

# Programozás alapjai

## *4. gyakorlat*

*Morvai Flórián, 2010  
Dr. Dévényi Károly előadásvázlata alapján*

# Vezérlési szerkezetek

- Szekvencia
- Elágazás
- Iteráció
  - Elöltesztelő
  - Hátultesztelő
  - Számlálósos

# Feltételes kifejezés

- `feltétel ? haigaz : hahamis`
- `(x%y == 0) ? "igen" : "nem"`

# Elágazás - if

- `if (feltétel1) {  
    utasítás1;  
} else if (feltétel2) {  
    utasítás2;  
} else {  
    utasítás;  
}`

# Elágazás - switch

- `switch (változó) {`  
    `case érték1:`  
        `utasítás1;`  
        `break;`  
    `case érték2:`  
        `utasítás2;`  
        `break;`  
    `...`  
    `case értékN:`  
        `utasításN;`  
        `break;`  
    `default:`  
        `utasítás;`  
        `break;`  
}

# Ismétlés

- Elöltesztelő

- `while (feltétel) {  
    utasítások;  
}`

- Háttülesztelő

- `do {  
    utasítások;  
} while (feltétel)`

# Számlálósos ism.

- `for (inicializáció;feltétel;növekmény) {  
utasítások;  
}`
- `inicializáció;  
while (feltétel) {  
utasítások;  
növekmény;  
}`

- Készíts egy programot, ami bekér egy egész számot és kiírja, hogy az adott szám páros vagy páratlan-e.
- Készíts egy programot, ami bekér két egész számot és kiírja, hogy az első szám osztható-e a másodikkal.
  - Mi történik, ha a második szám 0? Javítsd ki a programot!
  - Nézd meg mi történik, ha a nullával osztás elkerülését célzó feltételben az = vagy az == operátort alkalmazod! És ha az operátor két operandusát is felcseréled?

- Módosítsd a 3. programot úgy, hogy ez az oszthatósági vizsgálat egy függvényen belül legyen. Ha a két szám osztója egymásnak a függvény visszatérési értéke legyen 1, különben pedig 0.
- Írjuk ki egyetlen printf segítségével, hogy egy szám páros vagy páratlan-e.
- Írjuk ki egyetlen printf használatával, hogy egy szám osztója-e egy másiknak. A 0 semminek sem osztója.
- Írjuk meg az előző programot if utasítás és feltételes kif. használata nélkül!

- Írjunk egy függvényt, ami egy  $n$  egész számot kap paraméterként és kiírja, hogy a hét  $n$ . napja milyen nap. Írjunk hozzá főprogramot is.
- Próbáld ki, hogy a switch működik-e valós értékekkel. Módosítsd a paraméter típusát int-ről float-ra!
- Próbáld ki, mi történik ha elhagyjuk a break utasításokat.

- Írjunk egy programot, ami kiírja 1-től 10-ig számokat.
- Írjunk olyan programot, ami addig kér be számokat a billentyűzetről, amíg a beírt szám nem 0!
- Módosítsuk a programot úgy, hogy végeredményként írja ki a beírt számok összegét! (break nélkül + break)

- Írjunk egy olyan programot 'do-while' ciklus segítségével, ami 0 végjelig kér be számokat, majd kírja azok összegét. A ciklusban ne szerepeljen a 'break' utasítás.
- Írjunk egy programot, ami összeszorozza 1-10-ig a számokat. Írjuk meg for és while ciklussal is!
- Módosítsuk a for ciklust úgy, hogy csak minden 3-mal osztható számot szorozzon össze!

- Írj programot, mely bekéri a hónap sorszámát és a napot, és megállapítja, hogy helyes-e!
- Írj programot, mely bekér egy  $n$  pozitív egész számot, és megadja az első  $n$  természetes szám összegét!

- Számítsuk ki a fogyasztás és a hónap alapján az adott hónapban fizetendő gázárát, ha tudjuk, hogy a gázár sávosan változik:

- az első 1000 egység ára 100/egység
- a második 1000 egység ára már 200/egység;
- egyébként a gázár 300/egység
- a téli hónapokban (nov, dec, jan, feb) a gázárát 1.2 -vel szorozzák.

- Kérj be folyamatosan karaktereket addig, amíg egy '#' jelet nem kapsz. Ha megkaptad a végjelet, akkor írd ki, hogy addig hány kisbetűt, nagybetűt, és hány számot olvastál!
- Írj olyan programot, mely bekér egy pozitív egész számot és kiszámítja a Fibonacci-sorozat n-ik elemét! Az eljárást valósítsd meg ciklussal és rekurzív függvénnnyel is! Melyik gyorsabb?
  - $F(0) = 0$ ;  $F(1) = 1$
  - $F(n) = F(n-1) + F(n-2)$ , ha  $n > 1$