

## 13. gyakorlat

### Struktúrák használata

#### I. Új típus új műveletekkel

1. Definiáljon dátum típust. Olvasson be két dátumot, és határozza meg melyik a régebbi.

```
#include <stdio.h>
typedef struct datum {                               /*globális típus definíció*/
    int ev;
    int ho;
    int nap;
} Datum;

void beolvas(Datum * d);
void kiir(Datum d);
Datum regebbi_datum(Datum d1, Datum d2);

int main() {
    Datum d1, d2;
    printf("Első dátum:\n"); beolvas(&d1);
    printf("Második dátum:\n"); beolvas(&d2);
    Datum regi = regebbi_datum(d1, d2);
    kiir(regi);
    return 0;
}

void beolvas(Datum * d) {
    printf("Ev: "); scanf("%d", &d->ev);
    printf("Ho: "); scanf("%d", &d->ho);
    printf("Nap: "); scanf("%d", &d->nap);
    return ;
}

void kiir(Datum d) {
    printf("Regebbi: %d.%d.%d.\n", d.ev, d.ho, d.nap);
    return ;
}

Datum regebbi_datum(Datum d1, Datum d2) {
    if (d1.ev==d2.ev) {
        if (d1.ho==d2.ho) {
            if (d1.nap>=d2.nap) return d2;
            else return d1;
        }
        else if(d1.ho>d2.ho) return d2;
        else return d1;
    }
    else if (d1.ev>d2.ev) return d2;
    else return d1;
}
```

### Önálló feladatok:

- a) Egészítse ki a programot a beolvasott év, hónap, nap értékek ellenőrzésével.
- b) Oldja meg a beolvasást dátum formátumban.

```
void beolvasFormazott(Datum * d){
    char str[12];
    int ok;
    do {
        ok = 1;
        scanf("%s", str);
        if (sscanf(str, "%d.%d.%d.", &d->ev, &d->honap, &d->nap) != 3) {
            printf("Hibás a dátum!\n");
            ok = 0;
        }
    } while (!ok);
    return ;
}
```

- c) Megadott dátummal született ember hány éves.

```
int HanyEves(Datum d) {
    char ho[3];
    int ev, nap;
    sscanf(__DATE__, "%s %d %d", ho, &nap, &ev);
    return ev-d.ev;
}
```

A \_\_DATE\_\_ előredefiniált makró a rendszerdátumot tartalmazza "Nov 21 2017" formátumban.

2. Egy geometriai programban körök adatait kell tárolni: középpont (x, y koordináta) és sugár. Állapítsa meg két körről, hogy azok átfedik-e egymást. Két kör akkor van átfedésben, ha középpontjaik távolsága kisebb, mint a sugaraik összege (Pitagorasz-tétel).

```
#include <stdio.h>
#include <math.h>

typedef struct kor {
    double x, y, r;
} Kor;

void beolvas(Kor * k) {
    scanf("%lf %lf %lf", &k->x, &k->y, &k->r);
    return ;
}

int atfed(Kor k1, Kor k2) {
    return sqrt(pow(k1.x-k2.x, 2)+pow(k1.y-k2.y, 2))<k1.r+k2.r;
}

int main(void) {
    Kor a, b;
    beolvas(&a);
```

```

    beolvas(&b);
    printf("%s\n", atfed(a, b) ? "Atfedik egymast" : "Nem fedik at egymast.");
    return 0;
}

```

A kör struktúra definícióját adja meg úgy is, hogy a kör középpontja 2 taggú struktúra. Módosítsa a programot ennek megfelelően.

```

typedef struct pont {
    double x, y;
} Pont;
typedef struct kor {
    Pont kp;
    double r;
} Kor;

typedef struct kor {
    struct { double x,y; } kp;
    double r;
} Kor;

```

Ekkor a középpont koordinátáira hivatkozás: **k.kp.x** ha k Kor típusú változó, vagy **k → kp.x** ha k egy Kor típusú változóra mutató pointer.

## II. Struktúrát visszaadó függvények

3. Töltsön fel egy integer tömböt 1 és 100 közé eső véletlenszámokkal, majd válogassa szét az elemeket két csoportba: páros és páratlan számok.

```

#define N 10          //tömb mérete
struct tomb{ int eredmeny[N], db; }; //struktúra adattagja lehet tömb

void tombFeltolt(int * tomb);
void tombKiir(struct tomb t);
struct tomb tombSzetvalogat(int * tomb, int paritas);

int main() {
    int t[N];
    tombFeltolt(t);
    struct tomb paros = tombSzetvalogat(t, 0);
    struct tomb paratlan = tombSzetvalogat(t, 1);
    printf("\nPáros elemek: ");
    tombKiir(paros);
    printf("\nPáratlan elemek: ");
    tombKiir(paratlan);
    return 0;
}

/* 1 és 100 közötti véletlenszámokkal tölts fel a tömböt */
void tombFeltolt(int *tomb) {
    int i;
    srand(time(0));          //véletlenszám generátor inicializálása
    for (i=0; i<N; i++)
        tomb[i]=rand()%100+1; //értékkadás tömbelemeknek véletlenszám generátorral
    return ;
}

```

```

void tombKiir(struct tomb t) {           //fv paramétere struktúra
    int i;
    for (i=0; i<t.db; i++) printf("%d, ", t.eredmeny[i]);
    //struktúra adattagjára hivatkozás . operátorral
    return ;
}

struct tomb tombSzetvalogat(int * tomb, int paritas){ //fv visszatérési értéke struktúra
    struct tomb e;
    int i, j=0;
    for(i=0; i<N; i++) {
        if (tomb[i]%2==paritas) {
            e.eredmeny[j] = tomb[i];
            j++;
        }
    }
    e.db = j;
    return e;
}

```

A feladat másik megoldása, hogy dinamikusan foglalunk helyet a részhalmazoknak és a struktúra a dinamikusan foglalt tömbre mutató pointert és annak méretét tartalmazza.

### Házi feladatok:

1. Írjon C programot a másodfokú egyismeretlenes egyenletek megoldásához. Az ilyen egyenletek általános alakja:  $ax^2 + bx + c = 0$ , ahol a, b és c konstansok. Megoldása:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Ha a=0 de b és c nem 0, az egyenlet elsőfokú, megoldása:  $x = -c/b$ .

Ha c=0 de a és b nem 0, az egyenlet hiányos, megoldásai:  $x_1 = 0$ ,  $x_2 = -b/a$ .

Ha b=0 de a és c nem 0, az egyenlet tiszta másodfokú, megoldásai:  $x_1, x_2 = \pm \sqrt{-c/a}$

Ha mindhárom konstans 0, akkor bármely szám megoldás.

Ha a és b 0, viszont c nem, akkor nincs megoldás.

A megoldóképletben a gyök alatti kifejezés a másodfokú egyenlet diszkriminánsa. Ha  $d > 0$ , az egyenletnek két valós gyöke van. Ha  $d = 0$ , az egyenletnek egy valós gyöke van ( $x_1 = x_2$ ). Ha  $d < 0$  az egyenletnek két komplex gyöke van.

Struktúrák használata azért szükséges, mert a valós és komplex gyökök számítását végző fv több értéket kell visszaadjon.

Egy komplex számot leíró struktúra definíciója:

```

struct komplex {
    double valos;
    double kepzetes;
};

```

2. Egy egyetemi oktató 5 csoportba oszt szét 150 db ZH dolgozatot (max. 50 pont). 0-25 pont elégtelen, 26-32 pont elégséges, 33-39 pont közepes, 40-45 pont jó, 46-50 pont jeles. Készítsen C programot, amely elvégzi ezt a feladatot.

3. Írja meg azt a C programot, amely megállapítja egy megadott dátumról, hogy hányadik napja az évnek. Vegye figyelembe, hogy szökőévről van-e szó!

b) Készítse el az algoritmus fordítottját is: ha adott, hogy hányadik napja az évnek, mondja meg, hogy melyik napról van szó. Pl. 2016-ban (szökőév) az év 60. napja február 29.

c) Készítse el azt a függvényt, amely megállapítja egy megadott dátumról, hogy helyes-e. Pl. 2016.01.32. hibás dátum.

4. Olvassa be két síkbeli pont  $x$  és  $y$  koordinátáit. Számítsa ki a két pont távolságát (a Pitagorasz tétel alkalmazásával). A feladat megoldásához definiáljon saját `Sikpont` típust.

5. Geometriai programban egy négyzetet az oldalhosszúságával és a bal felső csúcsának koordinátaival adunk meg. Határozza meg a négyzet középpontjának (az átló felezőpontjának) koordinátáit.

6. Geometriai programban egy pont és egy kör egymáshoz viszonyított helyzetét vizsgáljuk. Állapítsa meg, hogy a megadott pont rajta van-e a megadott körvonalon, vagy azon belül, vagy azon kívül helyezkedik-e el. A pontot a koordinátaival adja meg, a kört pedig a középpont koordinátaival és a sugarával.