

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

1A. Algoritmusleíró eszközök

Sok feladat megoldása egymást követő lépésekből áll. Ezek megfelelő sorrendben történő végrehajtása vezet el a kiinduló helyzettől a kívánt végeredményig. A matematika és informatika feladatok megoldásához is sok esetben ilyen lépéssorozat szükséges, amit algoritmusnak hívunk. Az informatikai algoritmusok leírására többféle módszer alakult ki. Ebben a feladatban weboldalt és vektorgrafikus képet kell készítenie a minta és a leírás alapján, amely bemutat néhány algoritmusleíró módszert!

- Az *algoritmus.html* fájl már tartalmazza a weboldal szövegének forrását. Ebben a fájlban kell kialakítania a dokumentum szerkezetét HTML elemek segítségével.
- A weboldal formázását biztosítja a *stilus.css* állomány, amely már tartalmaz CSS elemeket. A továbbiakban ezt a fájlt kell módosítani és bővíteni a kért formázások megadásához.
- A weboldalon el kell helyezni a *folyamat.svg* vektorgrafikus képet, amely részben van kész. A megoldás során ezt a képet is módosítani kell.

A weboldalt csak stílusokkal formázza a *stilus.css* stíluslap módosításával, maga a weboldal ne tartalmazzon CSS stíuselemeket vagy a HTML elemek paraméterekkel történő formázását!

Az *algoritmus.html* oldal szerkesztése:

1. Az *algoritmus.html* oldalhoz csatolja stíluslapként a *stilus.css* állományt!
2. Állítsa be, hogy a böngésző címsorában az „Algoritmusleíró eszközök” szöveg jelenjen meg!
3. A weboldal szövegének szerkezetét alakítsa ki a minta és az alábbi leírás szerint:
 - a. Legyen az „Algoritmusleíró eszközök” szövegrész első szintű címsor, míg „A programfejlesztés lépései”, a „Folyamatábra”, a „Struktogram”, valamint a „Mondatszerű leírás” szövegrész kettes szintű címsor!
 - b. A további szövegek egy-egy bekezdésben szerepeljenek, kivéve azokat a részeket, amelyek sorszámozott listában vagy felsorolásban találhatók, valamint a mondatszerű leírás fejezet végén szereplő algoritmus sorai! Alakítsa ki a megfelelő szövegrészekből a bekezdéseket!
 - c. A sorszámozott listát és a felsorolást alakítsa ki a minta alapján!
 - d. Illessze be képként a folyamatábráról szóló fejezet végére a *folyamat.svg* vektorgrafikus állományt! A képhez állítsa be a „Folyamatábra” helyettesítő szöveget!
 - e. Hozzon létre egy 2 oszlopos és 7 soros táblázatot a struktogramról szóló szövegrész végére!
 - f. Az 1-3. és a 7. sor celláit vonja össze!
 - g. A 4-6. sor első cellájában a ` ` (nem törhető szóköz) legyen!
 - h. A többi cellába a minta szerinti tartalmat írja be vagy a mondatszerű leírásból másolja be!
 - i. Az első alcím utolsó bekezdésében szereplő „Euklideszi algoritmus” szövegrész legyen hivatkozás, amely az Euklideszi algoritmus Wikipédia oldalára mutat. A hivatkozást a HTML állomány végén találja. Másolja ide majd onnan törölje azt!

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

A *stilus.css* állomány elkészítése

A weboldal formázását a CSS állomány módosításával és a stíluselemek HTML állományban való alkalmazásával végezze el a következő feladatokban leírtak szerint!

4. A bekezdésekben szereplő szövegek legyenek sorkizárt igazításúak!
5. A beillesztett *folyamat.svg* képet igazítsa vízszintesen közepre!
6. A struktogramot bemutató táblázat formázásához több kijelölőt is kell készítenie, illetve módosítania. A táblázatra vonatkozó *struktogram* osztálykijelölőt egészítse ki úgy, hogy a táblázat szövege Open Sans betűtípusú legyen 12 pontos betűmérettel! A táblázatot szegélyezze 2 pixel vastagságú folytonos fekete vonallal a mintának megfelelően!
7. A táblázat celláinak formázását a *struktogram* osztálykijelölő módosításával állítsa be: a belső margó legyen 0.5 em méretű és a szegély 2 pixel vastagságú folytonos fekete vonal!
8. A 4-6. sor első celláinak formázására hozzon létre egy osztálykijelölőt *balsav* néven! Állítsa be a segítségével, hogy a cella szélessége 1.5 em legyen, és a cellának ne legyen szegélye!
9. A `<pre>` elem stílusát egészítse ki a következőképpen:
 - a. a blokk szélessége legyen 12 em;
 - b. a blokk vízszintesen középre igazítva jelenjen meg;
 - c. a blokk háttérszíne legyen világosszürke színű (RGB kód: #D3D3D3);
 - d. a blokk előtt és után 2 em nagyságú külső margó legyen!
10. A CSS lapon szereplő osztálykijelölőkre történő hivatkozásokat helyezze el a HTML oldal megfelelő elemeinél! A táblázatot rendelje a *struktogram* osztályhoz, a ciklusfeltételt tartalmazó cellát a *ciklus* osztályhoz, a bal oldali, szöveg nélküli (nem törhető szóközt tartalmazó) cellákat a *balsav* osztályhoz!

A feladat a következő oldalon folytatódik!

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Minta a weboldalhoz:

Algoritmusleíró eszközök

A programfejlesztés lépései

A számítógépes programok készítése összetett folyamat, mely általában az alábbi részekre bontható:

1. a feladat pontos meghatározása (specifikáció);
2. a feladat megoldásának kitalálása, majd a megoldás lépéseinek meghatározása (algoritmuskészítés);
3. a készült algoritmus alapján a program elkészítése egy adott programozási nyelven (kódolás);
4. az elkészült programkód fordítása és futtatása, a futási eredmények vizsgálata (tesztelés, hibakeresés);
5. a program használatához és továbbfejlesztéséhez szükséges leírások elkészítése (dokumentáció).

A fent leírt pontok a legegyszerűbb esetben időben egymás után következnek, de sokszor van eltérés. Például a dokumentáció mindegyik lépés tanulságait tartalmazza, így a programkészítés során folyamatosan bővül. Vagy a tesztelés során talált hibák néha a specifikáció pontosítását, az algoritmus és a program módosítását is magukkal hozzák. Ezután persze ismételt tesztelni kell a programot, vajon az az elvárásoknak megfelelően működik-e, és ha nem, akkor ismét változtatni kell.

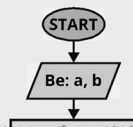
Az egyszerűbb feladatok megoldásakor a fenti lépések közül néhány elmarad, vagy csak gondolatban történik meg. Előfordul, hogy nem készül dokumentáció, csak a forrásprogramba kerül néhány sor, ami röviden ismerteti az egyes kódrészleteket. Sok esetben az algoritmus nem kerül leírásra, csak a forrásprogram készül el. Ez egyszerűsíti a fejlesztő munkáját, kiolvasható belőle a megoldás gondolatmenete, mégis vannak hátrányai. Például a programozó felhasználja az adott programozási nyelv sajátosságait, ezért a megoldás nem lesz általános, nehezen átváltható más programozási nyelvre.

Az összetett programok készítése során ezért javasolják az algoritmus lejegyzését. Az algoritmus leírása a programozás tanulása során is fontos, segíti az algoritmikus gondolkodás fejlődését, az egyes gondolati lépések jobb megértését.

A továbbiakban néhány algoritmusleíró eszközt mutatunk be az [Euklideszi algoritmus](#) segítségével, amely két pozitív egész szám legnagyobb közös osztójának kiszámítását végzi. Az algoritmus bemenete a két pozitív egész szám, eredménye a legnagyobb közös osztó.

Folyamatábra

Az egyik legkorábban kialakult algoritmusleíró eszköz. Neumann János és Herman Goldstine munkáiban találjuk meg először, ők fejlesztették és javasolták a használatát munkatársaiknak az első számítógépek építése idején, az 1940-es években. A lépések folyamatát egy irányított vonal jelzi, mely egy kezdő elemből indul és egy záró elembe végződik. Az egyszerű utasítások egy-egy téglalap alakú dobozban találhatók, míg a bejövő vagy kimenő adatokat egy paralelogramma jelzi. A folyamat döntési pontok segítségével több ágra bontható, így létrehozhatók elágazások és ismétlések.

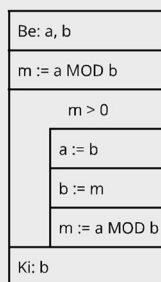


Struktogram

Az 1960-as években kialakult a strukturált programozás, amely egyik alapvetése, hogy csak bizonyos programszerkezetek használatát javasolja:

- szekvencia - az utasítások egymás utáni végrehajtása
- elágazás - egy feltétel értékétől függően különböző utasítások hajthatók végre, a programvezérlés több ágra bontható
- iteráció - egy feltétel értékétől függően az utasítások egy csoportja újra és újra végrehajtható

Sikerült igazolni, hogy a nem strukturált programszerkezetek használatával elkészített programoknak van strukturált megfelelője, így a megoldható problémák körét nem szűkíti, ha csak strukturált elemeket alkalmazunk. Így a '70-es évektől a strukturált programozást egyre szélesebb körben alkalmazták. Mivel a folyamatábra nem strukturált elemek létrehozását is lehetővé teszi, ezért létrehozták a csak strukturált elemeket tartalmazó algoritmusleíró eszközt, a struktogramot.



Mondatszerű leírás

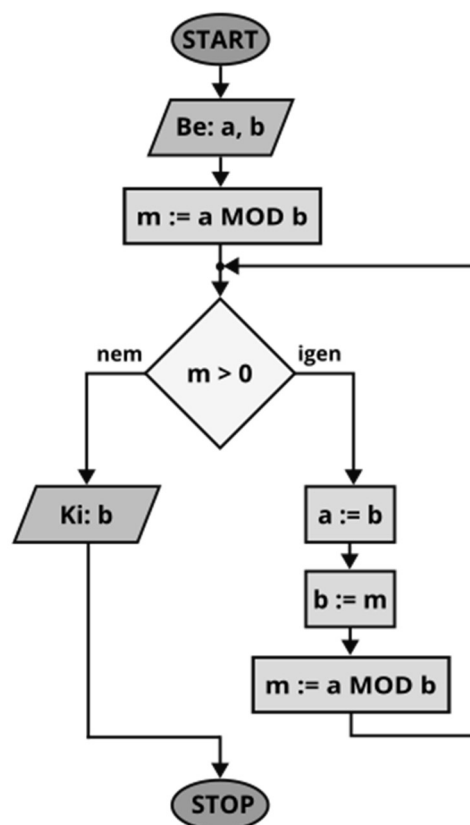
A forrásprogramhoz hasonlóan az algoritmusok is leírhatók szöveggel. Ebben az esetben is a strukturált programozásnál alkalmazott programszerkezeteket alkalmazzuk, tehát szekvencia, elágazás és iteráció fordul elő. A tagolás és a jobb olvashatóság érdekében általában egy sorban egy utasítás helyezünk el, valamint az elágazások ágait és a ciklusok magját tagolással is jelezzük.

```
Be: a, b
m := a MOD b
Ciklus amíg m > 0
  a := b
  b := m
  m := a MOD b
Ciklus vége
Ki: b
```

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

A kép elkészítése

11. A weboldalon megjelenő kép részben már elkészült. Nyissa meg az ön által használt vektorgrafikus szerkesztő programmal a *folyamato.svg* vagy *folyamato.odg* képet! Végezze el rajta a következő feladatokban leírt módosításokat, majd mentse el változatlan állománytípussal! Az ODG formátumú képet exportálja .SVG-be is!
12. A beolvasást követő, m változónak értéket adó utasítás hiányos. Egészítse ki az utasítás szövegét! Formázza a téglalapot és a benne lévő szöveget az alábbiak szerint:
 - a. A téglalap szélességét állítsa be úgy, hogy a szöveg még elférjen (a szöveg torzítása nélkül), de feleslegesen nagy üres hely ne maradjon!
 - b. A téglalapot és a benne lévő szöveget igazítsa vízszintesen és függőlegesen egymáshoz képest középre!
 - c. Igazítsa az utasítást tartalmazó téglalapot a felette lévő nyílhoz képest vízszintesen középre!
13. Az m változónak értéket adó utasítás a ciklus végéről hiányzik. Másolja le az előbb elkészített utasítást és helyezze el úgy, hogy a már ott lévő nyíl a téglalap felső vonalának középsőéhez csatlakozzon!
14. Az értékadó utasításokat tartalmazó négy téglalap kitöltését állítsa be 50%-os átlátszatlanság mellett RGB(0,255,0) kódú zöld színre!
15. A feltételt tartalmazó rombusz fölötti nyílra helyezzen el egy 0,15 cm átmérőjű, keretvonal nélküli, kitöltött kört! A kitöltésének színe egyezzen meg a többi alakzat keretvonalának színével! A kört vízszintesen igazítsa a nyílhoz képest középre!
16. Készítse el az utolsó utasítás téglalapjának aljáról induló minta szerinti nyilat, amely a feltétel fölötti körhöz csatlakozik! A törött vonal a téglalap aljának közepéről induljon! A törött vonal szakaszai legyenek a többi folyamatjelző szakasszal párhuzamosak vagy merőlegesek, a felső végén legyen nyíl! A törött vonal legyen a többi nyíl alakzat formátumával megegyező színű és vonalvastagságú!



35 pont