

# Telefonköltség

## Ismétlő kérdések

1. Mit jelent az algoritmusban a ciklus? Mikor alkalmazzuk?
2. Sorolja fel a ciklusok típusait!
3. Mi a szerepe a ciklusváltozónak?
4. Adjon össze a számológéppel 10 számot. Fogalmazza meg az összegezés algoritmusát!

Határozzuk meg, mennyibe kerül a telefonálás! Összegezzük eddigi költségeinket.

## Az összegezés algoritmusa

Összegezés  
összeg = 0  
CIKLUS AMÍG van még szám  
Be: szám  
összeg = összeg + szám  
CIKLUS VÉGE  
Ki: összeg  
Összegezés vége



Az összegezés algoritmusa

Mit jelent az összeg = összeg + szám értékadó utasítás? A számítógép előveszi az *összeg* nevű memóriaterületen tárolt mennyiséget, hozzáadja a szám értékét, majd az eredményt tárolja az *összeg* nevű memóriaterületen.

Hogyan jelezzük a számítógépnek, hogy véget ért az összeadásra kerülő számok sorozata? Utolsóként írunk be például 0-t. Ezt nyugodtan hozzáadhatjuk az összeghez, nem fog megváltozni. A 0 jelzi a sorozat végét.

Ciklus amíg szám  $\neq$  0

Sajnos a ciklus kezdetén még nem olvastunk be egyetlen számot sem, ezért a ciklus előtt adjunk valamilyen nem nulla kezdőértéket (például -1-et) a szám változónak! Mivel ezt az első beolvasás törölni fogja, így nem követünk el hibát az összeg kiszámításánál.

Összegezés  
összeg = 0  
szám = -1  
CIKLUS AMÍG szám  $\neq$  0  
Be: szám  
összeg = összeg + szám  
CIKLUS VÉGE  
Ki: összeg  
Összegezés vége



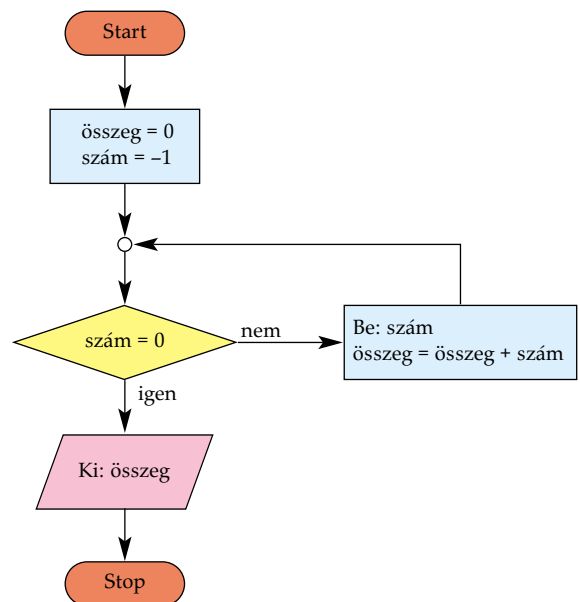
Az összegezés algoritmusa

```
while feltétel do
begin
utasítások;
end;

do while feltétel
utasítások
loop
```



A feltételes ciklus kódolása

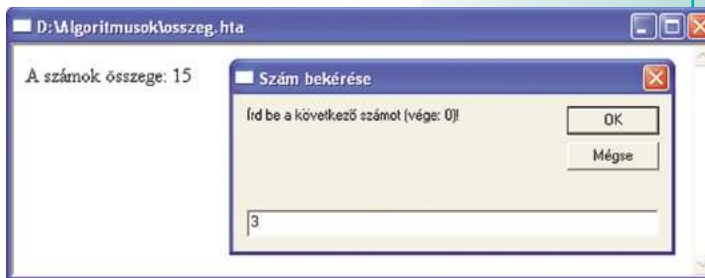
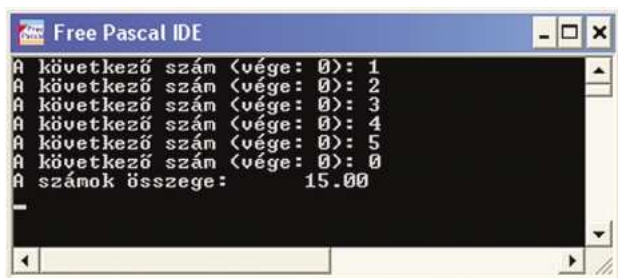


Az összegezés folyamatábrája

Figyeljünk arra, hogy a *do* kulcsszó után nincs pontosvessző. Ha a ciklus csak egyetlen utasítást tartalmaz, akkor a *begin/end* pár elhagyható.

Az összegezés algoritmusát az **osszeg** program mutatja be.





Az összeg program eredménye

## Elöl- és hátultesztelő ciklusok

Az összegzéshez használt ciklus az ismétlésre vonatkozó feltételt a ciklus végrehajtása előtt értékelte ki. Ha a feltétel hamis volt, akkor a ciklus utasításai egyszer sem kerültek végrehajtásra. Az ilyen ciklusokat előltesztelő ciklusoknak nevezzük.


**Elöltesztelő ciklus:** a végrehajtásra vonatkozó feltételt a ciklus elején értékeli ki. Ha a feltétel nem teljesül, a ciklus egyszer sem kerül végrehajtásra.

Az összegezés algoritmusában a ciklusba való belépéskor még nem ismerjük az első összeadandó értékét. Ezért jó lenne, ha az ismétlésre vonatkozó feltételt csak a ciklus végén vizsgálnánk meg. Az ilyen ciklusokat hátultesztelő ciklusoknak hívjuk.

**Hátultesztelő ciklus:** a végrehajtásra vonatkozó feltételt a ciklus végén értékeli ki. A ciklus utasításai egyszer mindenképpen végrehajtásra kerülnek.

Hátultesztelő ciklus használatakor nincs szükségünk az ismétlési feltételben szereplő változó értékének előzetes megadására.

Összegezés  
 összeg = 0  
 CIKLUS  
 Be: szám  
 összeg = összeg + szám  
 AMÍG szám ≠ 0  
 CIKLUS VÉGE  
 Összegezés vége





Összegezés hátultesztelő ciklussal

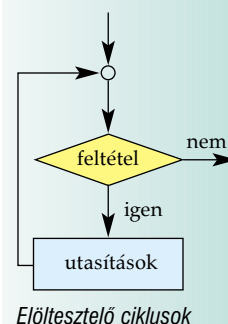
```

repeat
    utasítások;
until nem feltétel;

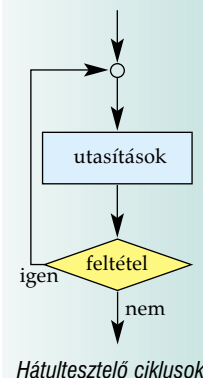
do
    utasítások
loop while feltétel
    
```

A hátultesztelő ciklus kódolása



Elöltesztelő ciklusok



Hátultesztelő ciklusok


Az összegezés algoritmusát hátultesztelő ciklussal a **hatul** program mutatja be.

## Kilépési és ismétlési feltételek

A feltételes ciklusok mondatszerű leírásában az ismétlés feltételét adtuk meg. Ha a feltétel igaz volt, akkor végrehajtottuk a ciklus utasításait, hamis értéke esetén pedig nem. Egyes esetekben könnyebben meg tudjuk fogalmazni a kilépésre vonatkozó feltételt. Ha a **kilépési feltétel** igaz, akkor nem hajtjuk végre a ciklus utasításait, hamis értéke esetén viszont ismételnünk.


- ismétlés, ha szám ≠ 0
- kilépés, ha szám = 0

CIKLUS AMÍG *ismétlési feltétel*  
 utasítások  
 CIKLUS VÉGE



Ciklus ismétlési feltétellel

CIKLUS MÍGNEM *kilépési feltétel*  
 utasítások  
 CIKLUS VÉGE



Ciklus kilépési feltétellel

Az algoritmusok mondatszerű leírásában az amíg szó helyett a mígnem szóval utalhunk a kilépési feltételre.

Vegyük észre, hogy az ismétlési és kilépési feltétel egymás tagadásaként fogalmazható meg:  $nem (ismétlési\ feltétel) = kilépési\ feltétel$

**A feltételes ciklusok tehát lehetnek elől- vagy hátultesztelő ciklusok, feltételként pedig megadhatunk ismétlési vagy kilépési feltételt.** Nem minden programozási nyelv rendelkezik mind a négyféle változattal.



Az előltesztelő ciklusba (a *while* után) csak ismétlési feltételt írhatunk, a hátultesztelő ciklusba (az *until* után) csak kilépési feltétel kerülhet.



A VBScriptben mind a négyféle feltételes ciklust használhatjuk:

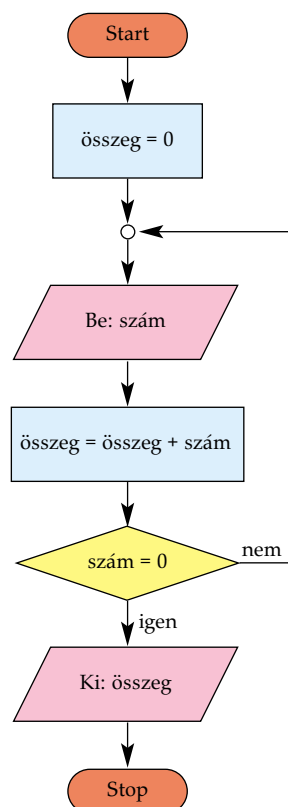
do while <i>ismétlési feltétel</i>	do
utasítások	utasítások
loop	loop while <i>ismétlési feltétel</i>
do until <i>kilépési feltétel</i>	do
utasítások	utasítások
loop	loop until <i>kilépési feltétel</i>



Az összegezés algoritmusát kilépési feltétellel a **kilépési** program mutatja be.

### Ellenőrző kérdések és feladatok

- Hogyan javíthatja ki az eredményt, ha hibás számot írt be, de a számítógép már hozzáadta az eddigi összeghez?
- Módosítsa az összegezés algoritmusának folyamatábráját mind a négyféle ciklusnak megfelelően!
- Írjon programot, amely beolvassa a telefonkártya értékét, bekéri és összeadja az eddigi költségeket, majd kiírja, hogy mekkora összeg maradt még a kártyán!
- Írjon programot, amely meghatározza az osztály diákjainak átlagos testmagasságát!
- Készítse el VBScriptben az **összeg** program változatát mind a négyféle ciklusra!



Összegezés hátultesztelő ciklussal

```

program kilepesi;
uses crt;
var szam, osszeg: real;
begin
  clrscr;
  osszeg:=0;
  repeat
  begin
    write ('Szám (vége: 0): ');
    readln(szam);
    osszeg :=osszeg+szam;
  end;
  until szam=0;
  writeln('Összeg: ', osszeg:10:2);
  readln;
end.
  
```



```

<script language="VBS">
  option explicit
  dim beolvas, szam, osszeg
  osszeg=0
  do
    beolvas=inputbox("Szám (vége: 0)!")
    szam=CSng(beolvas)
    osszeg=osszeg+szam
  loop until szam=0
  document.write "Összeg: ", osszeg
</script>
  
```



A **kilépési** program forráskódja