https://sjeprog.blogspot.com/p/20.html A struktúra több tetszőleges típusú adat együttese. struct struktúra_azonosító { típus1 név1; típus2 név2; }; #include <stdio.h> In [1]: struct diak { char nev[30]; int szuletesiEv; double atlag; }; int main() { struct diak d; printf("Kérem a diák nevét: "); scanf("%s", &d.nev); printf("Kérem a diák születési évét: "); scanf("%d", &d.szuletesiEv); printf("Kérem a diák jegyeinek átlagát: "); scanf("%lf", &d.atlag); printf("\n"); printf("%s, %d, %.2lf\n", d.nev, d.szuletesiEv, d.atlag); } Kérem a diák nevét: Kérem a diák születési évét: Kérem a diák jegyeinek átlagát: Ferenc, 1985, 4.10 Dinamikus memóriafoglalás struktúra számára A memóriát a struktúra számára dinamikusan is lefoglalhatjuk. Ebben az esetben kétféleképpen érhetjük el a struktúra adattagjait: pont operátorral, pl. (*d).nev • nyíl operátorral, pl. d->nev In []: #include <stdio.h> #include <stdlib.h> struct diak { char nev[30]; int szuletesiEv; double atlag; }; int main() { struct diak *d; d = (struct diak*)malloc(sizeof(struct diak)); printf("Kerem a diak nevet: "); scanf("%s", &d->nev); printf("Kerem a diak szuletesi evet: "); scanf("%d", &d->szuletesiEv); printf("Kerem a diak jegyeinek atlagat: "); scanf("%lf", &d->atlag); printf("\n"); printf("%s, %d, %.2lf\n", d->nev, d->szuletesiEv, d->atlag); printf("%s, %d, %.2lf\n", (*d).nev, (*d).szuletesiEv, (*d).atlag); printf("\n"); free(d); return 0; } Struktúratömb és dinamikus struktúratömb In []: #include <stdio.h> typedef struct diak { char nev[30]; int szuletesiEv; double atlag; } Diak; int main() { Diak d[3]; for (int i = 0; i < 3; i++) { printf("Kerem a diak nevet: "); scanf("%s", &d[i].nev); printf("Kerem a diak szuletesi evet: "); scanf("%d", &d[i].szuletesiEv); printf("Kerem a diak jegyeinek atlagat: "); scanf("%lf", &d[i].atlag); printf("\n"); for (int i = 0; i < 3; i++) { printf("%s, %d, %.2lf\n", d[i].nev, d[i].szuletesiEv, d[i].atlag); printf("\n"); return 0; } Dinamikus helyfoglalással In []: #include <stdio.h> #include <stdlib.h> typedef struct diak { char nev[30]; int szuletesiEv; double atlag; } Diak; int main() { Diak *d; int drb; printf("Kerem a diakok szamat: "); scanf("%d", &drb); printf("\n"); d = (Diak*)malloc(drb*sizeof(Diak)); for (int i = 0; i < drb; i++) { printf("Kerem a diak nevet: "); scanf("%s", &d[i].nev); printf("Kerem a diak szuletesi evet: "); scanf("%d", &d[i].szuletesiEv); printf("Kerem a diak jegyeinek atlagat: "); scanf("%lf", &d[i].atlag); printf("\n"); for (int i = 0; i < drb; i++) { printf("%s, %d, %.2lf\n", d[i].nev, d[i].szuletesiEv, d[i].atlag); printf("\n"); free(d); return 0; } 1. Feladat Kérjük be a diákok számát, majd dinamikusan alakítsunk ki egy tömböt diákokból. Mindegyik diáknál tároljuk a diák nevét és jegyeinek átlagát. Határozzuk meg és írjuk ki: a legjobb átlaggal rendelkező diák nevét és átlagát, a legrosszabb átlaggal rendelkező diák nevét és átlagát, az osztályátlagot (diákok átlagának átlagát). A program kimenete: Kérem a diákok számát: 4 1. diak neve: Aladar 1. diak jegyeinek atlaga: 4.17 2. diak neve: Emoke 2. diak jegyeinek atlaga: 1.19 3. diak neve: Katalin 3. diak jegyeinek atlaga: 2.71 4. diak neve: Erno 4. diak jegyeinek atlaga: 2.08 Legjobb diak: Emoke, 1.19 Legrosszabb diak: Aladar, 4.17 Osztalyatlag: 2.54 2. Feladat Bővítsük az előző feladatot! Határozzuk meg melyik a diákok közül a "legátlagosabb", tehát hogy melyik diák jegyeinek az átlaga van legközelebb az osztályátlaghoz! Megj.: Valós szám abszolút értékét a fabs() függvény segítségével tudjuk meghatározni. A program kimenete: Kerem a diakok szama: 4 1. diak neve: Aladar 1. diak jegyeinek atlaga: 4.17 2. diak neve: Emoke 2. diak jegyeinek atlaga: 1.19 3. diak neve: Katalin 3. diak jegyeinek atlaga: 2.71 4. diak neve: Erno 4. diak jegyeinek atlaga: 2.08 Legjobb diak: Emoke, 1.19 Legrosszabb diak: Aladar, 4.17 Osztalyatlag: 2.54 Atlaghoz legkozelebbi diak: Katalin, 2.71 3. Feladat Definiáljunk struktúrát, amely raktáron levő termékek adatait fogja tárolni (áru neve, egységára, raktáron levő darabszám). Kérjük be a felhasználótól, hogy mennyi fajta termék van a raktárban, majd mindegyik terméknek olvassuk be a nevét, egységárát és a darabszámát. Határozzuk meg: mennyi a raktáron levő termékek összára, melyik termékből van a legkevesebb a raktáron, melyik termékből van a legtöbb a raktáron, · melyik a legolcsóbb termék, melyik a legdrágább termék. A program kimenete: Mennyi fajta termeket szeretnel megadni? 5 1. termek neve: ceruza 1. termek ara: 0.55 1. termek darabszama: 100 2. termek neve: radir 2. termek ara: 0.35 2. termek darabszama: 70 3. termek neve: vonalzo 3. termek ara: 0.95 3. termek darabszama: 50 4. termek neve: fuzet 4. termek ara: 0.60 4. termek darabszama: 120 5. termek neve: kepeslap 5. termek ara: 0.75 5. termek darabszama: 80 Termekek osszara: 259.00 Legkevesebb van a raktaron: vonalzo (50 darab) Legtobb van a raktaron: fuzet (120 darab) Legolcsobb termek: radir (0.35 EUR) Legdragabb termek: vonalzo (0.95 EUR) 4. Feladat Készítsen programot tetszőleges intervallumról származó egész értékek kiíratására, írja ki az átlagot, összeget, legkisebb és legnagyobb értéket! alakítsa át a programot véletlen generált tört számokkal való munkára <u>Véletlen generálás tetszőleges intervallumon:</u> **Egész számok** x = rand() % (max + 1 - min) + min;**Törtek [0,max] között** y = ((float)rand()/(float)(RAND_MAX)) * max; **Törtek [min,max] között** y = ((float)rand()/(float)(RAND_MAX)) * (max - min) + min; In [4]: #include <stdio.h> #include <time.h> #include <stdlib.h> int main(){ srand(time(NULL)); int x = 0, n = 100; float y, sum = 0; int min = -10; int max = 10; float lowest = (float)max; float highest = (float)min; for (int i = 0; i < n; i++){ x = rand() % (max + 1 - min) + min;printf("%d ", x); printf("\n"); for (int i = 0; i < n; i++){ y = ((float)rand()/(float)(RAND_MAX)) * (max - min) + min; sum = sum + y;if (y < lowest){</pre> lowest = y;if (y > highest){ highest = y;if (i % 8 == 0){ printf("\n"); } printf("%10f ", y); printf("\n-----"); printf("\nHighest value: %f", highest); printf("\nLowest value: %f", lowest); printf("\nSum: %.3f", sum); printf("\nAverage: %.3f", sum/n); } -9 -3 -2 -10 9 3 7 5 -3 -2 6 3 5 2 -2 2 -4 7 -6 1 -8 6 1 -1 2 -8 -8 -6 3 1 3 4 6 -10 4 4 3 -1 -3 8 7 -10 -2 0 2 6 -9 7 3 -7 -3 5 7 9 -8 -1 9 -8 3 1 3 -7 -7 -2 -7 5 10 6 -7 -6 1 -10 -8 10 -2 3 5 -1 -1 -5 1 6 9 -5 2 -10 4 -1 -8 -5 10 3 -2 -8 9 0 8 6 -7 -10 1.705128 -1.364195 4.904046 4.807178 6.970860 7.988960 6.031376 1.243656 5.020386 -6.067975 -1.112619 -1.407558 6.049883 -8.853821 7.845280 -3.091904 6.052439 -5.150682 -1.369951 2.208429 -9.570568 6.731760 -8.1884289.314817 $-0.412106 \quad -1.976123 \quad -4.659428 \quad -5.805125 \quad -1.173049 \quad -0.224522 \quad -6.674973 \quad -9.467920$ 8.411283 8.229073 5.339258 5.382142 6.218035 1.370632 -3.374202 2.267919 2.516812 -5.528923 9.176014 -2.462802 -1.596898 -1.936604 6.129639 -5.544458 2.912713 -5.240312 6.663973 3.342146 -8.508551 8.475544 2.656963 1.079344 -3.500579 7.997536 5.274219 5.326372 -2.226987 8.599245 5.858452 -3.815703 $6.828320 \quad 1.197710 \quad -8.433561 \quad 3.046354 \quad -7.431658 \quad -1.807764 \quad -4.685728 \quad 5.085155$ 2.663315 -5.509714 -7.377647 -8.933584 2.553681 8.751991 -4.478042 -4.533606 -6.488319 -7.814071 8.808540 -4.996871 -9.338527 1.465503 6.082474 -2.839106 $-0.536963 \qquad 1.356692 \quad -7.512733 \qquad 7.236052 \quad -0.044062 \qquad 8.345718 \quad -6.579652 \quad -3.215742$ -0.456572 -5.013214 9.830612 2.111771 ------------Highest value: 9.830612 Lowest value: -9.570568 Sum: 37.131 Average: 0.371 5. Feladat Készítsen programot, ami beolvasott jelszóról meghatározza, hogy mennyire erős! A program értékeli a jelszó hosszát és hogy tartalmaz-e kis-, nagy karaktereket, számot, valamint szimbólumot. **Mintakimenet:** Adja meg a jelszót! asd123 A-Z a-z + sym hossz -A jelszava gyenge, mert 4 feltételből csak 1 teljesül. 6. Feladat Hozzon létre egy tetszőleges méretű tömböt, majd töltse fel véletlen számokkal a 0-10 közé eső számokkal. A tömböt írassa ki, majd hozzon létre egy új tömböt, ahova az eredeti tömb minden értéke annyiszor kerül be, amennyi a szám értéke.

pl: 24 --> 2 2 4 4 4 4

In []:

Struktúra és struktúratömb