fájlok folyamatosan frissülnek.

Funkciók:

- új hallgató adatainak rögzítése
 - adatok: név, Neptun kód, előadás-és gyakorlatkurzus
 - ezeket egy hallgato.txt fájlban tároljuk
- meglévő hallgató adatainak módosítása, hallgató törlése
 - ezek a hallgato.txt fájlban is módosulnak
- új eredmény beírása
 - az eredmény.txt fájlba kerülnek az eredmények, melyek Neptun kód alapján azonosíthatók
- új oktatók és adataik beírása, módosítása
 - adatok: név, csoportjainak száma, csoportnevek (Nem lehet 2 azonos nevű oktató!)
- hallgató adatainak lekérése
 - nevének, pontszámának és jegyének kiírása
 - hallgató adatainak és eredményeinek külön fájlba mentése
 - Megkeresi a csoportjához tartozó oktatót is, ha van (a szöveges fájl neve a hallgató egyedi azonosítója, a Neptun kódja)
- oktató adatainak lekérése
 - csoportjainak kiírása
 - adatainak, és a csoportjába járók Neptun kódjának és nevének fájlba írása
 - statisztika1: gyakorlatcsoport szerinti átlageredmények
 - statisztika2: oktató súlyozott eredménye (hallgatóik átlaga alapján) (a szöveges fájl neve az oktató egyedi azonosítója, a neve)
- rangsor, statisztika megtekintése
 - lista1: legjobb hallgatók kilistázása
 - x db hallgató, x a felhasználó által megadható
 - p pontszám feletti hallgatók, p a felhasználó által megadható
 - lista2: Adott osztályzatú hallgatók kilistázása fájlba
 - lista3: pótZH
 - lista4: pótNHF
 - statisztika3: hallgatók eredményének eloszlása oszlopdiagramon.
- kilépés

Lekérések esetén a bemenet egy szám, módosításkor pedig a megfelelő adatok. A program kimenete történhet a standard kimenetre, szöveges- vagy grafikus (SDL) fájlba.

3: Programozói dokumentáció

Mappa neve: prog1

Tartalma: source és header fájlok, szöveges fájlok

A program csak a latin ABC betűivel működik, így a neveket is ilyen formában képes beolvasni és műveleteket végezni vele. Minden adat perifériáról való bekérése után ENTER-t vár a program. Valamilyen mértékben hibakezelést használ a bekért adatok helyességének vizsgálatára, de a teljesség igénye nélkül, így ésszerű bemenetet vár.

A hallgato.txt valamint az oktato.txt fájlokban egy személy adatai pontosvesszővel vannak elválasztva, a személyek pedig külön sorokba vannak írva. Az eredmény.txt fájlban egy hallgató

eredményei szóközzel vannak elválasztva, és a hallgatók eredményei soronként vannak.

A program futtatásához az SDL könyvtár telepítése szükséges: [InfoC :: Az SDL telepítése \(bme.hu\)](#)

alapok.h

Az összes többi header fájlban include-olva van (kivéve a debugmalloc.h-t)

Tartalmazza a legfontosabb könyvtárakat: stdio.h, stdbool.h, string.h, debugmalloc.h

[debugmalloc.h](#) forrása: <https://infoc.eet.bme.hu/> Dinamikus memória hibakezeléséhez.

Tartalmazza a program fő struktúráit:

Eredmenyek: tárolja egy hallgató eredményeit. Az eredmények.txt egy sorának felel meg, ami maximum 65 karakter hosszúságú lehet.

- előadás, gyakorlat és labor hiányzások: egész számok, 0 és 14 közötti értékekkel
- kis ZH eredmények: 6 elemű egészekből álló tömb, 0 és 10 közötti értékekkel
- Első ZH, második ZH és pótló ZH eredmények, 0 és 40 közötti egész értékekkel
- Első ZH imsc és második zh imsc pont: 0 és 10 közötti egész értékekkel
- Nagy házi követelmény: logikai érték, vagy elfogadva vagy elutasítva
- Nagy házi pontszám: egész szám 0 és 20 közötti értékkel
- Szorgalmi pontszám: egész szám 0 és 15 közötti értékkel
- Követelmény: logikai változó, vagy teljesítve vagy megtagadva
- Pontszám és jegy egész típusú változók.

A követelmény, pontszám és jegy az előző eredmények alapján kerül kiszámításra az erdszmit() függvény által.

Hallgato: tárolja egy hallgató adatait, az első négy adata a hallgato.txt egy sorának felel meg, ami maximum 70 karakter hosszúságú lehet.

- Neptun kód: szöveg, a hallgató azonosítója, nem egyezhet másik hallgatóéval. Hat karakter, csak nagybetűkből és számokból állhat
- Név: szöveg, maximum 50 karakter lehet, a kis és nagybetűkön kívül a szóköz, kötőjel és a pont karakterek engedélyezettek benne
- Előadás és gyakorlatcsoport: szöveg, maximum 5 karakter lehet, a kis és nagybetűkön kívül a szóköz, kötőjel, pont és aláhúzásjel engedélyezett benne
- Eredmenyek struktúra a hallgató eredményeinek
- Hallgato struktúrára mutató pointer a láncolt listához

Oktato: tárolja egy oktató adatait. Utolsó adata kivételével az oktato.txt fájl egy sorának felel meg, ami maximum 83 karakter hosszúságú lehet

- Név: szöveg, maximum 50 karakter lehet, a kis és nagybetűkön kívül a szóköz, kötőjel és a pont karakterek engedélyezettek benne. Ez az oktató egyéni azonosítója, így a nyilvántartásban nem szerepelhet két azonos nevű oktató
- Mennyi: egész szám 1 és 5 között, az oktató csoportjainak száma
- Csop: Ötelemű szövegeket tartalmazó tömb. 'mennyi'-vel megegyező számú csoportnév van benne. Egy csoport maximum 5 karakter lehet, a kis és nagybetűkön kívül a szóköz, kötőjel, pont és aláhúzásjel engedélyezett benne
- Oktato struktúrára mutató pointer a láncolt listához

main.c

Include: alapok.h, egyebek.h, hallgmuveletek.h, oktmuveletek.h, fajlmuveletek.h, lista.h, diagram.h

Az infoc() függvénnyel nyitókép kirajzolása.

beolvashallg() függvénnyel a hallgato.txt-t beolvassuk ezzel létrehozva a hallgatók listáját

beolvasokt() függvénnyel az oktato.txt-t beolvassuk ezzel létrehozva az oktatók listáját

beolvaseredmeny() függvénnyel az eredmény.txt-t beolvassuk a hallgatók listájába

Menü: felsorolja a felhasználónak a lehetséges műveleteket, a megfelelő bemenettel. A bemenetet beolvasva átirányítja a megfelelő függvényhez, elvégzi az adott műveletet majd visszatér a menühöz, egészen a kilépés opció választásáig.

1. új hallgató létrehozása
 - ujhallg() függvény hívása
2. meglévő hallgató adatainak módosítása, hallgató törlése, eredmény beírása
 - modhallg() függvény hívása
3. hallgató adatainak és eredményének fájlba mentése
 - lekerhallg() függvény hívása
4. új oktató létrehozása
 - ujokt() függvény hívása
5. meglévő oktató adataik módosítása, oktató törlése
 - modokt() függvény hívása
6. oktató adatainak fájlba mentése
 - lekerokt() függvény hívása
7. Lista megtekintése
 - lista() függvény hívása
8. Statisztika megtekintése
 - stat() függvény hívása
0. kilépés
 - bagoly() függvény hívása a zárókép kirajzolásához
 - A menü ciklusából való kilépés engedélyezése

Default – menü újra, érvénytelen bemenet

Hallgató lista fájlba mentése (hallgato.txt és eredmeny.txt) a menteshallg() függvénnyel.

Oktató lista fájlba mentése (oktato.txt) a mentesokt() függvénnyel.

Program vége.

A továbbiakban minden .c fájlban a megegyező nevű .h fájl van csak include-olva. A .h fájlokban az alapok.h és az adott függvényekhez szükséges további könyvtárak, melyeket részletezek majd.

fajlmuveletek (.c és .h)

.h-ban további könyvtárak: stdlib.h, .c-ben: szabalyosbemenet.h, hallgmuveletek.h, oktmuveletek.h

A dinamikus memóriafoglálások és fájlok megnyitása során hibakezeléssel ellenőrzöm a sikerességet.

beolvashallg()

Feladata: a hallgato.txt fájl beolvasása a hallgatók listájába

Paraméterei: nincs

Visszatérési értéke: a lista első elemére mutató pointer

Működése: megnyitja olvasásra a hallgato.txt fájlt, amiből a fájl végéig sorokat olvas be.

Ezeket a választó karakter szerint osztja fel és az adatokat elmenti a dinamikusan foglalt hallgatók listájába. Az összes hallgató listába mentése után bezárja a szöveges fájlt és visszatér a lista első elemére mutató pointerrel.

beolvasokt()

Feladata: az oktato.txt fájl beolvasása az oktatók listájába

Paraméterei: nincs

Visszatérési értéke: a lista első elemére mutató pointer

Működése: megnyitja olvasásra az oktato.txt fájlt, amiből a fájl végéig sorokat olvas be.

Ezeket a választó karakter szerint osztja fel és az adatokat elmenti a dinamikusan foglalt oktatók listájába. Az összes oktató listába mentése után bezárja a szöveges fájlt és visszatér a lista első elemére mutató pointerrel.

beolvaseredmeny()

Feladata: az eredmeny.txt fájl beolvasása a hallgatók listájába

Paraméterei: a hallgatók listájának első elemére mutató pointer

Visszatérési értéke: logikai érték a beolvasás sikerességéről

Működése: megnyitja olvasásra az eredmény.txt fájlt, amiből a fájl végéig sorokat olvas be. Ezeket a választó karakter szerint osztja fel és az adatokat elmenti a már elmentett hallgatókhoz. A megfelelő hallgatót a program a Neptun kód alapján találja meg.

menteshallg()

Feladata: hallgatók listájának hallgato.txt és eredmény.txt fájlba mentése,
dinamikusan foglalt terület felszabadítása

Paraméterei: hallgatók listájának első elemére mutató pointer

Visszatérési értéke: nincs

Működése: megnyitja írásra a hallgato.txt és eredmény.txt fájlokat, amibe az elválasztó karakterekkel megfelelően visszaírja az adatokat, így a program későbbi használatakor mindig a frissített verzió használható. A hallgató adatainak elmentésével felszabadítja a dinamikusan foglalt területet. Ha minden hallgatóval végzett bezárja a fájlokat.

mentesokt()

Feladata: oktatók listájának oktato.txt fájlba mentése, din. foglalt terület felszabadítása

Paraméterei: oktatók listájának első elemére mutató pointer

Visszatérési értéke: nincs

Működése: megnyitja írásra az oktato.txt fájlt, amibe az elválasztó karakterekkel megfelelően visszaírja az adatokat, így a program későbbi használatakor mindig a frissített verzió használható. Az oktató adatainak elmentésével felszabadítja a dinamikusan foglalt területet. Ha minden oktatóval végzett bezárja a fájlt.

lekerhallg()

Feladata: hallgató adatainak fájlba mentése, a fájl neve a hallgató neptun kódja

Paraméterei: hallgatók listájának első elemére mutató pointer,
oktatók listájának első elemére mutató pointer

Visszatérési értéke: nincs

Működése: Beolvassa a menteni kívánt hallgatót a kodbe() függvénnyel, majd a kereshallg() függvénnyel megkeresi a hallgatót. A függvény csak akkor folytatódik, ha a hallgató szerepel a nyilvántartásban. Ha megtalálta, a konzolos felületre kiírja a nevét, pontszámát és jegyét. Továbbá megnyitja a hallgató Neptun kódjával megegyező szöveges fájlt írásra. Ebbe kiírja a nevét és Neptun kódját, valamint az előadás és gyakorlatcsoportját, amihez az oktatocsopszerint() függvénnyel megkeresi a csoportot oktató személy nevét, ha megtalálta beleírja ezt is az adott csoporthoz, ha nincs, akkor pedig azt hogy nincs bejegyezve még a csoport oktatója a nyilvántartásba. Ezután kilistázza a fájlba a hallgató eredményeit is, majd bezárja a fájlt.

lekerokt()

Feladata: oktató adatainak fájlba mentése, a fájl neve az oktató neve

Oktató csoportjába járók kilistázása, csoportok átlag eredményei, oktató csoportjainak összesített átlageredménye

Paraméterei: oktatók listájának első elemére mutató pointer,
hallgatók listájának első elemére mutató pointer

Visszatérési értéke: nincs

Működése: Beolvassa a menteni kívánt oktatót a nevbe() függvénnyel, majd a keresokt() függvénnyel megkeresi az oktatót a listában. A függvény csak akkor folytatódik, ha az oktató szerepel a nyilvántartásban. Ha megtalálta, a konzolos felületre kiírja a nevét és csoportjait. Továbbá megnyitja az oktató nevével megegyező szöveges fájlt írásra. Ebbe kiírja a nevét és a csoportjait. A csoportokhoz kilistázza az oda járó hallgatók Neptun kódját és nevét is, és közben folyamatosan számolja a csoport hallgatóinak létszámát, összpontszámát és összjegyét. Ha egy csoport végére ér kiírja az adott csoport átlageredményét (pontszám és jegy), valamint hozzáadja a csoport létszámát,

összpontszámát és összegjét az oktató összes csoportjának létszámához, összpontszámához és összegjéhez. Ha az összes csoport végére ért kiírja az oktató összes hallgatójának átlageredményét, így egyfajta súlyozott eredményét az oktátónak, majd bezárja a fájlt. (Azért súlyozott eredmény, mert ha egy hallgatónak ő az előadója és a gyakorlatvezetője is, akkor a teljes átlagba a hallgató eredményét kétszer számolja, kétszer annyira függ az oktatótól a hallgató eredménye – ezt persze fenntartásokkal kezelve, mert leginkább a hallgatótól függ a saját eredménye.)

hallgmuveletek (.c és .h)

.h-ban további könyvtárak: stdlib.h, .c-ben: szabalyosbemenet.h, egyebek.h

A dinamikus memóiafogalások és fájlok megnyitása során hibakezeléssel ellenőrzöm a sikerességet.

kereshallg()

Feladata: listában hallgató keresése Neptun kód alapján

Paraméterei: hallgatók listájának első elemére mutató pointer,
keresett Neptun kód

Visszatérési értéke: ha megtalálja akkor a hallgatóra mutató pointer,
ha nincs bejegyezve NULL pointer

Működése: a hallgatók listáján fut végig és ha a keresett Neptun kód megegyezik a hallgató Neptun kódjával akkor visszatér a hallgatóra mutató pointerrel. Ha találat nélkül végigmegy a listán akkor NULL pointerrel tér vissza.

erdszmit()

Feladata: hallgató végleges eredményének kiszámítása

Paraméterei: hallgatóra mutató pointer

Visszatérési értéke: nincs

Működése: A hallgató adataival rendelkezve kiszámolja végleges eredményeit.

Aláírás/követelmény elutasítva (a következők között logikai VAGY kapcsolat van!):

Ha több mint 4 hiányzása van előadásról, gyakorlatról vagy laborról

Ha a kis ZH-kon nem ért el legalább 20 pontot (a legjobb 4 számít bele)

Ha a ZH eredménye kevesebb mint 40 pont

Ha a második ZH kevesebb mint 20 pont és nem írt még pót ZH-t

Ha a NHF-t nem fogadták el vagy nincs legalább 10 pont

A végleges pontszám:

Legjobb 4 kis ZH eredmény+ZH eredmény+NHF eredmény+Szorgalmik+

ZH eredmény: Ha a 2.ZH jobb lett, mint az 1. akkor a 2. pontszáma duplázódik.

Ha pót ZH-t írt akkor mindenképp azt duplázza.

Egyébként a két ZH összege + ha 75%-ot legalább elért akkor hozzáadódik az adott ZH-n szerzett iMSc pont is.

A végleges jegy:

5: 125- pont

4: 110-124 pont

3: 90-109 pont

2: 70-89 pont

1: -69 pont vagy Aláírás elutasítva

ujhallg()

Feladata: új hallgató adatainak beszúrása a lista elejére

Paraméterei: hallgatók listájának első elemére mutató pointer

Visszatérési értéke: hallgatók listájának első elemére mutató pointer

Működése: Bekéri a létrehozandó hallgató Neptun kódját a kodbe() függvénnyel, majd a kereshallg() függvénnyel megkeresi a hallgatót a már létező listában. A függvény csak akkor

folytatódik, ha a hallgatót nem találta meg a nyilvántartásban. Ekkor dinamikusan foglal helyet az új hallgatónak és egyesével bekéri a nevét, előadás-és gyakorlatcsoportját. A nevet a nevbe() függvénnyel kéri be ahol ellenőrzi is a szabályos bemenetet, a csoportnevek helyességét pedig a szabalyos_gyakvagyea() függvény ellenőrzi. Az eredmény_inicializalas() függvénnyel inicializálja a hallgató eredményeit (kinullázza). Visszatér az új hallgatóra mutató pointerrel, mint a hallgatólista új kezdetével.

torleshallg()

Feladata: listában szereplő hallgató törlése

Paraméterei: hallgatók listájának első elemére mutató pointer,
törlendő hallgató Neptun kódja

Visszatérési értéke: hallgatók listájának első elemére mutató pointer

Működése: lemaradó és mozgó pointer segítségével végigfut a hallgatók listáján a megfelelő Neptun kódot keresve. Ha megtalálja, akkor a lemaradó pointer segítségével az előtte levő hallgató kov pointerét átállítja a törlendő hallgatót követő hallgató pointerére és felszabadítja a törlendő hallgató dinamikusan foglalt területét. Külön esetként kell kezelni, ha a lista első hallgatóját kell törölni, ekkor ideiglenesen foglalt pointerrel elmentjük a törlendő hallgatót, a hallgatók listájára mutató pointert pedig átállítjuk a következő hallgatóra. Viszont mivel elmentettük a törlendő hallgatóra mutató pointert, így fel tudjuk szabadítani a memóriáját. Ebben az esetben is az első hallgatóra mutató pointerrel tér vissza, mint egyébként, csak ilyenkor az meg is változott.

adatmodhallg()

Feladata: listában szereplő hallgató adatainak módosítása

Paraméterei: hallgatók listájának első elemére mutató pointer,
aktuális hallgatóra mutató pointer

Visszatérési értéke: nincs

Működése: Végigmegy egyesével a hallgató adatain és bekéri a felhasználótól, hogy szeretne-e módosítani rajta. Ha igen akkor I vagy i karaktert ad a bemenetre, ha nem akkor N-t (de igazából bármi más esetén is ez lesz az eredmény). Ha meg szeretné változtatni akkor az új hallgató létrehozásához hasonlóan jár el: beolvassa az új adatot és megvizsgálja helyességét. Neptun kódnál külön megnézi, hogy ne létezzen más azonos kódú hallgató.

eredmenymodhallg()

Feladata: listában szereplő hallgató eredményének módosítása

Paraméterei: aktuális hallgatóra mutató pointer

Visszatérési értéke: nincs

Működése: Az adatmodhallg() függvényhez hasonló működésű. Egyesével kérdezi végig a számontartott eredményekről, hogy módosításra kerüljenek vagy sem. A felhasználó azt is láthatja, hogy mi volt az eddigi eredmény. A bemenet ismét I(i)/N (bármi más), majd bekéri az új eredményt, vizsgálva, hogy az valós eredmény legyen (például nem lehet egy kis ZH eredmény 10 pontnál több, mivel mindegyik eredmény más feltételt kell vizsgálni így külön részekkel vannak kezelve, nincs egységesítve, így dinamikusabb lehet a program). Mivel a felhasználó az eredményeken változtatott így a végleges eredmények is változhattak, ezért ezt az erdszamt() függvénnyel frissítjük.

modhallg()

Feladata: Hallgató módosítása/törlése/eredmény megadása menüopció utáni almenü

Paraméterei: hallgatók listájának első elemére mutató pointer

Visszatérési értéke: hallgatók listájának első elemére mutató pointer

Működése: a kodbe() és kereshallg() függvényekkel bekéri és megkeresi a kezelendő hallgatót a listában. Ha megtalálja akkor folytatódik a függvény egy almenüvel, ami 4 opciót tár a felhasználó elé, így azokból tud a 0-3 számokkal választani. 0: Visszalépés a főmenübe. 1: hallgató törlése torleshallg() függvénnyel. 2: adatok módosítása adatmodhallg() függvénnyel. 3: eredmények

módosítása eredmennyomdallg() függvénnyel. Más bemenet esetén újra következik az almenü.

oktmuveletek (.c és .h)

Header fájljában további könyvtárak: stdlib.h, .c-ben: szabalyosbemenet.h

A dinamikus memóiafoglalások és fájlok megnyitása során hibakezeléssel ellenőrzöm a sikerességet.

keresokt()

Feladata: listában oktató keresése név alapján

Paraméterei: oktatók listájának első elemére mutató pointer,
keresett oktató neve

Visszatérési értéke: ha megtalálja akkor a hallgatóra mutató pointer,
ha nincs bejegyezve NULL pointer

Működése: az oktatók listáján fut végig és ha a keresett név megegyezik az oktató nevével akkor visszatér az oktatóra mutató pointerrel. Ha találat nélkül végigmegy a listán akkor NULL pointerrel tér vissza.

oktatocsoapszerint()

Feladata: listában oktató keresése csoport alapján

Paraméterei: oktatók listájának első elemére mutató pointer,
keresett csoport neve

Visszatérési értéke: ha megtalálja akkor a hallgatóra mutató pointer,
ha nincs bejegyezve NULL pointer

Működése: az oktatók listáján fut végig és ha a keresett csoport megegyezik az aktuális oktató valamelyik csoportjával akkor visszatér az oktatóra mutató pointerrel. Ha találat nélkül végigmegy a listán akkor NULL pointerrel tér vissza.

ujokt()

Feladata: új oktató létrehozása a lista elejére

Paraméterei: oktatók listájának első elemére mutató pointer

Visszatérési értéke: oktatók listájának első elemére mutató pointer

Működése: Bekéri a létrehozandó oktató nevét a nevbe() függvénnyel, majd a keresokt() függvénnyel megkeresi az oktatót a már létező listában. A függvény csak akkor folytatódik, ha az oktatót nem találta meg a nyilvántartásban. Ekkor dinamikusan foglal helyet az új oktátónak és bekéri a csoportjainak számát és a csoportok nevét. Bekéréskor ellenőrzi is a szabályos bemenetet a „mennyi”-nél 1 és 5 közötti szám, a csoportneveket pedig a szabalyos_gyakvagyea() (helyes bemenet) és a mascsoportja() (nem oktatja még más a csoportot) függvények ellenőrzik. Visszatér az új oktatóra mutató pointerrel, mint az oktatólista új kezdetével.

torlesokt()

Feladata: oktató törlése a listából

Paraméterei: oktatók listájának első elemére mutató pointer,
törlendő oktató neve

Visszatérési értéke: oktatók listájának első elemére mutató pointer

Működése: lemaradó és mozgó pointer segítségével végigfut az oktatók listáján a megfelelő nevet keresve. Ha megtalálja, akkor a lemaradó pointer segítségével az előtte levő oktató kov pointerét átállítja a törlendő oktatót követő oktató pointerére és felszabadítja a törlendő oktató dinamikusan foglalt területét. Külön esetként kell kezelni, ha a lista első oktatóját kell törölni, ekkor ideiglenesen foglalt pointerrel elmentjük a törlendő oktatót, az oktatók listájára mutató pointert pedig átállítjuk a következő oktatóra. Viszont mivel elmentettük a törlendő oktatóra mutató pointert, így fel tudjuk szabadítani a memóriáját. Ebben az esetben is az első oktatóra mutató pointerrel tér vissza, mint egyébként, csak ilyenkor az meg is változott.

adatmodokt()

Feladata: listában szereplő oktató adatainak módosítása

Paraméterei: oktatók listájának első elemére mutató pointer,
aktuális oktatóra mutató pointer

Visszatérési értéke: nincs

Működése: Végigmegy egyesével az oktató adatain és bekéri a felhasználótól, hogy szeretne-e módosítani rajta. Ha igen akkor I vagy i karaktert ad a bemenetre, ha nem akkor N-t (de igazából bármi más esetén is ez lesz az eredmény). Ha meg szeretné változtatni akkor az új oktató létrehozásához hasonlóan jár el: beolvassa az új adatot és megvizsgálja helyességét. Névnél külön megnézi, hogy ne létezzen más azonos nevű oktató. Megjegyzés: a mascsoportja() függvény alkalmazásakor a saját magához korábban beírt csoportot is vizsgálja még. (Példa: a korábbi 4. csoportját módosításkor nem lehet 2.nak írni, mert az még tartozik egy oktatóhoz, mégpedig önmagához a 4. csoportként.)

modokt()

Feladata: oktató módosítása/törlése menüopció utáni almenü

Paraméterei: oktatók listájának első elemére mutató pointer

Visszatérési értéke: oktatók listájának első elemére mutató pointer

Működése: a nevbe() és keresokt() függvényekkel bekéri és megkeresi a kezelendő oktatót a listában. Ha megtalálja akkor folytatódik a függvény egy almenüvel, ami 3 opciót tár a felhasználó elé, így azokból tud a 0-2 számokkal választani. 0: Visszalépés a főmenübe. 1: oktató törlése torlesokt() függvénnyel. 2: adatok módosítása adatmodokt() függvénnyel. Más bemenet esetén újra következik az almenü.

szabalyosbemenet (.c és .h)

Header fájljában további könyvtárak: ctype.h

szabalyos_neptun()

Feladata: Neptun kód szabályosságának vizsgálata

Paraméterei: Neptun kód

Visszatérési értéke: logikai változó, szabályos-e vagy sem

Működése: Ha 6 karakterből áll a kód, amik latin ABC nagybetűi vagy 0-9 számok, akkor igaz értéket ad vissza, egyébként hamisat.

szabalyos_nev()

Feladata: Név szabályosságának vizsgálata

Paraméterei: név

Visszatérési értéke: logikai változó, szabályos-e vagy sem

Működése: Ha nem több mint 50 karakter a név, amik latin ABC nagy-vagy kisbetűi vagy pont (.), kötőjel (-) vagy szóköz, akkor igaz értéket ad vissza, egyébként hamisat.

szabalyos_gyakvagyea()

Feladata: Csoportkód szabályosságának vizsgálata

Paraméterei: csoportnév

Visszatérési értéke: logikai változó, szabályos-e vagy sem

Működése: Ha nem több mint 5 karakter, amik latin ABC nagy-vagy kisbetűi, számok vagy pont (.), kötőjel (-), alsó kötőjel (_) vagy szóköz, akkor igaz értéket ad vissza, egyébként hamisat.

kodbe()

Feladata: Neptun kód beolvasása

Paraméterei: Neptun kód
Visszatérési értéke: nincs
Működése: addig olvassa be a Neptun kódot, amíg a szabalyos_neptun() függvény igazzal nem tér vissza.

nevbe()

Feladata: Név beolvasása
Paraméterei: név
Visszatérési értéke: nincs
Működése: addig olvassa be a nevet, amíg a szabalyos_nev() függvény igazzal nem tér vissza.

mascsoportja()

Feladata: az oktató csoportjainak beolvasásakor ellenőrzi, hogy nincs-e már más oktatóhoz beírva a csoport
Paraméterei: oktatók listájának első elemére mutató pointer, keresett csoportnév
Visszatérési értéke: logikai érték, igaz=máshoz tartozik a csoport, hamis=senkié
Működése: végigmegy az oktatók listáján végig nézve minden oktató csoportját és ha valamelyiknél megtalálja a csoport nevét igaz értékkel tér vissza, ha nem akkor hamissal.

lista (.c és .h)

Header fájljában külön struktúra adott hallgatók listájához és könyvtár: stdlib.h

A dinamikus memóriakezelés sikerességét és fájlok megnyitását ellenőrzöm.

ListHallg: bizonyos hallgatók bizonyos adatait tárolja

- Neptun kód: szöveg, a hallgató azonosítója, nem egyezhet másik hallgatóéval. Hat karakter, csak nagybetűkből és számokból állhat
- Pont: egész szám a hallgató adatai alapján
- Jegy: egész szám a hallgató adatai alapján
- ListHallg kov struktúrára mutató pointer a láncolt listához

sorrendbeszur()

Feladata: Új, rendezett listába fűzes (Neptun kód vagy pontszám szerint)
Paraméterei: lista eleje pointer, aktuális hallgató pointere, mi: egy egész szám, ami befolyásolja a rendezés típusát. Neptun kód szerinti, ha mi=1, pontszám szerinti, ha mi=0.
Visszatérési értéke: lista első elemére mutató pointer
Működése: az aktuális hallgatónak dinamikus memóriát foglal, majd átmásolja a hallgató adatait. Ezt később a program során még fel kell szabadítani. Lemaradó és mozgó pointer segítségével végigmegy a listán egészen addig amíg a rendezési szempont szerint szükséges. Ha Neptun szerinti, akkor megáll, ha az aktuális hallgató neve már az újnak a neve után következik. Ha pontszám szerinti akkor megáll, ha az aktuális hallgató pontja kisebb, mint az új hallgatóé. Ekkor a lemaradó és a mozgó pointer segítségével beilleszti a listába a kettő közé az új hallgatót, így a listát rendezetten építi fel.

tophallg()

Feladata: Adott létszámú legjobb hallgató kiírása pontszám szerint
Paraméterei: hallgatólista elejére mutató pointer, a létszám egészben megadva
Visszatérési értéke: Nincs
Működése: Végigmegy a hallgatólistán és a sorrendbeszur() függvény segítségével minden hallgató hozzáadásával létrehozza a hallgatók pontszám szerinti rendezett listáját. Ezután végigmegy a lista összes elemén számlálással, ami amíg kisebb, mint a megadott létszám, addig kiírja a hallgatók helyezését, Neptun kódját és pontszámát. Emellett a lista bejárása során minden dinamikusan foglalt listaelemet felszabadít.

Megjegyzés: ha a nyilvántartásban n db hallgató van maximum n-1 hallgató kérhető le.

toppont()

Feladata: Adott pontszám feletti hallgatók kiírása névsor szerint

Paraméterei: hallgatólista elejére mutató pointer, a pontszám egészben megadva

Visszatérési értéke: Nincs

Működése: Végigmegy a hallgatólistán és a sorrendbeszur() függvény segítségével a hallgató hozzáadásával létrehozza a hallgatók névsor szerinti rendezett listáját. Feltétellel vizsgálja, hogy a hallgató nagyobb pontszámmal rendelkezik-e, mint a megadott pontszám, így csak a megfelelő hallgatók kerülnek bele az új listába. Ezután végigmegy a lista összes elemén és kiírja a hallgatók Neptun kódját és pontszámát, majd felszabadítja a dinamikusan foglalt listaelemeket.

osztalyzat()

Feladata: Adott osztályzattal rendelkezők fájlba írása pontszám szerinti sorban

Paraméterei: hallgatólista elejére mutató pointer, a jegy egészben megadva

Visszatérési értéke: Nincs

Működése: A jegy függvényében megnyitja a jegynek megfelelő nevű szöveges fájlt írásra (1-elegtelen, 2-elegseges, 3-kozepes, 4-jo, 5-jeles). Végigmegy a hallgatólistán és a sorrendbeszur() függvény segítségével a hallgató hozzáadásával létrehozza a hallgatók pontszám szerinti rendezett listáját. Feltétellel vizsgálja, hogy a hallgató azzal a jeggyel rendelkezik-e, mint a megadott jegy, így csak a megfelelő hallgatók kerülnek bele az új listába. Ezután végigmegy a lista összes elemén és kiírja a hallgatók Neptun kódját és pontszámát a fájlba, majd felszabadítja a dinamikusan foglalt listaelemeket. A bejárás és felszabadítás után bezárja a fájlt.

potzhlista()

Feladata: Pót ZH megírására kötelezettek listája

Paraméterei: hallgatólista elejére mutató pointer

Visszatérési értéke: Nincs

Működése: Végigmegy a hallgatólistán és a sorrendbeszur() függvény segítségével a hallgató hozzáadásával létrehozza a hallgatók névsor szerinti rendezett listáját. Minden hallgatónál kiszámolja a ZH pontszámot: Ha a 2.ZH jobb lett, mint az 1. akkor a 2. pontszáma duplázódik, egyébként a kettő összeadódik. Ha az így keletkezett pontszám kisebb, mint 40 vagy a 2.ZH kevesebb mint 20 pont, és nem írt még pót ZH-t, akkor a feltétel teljesül, csak ekkor adja hozzá az új listához a hallgatót. Ezután végigmegy a lista összes elemén és kiírja a hallgatók Neptun kódját, majd felszabadítja a dinamikusan foglalt listaelemeket.

pothazilista()

Feladata: Pót házi beadására kötelezettek listája

Paraméterei: hallgatólista elejére mutató pointer

Visszatérési értéke: Nincs

Működése: Végigmegy a hallgatólistán és a sorrendbeszur() függvény segítségével a hallgató hozzáadásával létrehozza a hallgatók névsor szerinti rendezett listáját. Ha a hallgató nagy házi kevesebb, mint 10 pont vagy a követelmény nincs elfogadva, akkor a feltétel teljesül, csak ekkor adja hozzá az új listához a hallgatót. Ezután végigmegy a lista összes elemén és kiírja a hallgatók Neptun kódját, majd felszabadítja a dinamikusan foglalt listaelemeket.

lista()

Feladata: Listák almenüje

Paraméterei: hallgatólista elejére mutató pointer

Visszatérési értéke: Nincs

Működése: A felhasználó egy almenü 5 opciója közül választhat, 0-4 bemenetekkel.

0: Visszalépés a menübe.

1: Pontszám szerinti rangsor, ekkor egy újabb almenü következik 3 lehetőséggel, 0-2. bemenetekkel.

0: Visszalépés a menübe.

1: Hallgatók száma szerint, ekkor bekér egy számot (ellenőrzi) a felhasználótól, ami meghatározza a tophallg() függvény létszám paraméterét, mely ezután lefut.

2: Adott pontszám felettiek listázása, ekkor szintén egy számot kér be (ellenőrzi), ami a lefutandó toppont() függvény pontszám paraméterét határozza meg.

Más bemenet esetén újrakérdezi az almenüt.

2: Osztályzat szerinti kilistázás fájlba. Bekéri az adott jegyet (ellenőrzi), aminek segítségével lefuttatja az osztalyzat() függvényt.

3: Pót ZH lista: potzhlista() függvényt hívja meg.

4: Nagy házi pótlás listája: pothazilista() függvényt hívja meg.

Más bemenet esetén újakezdi az almenüt.

diagram (.c és .h)

Mivel SDL megjelenítés történik benne, így a program is az InfoC SDL2 típusú projectjeként lett létrehozva Code::Blocks-ban. A diagram.h tartalmazza az alábbi könyvtárakat: SDL.h, SDL2_gfxPrimitives.h, math.h, stdlib.h

sdl_init ()

Feladata: SDL ablak létrehozása

Paraméterei: név, szélesség, magasság, ablak, renderer

Visszatérési értéke: nincs

Forrás: InfoC honlapról (az ilyen típusú projekt létrehozásakor automatikusan legenerálódik mint a Hello World)

diagram()

Feladata: Diagram adatainak kiszámolása, oszlopok rajzolása adatoknak megfelelően

Paraméterei: hallgatólista elejére mutató pointer

Visszatérési értéke: nincs

Működése: gfxPrimitivesSetFont() függvény használatával, ha a program egy futásideje alatt többször lesz meghívva a függvény, akkor többedjére is jól jeleníti meg az adatokat (egyébként a stringRGBA-kat nem rajzolná ki). sdl_init() függvénnyel ablak létrehozása, majd a hallgatólista bejárása. 6 lehetséges jegy lehet: 0-6, ezeket egy 6 elemű tömbben számoljuk a bejárás során a hallgatók eredményének megfelelően.

Oszlopok magasságának kiszámítása (mégegy 6 elemű tömbbe kiszámítva az oszlop felső csúcsának koordinátáit) A számítás hátterében: Ha minden hallgatónak megegyezne a jegye, akkor az az oszlop y1=50 y2=375 koordinátákkal rendelkezzen. Tehát a maximális oszlopméret 325 képpont (a 6 oszlop magasságának összege is mindig ennyi lesz, kivéve ha mind 0). Az oszlop alja mindig fix, y2=375. A változó oszlopmagassághoz az y1=50 koordinátát kell változtatni. Adott az aktuális darabszám db[i] es az összes hallgató db száma ossz.

Ha a kettő hányadosát nézzük db/ossz, akkor lesz 1 ha mindenki azzal az eredménnyel rendelkezik. Ekkor 50 képpontnak kell lennie, így az aktuális kezdő képpont y1 koordinátája: pix[i]=50.

Ha 0 fő rendelkezik a jeggyel pix[i]=375.

Általánosan megadható képlet: $\text{pix}[i] = 375 - (\text{db}[i] / \text{ossz}) * 325$

Plusz példa: 50 hallgatóból 10-en kaptak 4-est. (Ez a hallgatók 20%-a)

$\text{pix}[4] = 375 - (10/50) * 325 = 375 - 65 = 310$ (A magassága $375 - 310 = 65$ lesz ami a 325 max magasságnak tényleg 20%-a)

Oszlopok kirajzolása szöveggel: milyen jegy és hány db, tengelyek kirajzolása szöveggel. Az y tengelyen be van jelölve és mellé írva a maximális létszám.

SDL_RenderPresent() függvénnyel kirajzoljuk az elvégzett műveleteket a képernyőre.

Kilépés az ablakból x-re kattintással vagy egér kattintásával.

egyebek (.c és .h)

infoc()

Feladata: nyitókép kirajzolása

Paraméterei: nincs

Visszatérési értéke: nincs

bagoly()

Feladata: kilépés előtti rajz kiírása

Paraméterei: nincs

Visszatérési értéke: nincs

eredmeny_inicializalas()

Feladata: új hallgató létrehozásakor eredmény inicializálása nullákkal

Paraméterei: hallgatóra mutató pointer

Visszatérési értéke: nincs

Program futtatásáról rövid videó:

<https://drive.google.com/file/d/1kX1vpOY84gMYj-bl3hDjwmzgQc659nmN/view?usp=sharing>