

ELÁGAZÁSOK

GYAKORLÓ FELADATOK C#-BAN

EGYIRÁNYÚ ELÁGAZÁS

MAGYARÁZAT

Egy műveletsort csak egy feltétel teljesülésekor kell végrehajtani. Szokták ezt **feltételes utasítás**nak is nevezni.

Ha *Feltétel* teljesül, akkor *Utasítás(ok)* végrehajtásra kerül(nek), egyébként nem. A program az *Elágazás vége* után folytatódik (itt az *Elágazás vége* kiírás elhagyható).

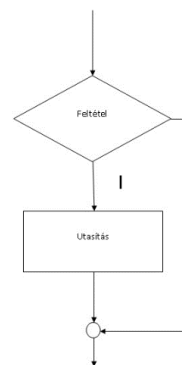
ALGORITMUSA PSZUDOKÓDDAL

Ha Feltétel **akkor** Utasítás(ok)
Elágazás vége

vagy rövidebben:

Ha Feltétel **akkor** Utasítás(ok)

FOLYAMATÁBRÁJA



C# KÓDJA

```
if (logikai kifejezés)
{
    Igaz érték esetén végrehajtandó utasítások;
}
```

...vagy rövidebben (egy utasítás esetén):

```
if (logikai kifejezés) Igaz érték esetén végrehajtandó utasítás;
```

PÉLDA

Algoritmusa:

Ha $y > 0$ **akkor**
Ki: A szám pozitív.
Elágazás vége

C# kódja:

```
if (y > 0) Console.WriteLine("A szám pozitív.");
```

Jelentése:

Ha az y a 0-nál nagyobb, akkor kiírja, hogy „A szám pozitív.”

FONTOS INFORMÁCIÓK

Az elágazás létrehozásának kulcsszava **if**, amely létező angol szó, és magyar jelentése **ha**. Az if után gömbölyű zárójelek között áll a feltétel, mely logikai kifejezés kell legyen - vagyis kiértékelésének eredménye vagy **igaz** vagy **hamis** kell legyen. A gömbölyű zárójelek után áll a feltételhez tartozó utasítás vagy utasításblokk.

Az elágazás egyetlen utasításnak fogható fel - mely bár több részből áll, de illeszkedik a szekvencia szabályhoz. Vagyis az elágazás előtti utasítások végrehajtása után következik az if végrehajtása, majd történjen bármi az if végrehajtása közben - utána folytatódik a program a rákövetkező utasítások végrehajtásával.

Az if először kiszámítja a kifejezés értékét. Amennyiben az true (igen), úgy végrehajtódik az if-hez tartozó utasításblokk. Amennyiben a kifejezés értéke false (nem), úgy az utasításblokk nem hajtódik végre - de a program ettől nem állhat le. Ez a kifejezés értéke csak az if belső végrehajtását befolyásolja - a program futását nem.

Amennyiben az if-hez nem egyetlen utasítás, hanem több utasítás tartozik, úgy azt **blokk**ba kell foglalni. Az utasításblokkokat a kapcsos zárójelek határolják - így ezeket kell használnunk.

A kapcsos zárójelekkel egyetlen utasítást is körbe lehet zárni. Nincs olyan szabály, hogy az utasításblokk legalább két utasítást tartalmazzon. Sőt, **az utasításblokk üres is lehet** (nulla darab utasítás). E módon az alábbi változatok szintaktikailag mind helyesek:

```
// szintaktikailag helyes if kifejezések
if (a == 0) { } // nincs utasítás, üres az utasításblokk
if (a == 0)
{
    // egyetlen utasítás van, itt is használhatók a kapcsos zárójelek, de el is hagyhatók
    Console.WriteLine("Ez éppen nulla");
}
```

GYAKORLÓ FELADATOK

1. BETŰ?

Döntsd el egy bekért adatról, hogy betű-e! Csak akkor írd ki a konzolra, ha betű.

Segítség:

- A **betu** nevű változó betű-e: `Char.IsLetter(betu)`

2. KÖSZÖNJEK?

Kérdezd meg a felhasználót, hogy köszönjön-e el a program, és ha a válasz „igen”, akkor írd ki: „Viszontlátásra!”. Egyéb válasz esetén ne történjen semmi.

3. SZÓKÖZ

Kérj be egy szöveget, s ha tartalmaz szóközt, akkor írd ki: „A beírt szöveg több szóból áll.”, majd írd ki azt is, milyen hosszú a szöveg. Amennyiben a szöveg nem tartalmaz szóközt, ne írd ki semmit!

Segítség:

- Annak ellenőrzése, hogy a **szoveg** nevű változó tartalmaz-e szóközt: `szoveg.Contains(" ")`
- A **szoveg** nevű változó karakterszámának (hosszának) lekérdezése: `szoveg.Length`

A program egy lehetséges kimenete:

```
Szóköz program. Írj be egy szöveget! Két szó
A beírt szöveg több szóból áll.
A beírt szöveg hossza: 7
```

4. OSZTHATÓSÁG

Írjunk programot, mely bekér egy számot, és eldönti, hogy osztható-e 3-mal, 4-gyel vagy 9-cel. Ebben a feladatban elég, ha a matematikai megközelítés helyett a maradékos osztás alapján dolgozunk.

KÉTIRÁNYÚ ELÁGAZÁS

MAGYARÁZAT

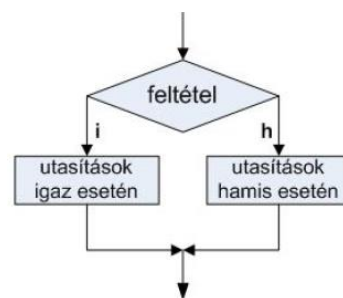
Két műveletsor közül az egyiket akkor kell végrehajtani, amikor a feltétel igaz, míg a másikat akkor, ha a feltétel hamis.

Ha *Feltétel* teljesül, akkor *Utasítás(ok)1* kerül(nek) végrehajtásra, egyébként *Utasítás(ok)2*. Mindkét esetben a program az *Elágazás vége* után folytatódik:

ALGORTMUSA PSZUDOKÓDDAL

```
Ha Feltétel
    akkor Utasítás(ok)1
    különben Utasítás(ok)2
Elágazás vége
```

FOLYAMATÁBRÁJA



C# KÓDJA

```
if (logikai kifejezés)
{
    Igaz érték esetén végrehajtandó utasítások;
}
else
{
    Hamis érték esetén végrehajtandó utasítások;
}
```

...vagy rövidebben (egy utasítás esetén):

```
if (logikai kifejezés) Igaz érték esetén végrehajtandó utasítások;
else Hamis érték esetén végrehajtandó utasítások;
```

PÉLDA

Algoritmus:

Ha $i \geq 0$
akkor Ki: i
különben Ki: -i
Elágazás vége

C# kódja:

```
if ( i > 0) Console.WriteLine(i);  
else Console.WriteLine(-i);
```

Jelentése:

Ha i nem negatív, akkor önmagát írja ki, ha i negatív, akkor az ellentettjét (ez i abszolút értéke).

GYAKORLÓ FELADATOK

5. MIT CSINÁL?

Add meg az alábbi program pszeudokóddal és folyamatábrával leírt algoritmusát és találd ki, milyen feladatot hajt végre, majd végül készítsd el a teljes programot C#-ban!

```
int szam;  
szam = int.Parse(Console.ReadLine());  
if (szam % 2 == 0)  
    Console.WriteLine("????");  
else  
    Console.WriteLine(" ???? ");
```

6. MELYIK SZÁM A NAGYOBB?

Írj programot, amely bekér két számot, majd kiírja a nagyobbbat.

- Bemenő adatok: Két szám:
 - Szám1:
 - Szám2:
- Kimenő adatok:
 - Az első szám nagyobb.
 - A második szám nagyobb.

Készítsd el a feladat algoritmusát pszeudokóddal, majd folyamatábrával, végül valósítsd meg C# nyelven az algoritmusokat.

7. KISKORÚ...

Készíts programot, mely bekéri a felhasználó életkorát, majd kiírja, hogy a felhasználó megnézheti-e a „piros karikás” (18 év alattiaknak nem ajánlott) filmet!

8. ELŐJEL

Kérj be egy számot, és dönts el (írasd ki), milyen előjelű.

ÖSSZETETT FELTÉTEL

Egy **if** kifejezésbe összetett feltétel is kerülhet, melyek között logikai kapcsolatot létesíthetünk:

- A **tagadás** megvalósítása: **!**
- Logikai **ÉS** kapcsolat: **&&**
- Logikai **VAGY** kapcsolat: **||**

9. NAGYOBB VAGY EGYENLŐ

Írj programot, amely bekér egy számot, majd eldönti, nagyobb-e vagy egyenlő, mint 50. Ha igen, kiírja, hogy: 50-nél nem kisebb, ellenkező esetben azt írja ki, hogy 50-nél kisebb.

10. KÖZÖTT

Kérjünk be egy egész számot, és döntsük el, majd írassuk ki, hogy szigorúan -50 és +50 közé esik-e, vagy sem.

TÖBBIRÁNYÚ ELÁGAZÁS: EGYMÁSBA ÁGYAZOTT IF-EK

MAGYARÁZAT

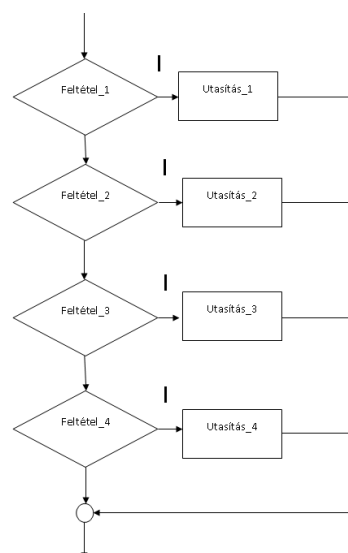
Az **if** elágazás esetén a kulcs szerepkört egyetlen logikai kifejezés tölti be. A logikai kifejezés két lehetséges értékkel bírhat - **igaz** vagy **hamis**. Ennek segítségével két elágazási irányban folytatódhat tovább a program végrehajtása.

Érdekes, de a két elágazási irány a programok tervezése során sokkal gyakrabban elegendő, mint első pillantásra látszik. **A három irányú elágazást** leggyakrabban például két darab összekapcsolt kétirányú elágazással oldjuk meg.

ALGORITMUSA PSZEUDOKÓDDAL

Ha Feltétel1 **akkor** Utasítások1
különben ha Feltétel2 **akkor** Utasítások2
különben utasítások3
Elágazás vége

FOLYAMATÁBRÁJA



C# KÓDJA

```
if (a == 0)
{
    Console.WriteLine("A szám éppen nulla.");
}
else if (a > 0)
{
    Console.WriteLine("A szám pozitív.");
}
else
{
    Console.WriteLine("A szám negatív.");
}
```

Jelentése:

Ha a szám nulla, akkor kiírja, hogy „A szám éppen nulla.”, ha a szám nagyobb nullánál, akkor pozitív, különben negatív.

GYAKORLÓ FELADATOK

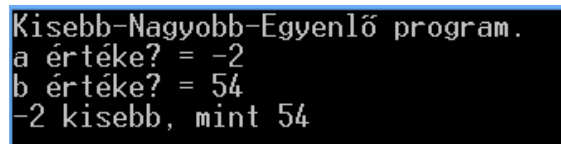
11. CSELEKVŐKÉPES

Készíts programot, mely a beolvasott életkor alapján kiírja, hogy a felhasználó cselekvőképtelen (<14 éves), korlátozottan cselekvőképes (14-18 éves), vagy nagykorú (>18 éves)!

12. KISEBB-NAGYOBB-EGYENLŐ

Írj programot, mely bekér két egész számot, majd eldönti, melyik a nagyobb, és ez alapján kiírja, hogy melyik kisebb/nagyobb a másiknál, egyenlőség esetén azt írja ki. A kiíratásnál formátum string-et használj!

Egy lehetséges kimenet az alábbi:



```
Kisebb-Nagyobb-Egyenlő program.
a értéke? = -2
b értéke? = 54
-2 kisebb, mint 54
```

13. VÍZ HALMAZÁLLAPOTA

Készíts programot, amely bekéri a víz hőmérsékletét, majd eldönti, hogy az milyen halmazállapotú. A halmazállapot lehet folyékony, gőz, vagy jég.

14. POZITÍV-NEGATÍV-NULLA

Írjon programot, amely bekér egy számot, majd eldönti, hogy negatív szám-e. Ha nem, akkor írja ki, hogy nulla vagy pozitív.

15. BALATON VÍZMÉLYSÉGE

Kérd be a Balaton vízmélységét méterben. Ha a vízmélység kisebb, mint 1 méter, akkor pancsolunk benne, ha a vízmélység 1 és 2 m közötti, akkor úszunk, ha mélyebb, mint két méter, akkor nem megyünk be.

TÖBBIRÁNYÚ ELÁGAZÁS: SWITCH-CASE

MAGYARÁZAT

A többirányú elágazás megoldására saját nyelvi konstrukció van - **switch** a neve. Ez angolul **kapcsolót** jelent, mely több irányba is elfordítható.

A switch saját blokkal rendelkezik, melynek kezdetét és végét a szokásos kapcsos zárójelpár jelzi. Ezen blokk összességében több programutasítást is tartalmaz, mely utasítássorozatba belépési pontokat definiálhatunk a case segítségével. A **case** angol szó, magyar **jelentése eset**.

Az esetek az utasítássorozat belépési pontjai, tehát nem önálló programblokkok. Ezért is nem tartozik hozzájuk önálló kapcsos zárójeles blokk. Amennyiben mégis azt szeretnénk, hogy valamely esethez tartozó utasítássorozat végrehajtása után már egyéb switch belsejében szereplő utasítás ne hajtódjon végre, úgy ezt kérhetjük a **break** segítségével. A break hatására a switch hátralévő utasításai nem hajtódnak végre, hanem **ugrunk a switch mögötti (rákövetkező) utasításra**.

Amennyiben valamely belépési ponthoz tartozó utasítássorozatot nem zárunk le break segítségével, úgy a kilépési utasítás hiányában a vezérlés folytatódhat a következő belépési ponthoz tartozó utasítások végrehajtásával. Ez lehet szándékos, amennyiben eseteket kívánunk összevonni.

ALGORITMUSA PSZUDOKÓDDAL

Elágazás

változó = érték1 esetén:

Utasítások1

változó = érték2 esetén:

Utasítások2

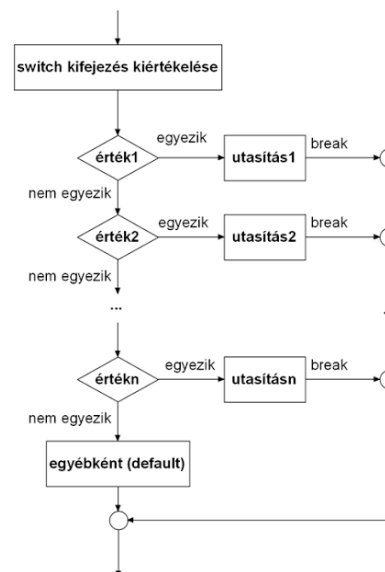
változó = érték3 esetén:

Utasítások3

...

Elágazás vége

FOLYAMATÁBRÁJA



C# KÓDJA

```

switch (kifejezés)
{
    case érték1:
        Utasítások1;
        break;
    case érték2:
        Utasítások2;
        break;
    default:
        Utasítások;
        break;
}

```

A kifejezés (vagy változó) típusa egész szám (int) vagy szöveg (string) kell legyen!

PÉLDA

Algoritmus:

Változó: jegy [egész]

Be: jegy

Elágazás

jegy = 1 esetén

ki: ez egyes

jegy = 2 esetén

ki: ez kettes

jegy = 3 esetén

ki: ez hármas

jegy = 4 esetén

ki: ez négyes

jegy = 1 esetén

ki: ez ötös

egyéb esetben

ki: nincs ilyen jegy

Elágazás vége

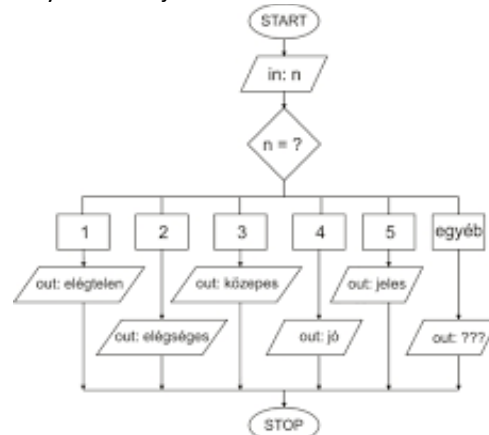
C# kódja:

```
switch (jegy)
{
    case 1:
        // az érdemjegy értéke 1. Ide jönnek az utasítások egészen a break-ig
        Console.WriteLine("ez egyes");
        break;
    case 2:
        // az érdemjegy értéke 2
        Console.WriteLine("ez kettes");
        break;
    case 3:
        // az érdemjegy értéke 3
        Console.WriteLine("ez hármas");
        break;
    case 4:
        // az érdemjegy értéke 4
        Console.WriteLine("ez négyes");
        break;
    case 5:
        // az érdemjegy értéke 5
        Console.WriteLine("ez ötös");
        break;
    default:
        // egyéb érték esetén
        Console.WriteLine("nincs ilyen jegy");
        break;
}
```

Jelentése:

- A **jegy** nevű változóban tároljuk a dolgozatjegyet (egész szám), melynek 1 értéke esetén a **case 1:** eset utasításai kerülnek végrehajtásra, majd a **}** után folytatódik a program végrehajtása.
- 2 értéke esetén a **case 2:** -nél lévő utasítások hajtódnak végre, és így tovább.
- 1-5-ig definiált az elágazás. Egyéb esetekben a **default:** ágban folytatódik a program végrehajtása.

Folyamatábrája:



GYAKORLÓ FELADATOK

16. HÁNY JEGYŰ?

Írasd ki egy kétmilliárd alatti bekért számról azt, hogy hány jegyű! Jegyeinek számát 8-ig szöveggel írasd ki, ez felett már csak annyit, hogy „A szám 8, vagy annál többjegyű!”.

Ha ezzel készen vagy, add meg a feladatnak egy if-else szerkezettel is működő megoldását (a számjegyek számát itt elég számjeggyel kiírtni).

17. AZ ÉV HÓNAPJAI

Beolvasott szám alapján jelenítsd meg az év hónapjait betűkkel, a változót értékei alapján nevezd **honapnak**. Kezeld le az érvénytelen sorszámu hónapot.

18. ÉVSZAKOK

Kérdezd meg a hónap sorszámat, és írd ki, milyen évszakban van. Itt is legyen kiírva, ha érvénytelen sorszámu hónapot írunk be.

19. CELSIUS ↔ KELVIN ÁTVÁLTÓ

Készíts programot, mely bekér egy hőmérséklet értéket, majd felajánlja, hogy Celsiusból Kelvinbe, vagy Kelvinből Celsiusba váltja át. Készítsd el a programot úgy, hogy az több információt közöljön a felhasználóval arra nézve, hogy valójában mire képes!

Segítség:

- Az átváltáshoz használjuk a következő összefüggést:
 - $0\text{ C}^\circ = 273,15\text{ K}^\circ$
 - $0\text{ K}^\circ = -273,15\text{ C}^\circ$

A program egy-egy lehetséges kimenete:

```
Celsius <--> Kelvin átváltó program
Adj meg egy hőmérséklet értéket: 100
Válassz opciót: (1) C° --> K°
                (2) K° --> C° : 1
100 C° = 373,15 K°
```

```
Celsius <--> Kelvin átváltó program
Adj meg egy hőmérséklet értéket: 0
Válassz opciót: (1) C° --> K°
                (2) K° --> C° : 2
0 K° = -273,15 C°
```

20. ÁTMENT VAGY MEGBUKOTT?

Az osztály diákjainak jegyei alapján meg kell mondani, hogy sikeresen elvégezték-e az egyes tanulók a kurzust, vagy sem, és azt is, hogy a tanár adott-e valakinek tévedésből nem létező jegyet. A program 1 és 5 közötti tartományban fogadja el az osztályzatokat. Akkor ment át egy diák, ha jegye eltér 1-től.

21. MARADÉK 3-MAL OSZTVÁ

Írj programot, amely kiírja egy szám 3-mal vett maradékát! Ha nincs maradék, írjuk ki a következő szöveget: „Nincs maradék”. Ha van, írjuk ki a maradék értékét a „Maradék” szó után.