# Általános bevezető

## Elemi [programozási tételek](https://etananyag.szamalk-szalezi.hu/mod/book/view.php?id=13473):

* sorozatszámítás,

Ezekkel fogunk bővebben foglalkozni.

* megszámolás,
* maximum, minimum kiválasztás.
* eldöntés,
* kiválasztás,
* (lineáris) keresés,

## Összetett [programozási tételek](https://etananyag.szamalk-szalezi.hu/mod/book/view.php?id=13473) (később más tantárgyban fognak előfordulni):

* másolás,
* kiválogatás,
* szétválogatás,
* metszet,
* egyesítés (unió),
* összefuttatás.

### [Programozási tételek](https://etananyag.szamalk-szalezi.hu/mod/book/view.php?id=13473) jól használhatók - mert típusfeladatok, könnyen alkalmazhatók

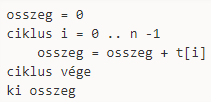
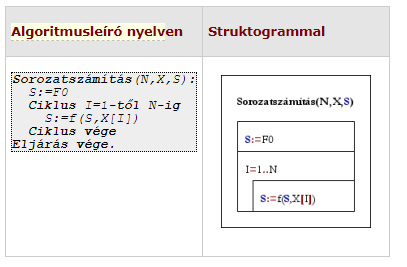
* problémamegoldás, algoritmizálás, programkészítés során.

### Programozási [feladatok](https://etananyag2.szamalk-szalezi.hu/mod/folder/view.php?id=14505) 4 típusa be- és kimenet alapján:

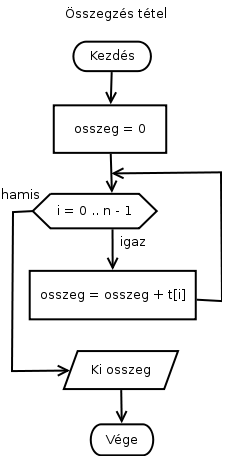
* 1 sorozathoz 1 értéket rendelő [feladatok](https://etananyag2.szamalk-szalezi.hu/mod/folder/view.php?id=14505),
* 1 sorozathoz 1 sorozatot rendelő [feladatok](https://etananyag2.szamalk-szalezi.hu/mod/folder/view.php?id=14505),
* egy sorozathoz több sorozatot rendelő [feladatok](https://etananyag2.szamalk-szalezi.hu/mod/folder/view.php?id=14505),
* több sorozathoz 1 sorozatot rendelő [feladatok](https://etananyag2.szamalk-szalezi.hu/mod/folder/view.php?id=14505).

Összegzés tétele

#### **Mondatszerű leírás:**



#### Folyamat ábra:



#### Tipikus összegzési feladatok

**F1.** Egy osztály N db tanuló osztályzatának ismeretében adjuk meg az osztály átlagát!

**F2.** Egy M elemű betűsorozat betűit fűzzük össze egyetlen szöveg típusú változóba!

**F3.** Készítsünk algoritmust, amely egy autóversenyző körönkénti ideje alapján meghatározza a versenyző egy kör megtételéhez szükséges átlagidejét!

**F4.** A Balaton mentén K db madarász végzett megfigyeléseket. Mindegyik megadta, hogy milyen madarakat látott. Készítsünk algoritmust, amely megadja a megfigyelések alapján a Balatonon előforduló madárfajokat!

**F5.** Adjuk meg az első N természetes szám szorzatát (N faktoriális)!

#### Kidolgozott minta példák:

#### Első N szám összege (ElsőNÖsszege1): A program írja ki az első N darab természetes szám összegét! Bemenetként ne fogadjon el negatív számot!

* + 1 sorozathoz (1-től n-ig) 1 értéket (összeget) rendelünk.
  + Számok összege előtt a gyűjtőváltozót 0-ra kell állítani.

Python programkód:  
  
n=beolvas()  
  
#összegzés tétele  
osszeg=0

ciklusValtozo=1

while ciklusValtozo<n+1:

osszeg+=ciklusValtozo

ciklusValtozo+=1

print("Az első ", n, "természetes szám összege: ", osszeg)

#### Faktoriális (Faktoriális1): A program írja ki egy szám faktoriálisát! Ne fogadjon el negatív számot bemenetként! n>=0, 0!=1, n!=1\*2\*3\*...(n-1)\*n.

Ha számokat kell összeszorozni, mindig 1-ről kell indítani a sorozatot tartalmazó változót.

Python programkód:

#beolvasás meghívása  
n=beolvas()  
  
#összegzés tétele  
faktorialis=1

ciklusValtozo=1

while ciklusValtozo<n+1:

faktorialis\*=i

ciklusValtozo+=1

print("Az ", n, "!= ", faktorialis)

1. Átlag1:Add meg az első n természetes szám átlagát! N éréke nem lehet negatív!

Python programkód:

#beolvasás meghívása  
n=beolvas()

#összegzés tétele, átlagszámítás1  
osszeg=0

ciklusValtozo=1

while ciklusValtozo<n+1:

osszeg+=ciklusValtozo

ciklusValtozo+=1

#kiiratás, 0db számra nincs értelme átlagot számolni  
if(n>0):  
 atlag = osszeg / n  
 print("Az első ", n, "természetes szám átlaga: ", atlag)  
else:  
 print("0 db számnak nincs átlaga.")

#### Átlag2: A program írja ki az első N darab természetes szám átlagát! Bemenetként ne fogadjon el negatív számot!

Python programkód:

#beolvasás meghívása  
n=beolvasa()  
#n=beolvasb()  
  
#összegzés tétele, átlagszámítás2  
  
atlag=0

ciklusValtozo=1

while ciklusValtozo<n+1:

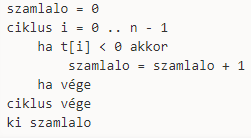
atlag=((i-1)\*atlag+i)/i

ciklusValtozo+=1

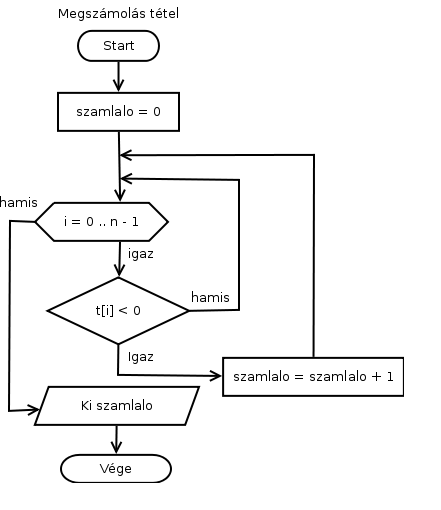
if(n>0):  
 print("Az első ", n, "természetes szám átlaga: ", atlag)  
else:  
 print("0 db számnak nincs átlaga.")

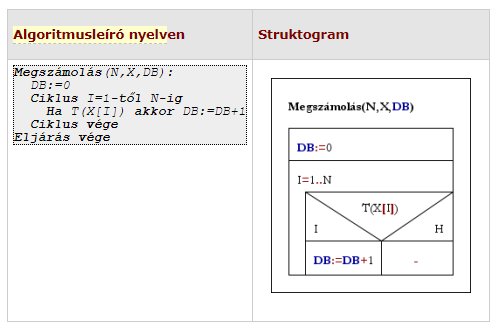
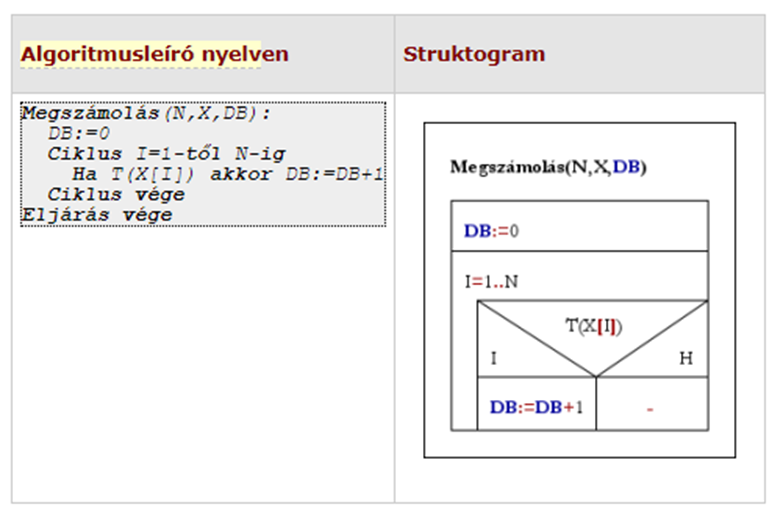
Megszámlálás

Mondatszerű leírás:



Folyamatábra:





#### Tipikus megszámlálási feladatok

**F16.** Családok létszámának, illetve jövedelmének alapján állapítsuk meg, hogy hány család él a létminimum alatt!

**F17.** Egy futóverseny végeredménye határozzuk meg, hogy a versenyzők hány százaléka teljesítette az olimpiai induláshoz szükséges szintet!

**F18.** Adjuk meg egy szöveg magánhangzóinak számát!

#### Kidolgozott minta példák:

1. A program számolja meg, hogy két beolvasott egész szám között(határokat is beleértve) hány páratlan van!  
   beolvasás:  
    #fájlból  
    szamom=beolvas()  
    szamom2=beolvas()  
     
   #megszámlálás  
   dbparatlan=0

ciklusValtozo=alsoHatar

while ciklusValtozo<felsoHatar+1:

if ciklusValtozo%2!=0:  
 print(ciklusValtozo)  
 dbparatlan+=1  
  
print("({},{}) között a páratlanok száma: {}".format(alsoHatar,felsoHatar,dbparatlan))

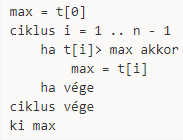
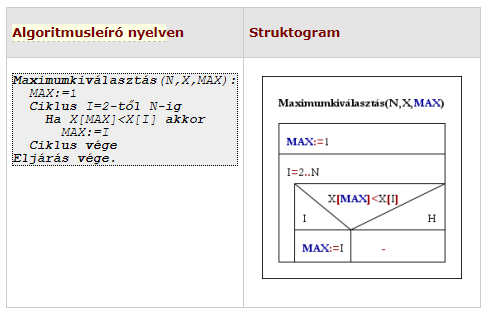
#### Kosáreredmények száma: Egy iskolai sportnapon kosárra dobó versenyt rendeztek és téged bíztak meg, hogy A: számold össze összesen hányszor dobott kosárra a 6 játékos, 1 pontérték 1 dobásnak számít! (pontértékek magyarázata: 0: nem talált bele a kosára, 3: távoli dobás 2:középtávoli dobás) 5:adatbevitel végét jelenti

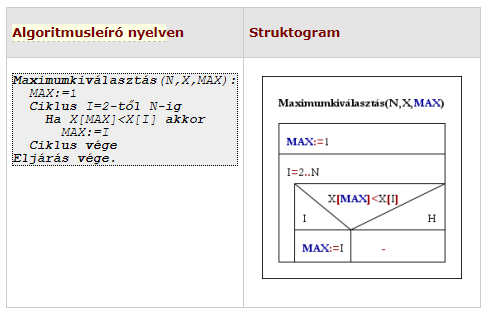
#véletlen generált  
  
eredmenyek=adatBeolvas()  
  
db=0  
while(eredmenyek!=5):  
 dbA+=1  
 eredmenyek=adatBeolvas()

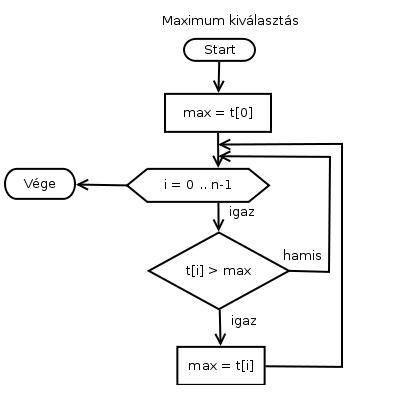
print("Kosára dobások száma: ", db)

MaximumKiválasztás

Mondatszerű leírás:





Folyamatábra: 

#### Tipikus maximumkeresési feladatok

**F19.** Egy kórházban megmérték minden beteg lázát, adjuk meg, hogy ki a leglázasabb!

**F20.** Egy család havi bevételei és kiadásai alapján adjuk meg, hogy melyik hónapban tudtak a legtöbbet megtakarítani!

**F21.** Egy osztály tanulói nevei alapján adjuk meg a névsorban legelső tanulót!

#### Kidolgozott minta példa:

#### Legnagyobb érték: Adott egy egy sorozat!(pl:: **-2, 8, 7, -12, -22, 19, -5, 4**) Keresse meg a legnagyobb érték? Hányadik érték ez? beAdat=beolvas()

#### maxe=elsoSzam

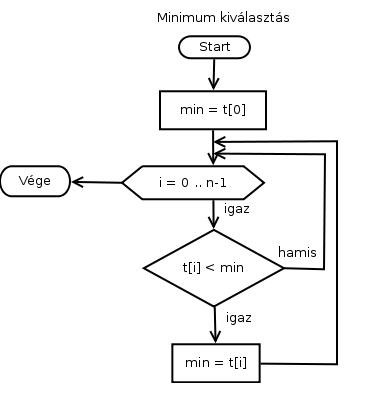
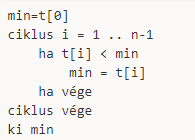
#### maxh=1

ciklusValtozo=2

adatokHossza=len(beAdat)  
while ciklusValtozo<adatokHossza:  
   
 if(aktuálisSzamErteke >maxe):  
 maxe = aktuálisSzamErteke  
 maxh = ciklusValtozo

ciklusValtozo+=1  
  
print("A legnagyobb elem: ", maxe, "helye: ", maxh)

Minimum kiválasztás



#### Tipikus minimumkiválasztásos feladatok

**F19.** Egy hűtőházban megmérték minden hús hőmérsékletét, adjuk meg, hogy ki a leghidegebb!

**F20.** Egy család havi bevételei és kiadásai alapján adjuk meg, hogy melyik hónapban volt a legkisebb abevételük!

**F21.** Egy osztály tanulói nevei alapján adjuk meg a névsorban legutolsó tanulót

#### Kidolgozott minta példa:

#### Legnagyobb érték: Adott egy egy sorozat!(pl:: **-2, 8, 7, -12, -22, 19, -5, 4**) Keresse meg a legnagyobb érték? Hányadik érték ez? beAdat=beolvas()

#### mine=elsoSzam

#### minh=1

ciklusValtozo=2

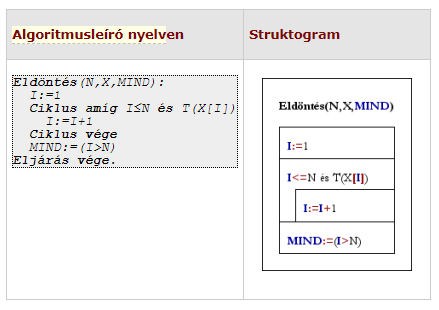
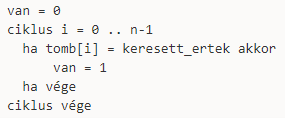
adatokHossza=len(beAdat)  
while ciklusValtozo<adatokHossza:  
   
 if(aktuálisSzamErteke <mine):  
 mine = aktuálisSzamErteke  
 minh = ciklusValtozo

ciklusValtozo+=1  
  
print("A legnagyobb elem: ", maxe, "helye: ", maxh)

További elemi programozási tételek érdeklődőknek:

Eldöntés tétele

#### Mondatszerű leírás:



#### Tipikus eldöntési feladatok

**F6.** Döntsük el egy számról, hogy prímszám-e!

**F7**. Döntsük el egy szóról a hónapnevek sorozata alapján, hogy egy hónap neve-e!

**F8.** Döntsük el egy tanuló év végi osztályzatai alapján, hogy kitűnő tanuló-e!

**F9.** Júniusban minden nap délben megmértük, hogy a Balaton Siófoknál hány fokos. Döntsük el a mérések alapján, hogy a víz hőfoka folyamatosan emelkedett-e!

#### Kidolgozott minta példa:

#### Prímszáme: Prímszám1: A program döntse el, hogy egy beolvasott egész szám prímszám-e!

Pozitív prímszámnak két osztója van: 1 és önmaga.  
0 és 1 nem prímszám.

Python kód:

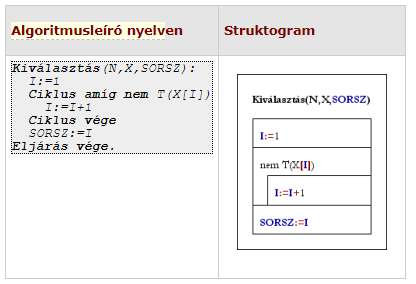
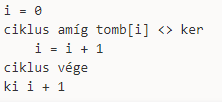
#beolvasás  
szam=beolvasEgesz()  
  
#eldöntés  
prim=True  
if szam<2:  
 prim=False  
else:  
 i=2  
 while(i<=sqrt(szam) and szam%i!=0):  
 i+=1  
 prim=i>sqrt(szam)  
  
#kiiratás  
if prim:  
 print("Prím.")  
else:  
 print("Nem prím.")

#### Páros-e: A program döntse el, hogy egy beolvasott egész szám páros-e!

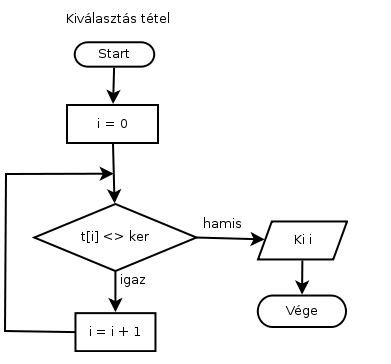
#beolvasás  
szam=beolvasEgeszTeljes()  
  
#eldöntés tétele  
parose=True  
if szam%2!=0:  
 parose=False  
  
#kiiratás  
if parose:  
 print("A {} páros.".format(szam))  
else:  
 print("A {} nem páros.".format(szam))

Kiválasztás

#### Mondatszerű leírás:



#### Folyamatábra:



#### Tipikus kiválasztási feladatok

**F10.** Ismerjük egy hónap nevét, a hónapnevek sorozata alapján mondjuk meg a sorszámát!

**F11.** Adjuk meg egy természetes szám legkisebb, 1-től különböző osztóját!

**F12.** A naptárban található névnapok alapján adjuk meg legjobb barátunk (barátnőnk) névnapját! (Itt nem a „legjobbság” megfogalmazása okozza az algoritmikai problémát.)

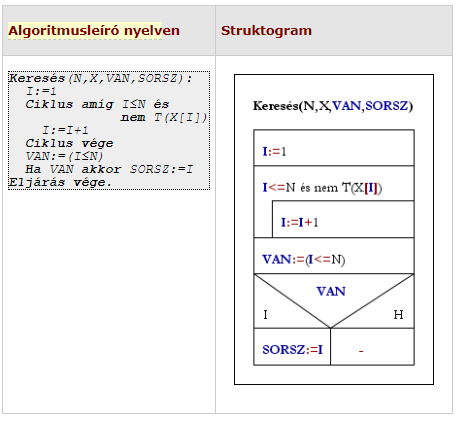
#### Kidolgozott minta példák:

#### 7-el osztható: Fájlból olvasd be az egész számokat! Szerepeljen benne 7-el osztható szám! Hányadik szám az első ami 7-el osztható? beolvas() db=0 talalat=False for sor in beFajl: szam=int(sor.strip()) db+=1 if szam%7==0 and (not talalat): hely=db talalat=True print("Az első 7-el osztható szám a: ",hely, "." )

#### Határok közötti: A fájlból olvasd be a számokat! Szerepeljen benne a 60 és 70 közé eső szám! Hányadik szám az első ami 60 és 70 közé esik? beFajl=beolvas() db=0 talalat=False for sor in beFajl: szam=int(sor.strip()) db+=1 if szam>60 and szam<70 and (not talalat): hely=db talalat=True print("Az első 60 és 70 közé eső szám a: ",hely, "." )

#### Hanyadik szám a 12: A felhasználótól kérd be hány számot szeretnél beolvasni, majd olvasd be a számokat! Szerepeljen benne a 12 érték egyszer! Hányadik szám a 12 a beolvasotak közül? szamokSzama=int(input("Add meg az adatok számát! ")) i=1 hely=-1 while (i<=szamokSzama): szam=int(input("{}. szám: ".format(i))) if szam==12: hely=i i+=1 print("A 12 érték ", hely, ". szám volt.")

lineáris keresés (Eldöntés és kiválasztás együtt.)



#### Tipikus lineáris kereséses feladatok

**F13**. Ismerjük egy üzlet egy havi forgalmát: minden napra megadjuk, hogy mennyi volt a bevétel és mennyi a kiadás. Adjunk meg egy olyan napot **–** ha van **–**, amelyik nem volt nyereséges!

**F14.** A Budapest–Nagykanizsa vasúti menetrend alapján két adott állomáshoz adjunk meg egy olyan vonatot, amellyel el lehet jutni átszállás nélkül az egyikről a másikra!

**F15.** Egy tetszőleges (nem 1) természetes számnak adjuk meg egy osztóját, ami nem az 1 és nem is önmaga.

#### Kidolgozott minta példák:

#### 3-al osztható: Adott egy fájlban egy sorozat! Van-e benne 3-al osztható érték? Ha igen hol? beFajl=beolvas() van=False poz=0 for sor in beFajl: szam=int(sor) poz+=1 if szam%3==0 and not van: hely=poz van=True if van: print("Hárommal osztható érték a", hely, "helyen van.") else: print("Nincsne hárommal osztható érték.")

#### Azonos szomszédok: Adott az 5, 4, 3, 4, 5, 4, 4, 5, 3, 1 sorozat. Van-e benne egymás mellett két azonos szám? Ha igen, hol? beFajl=beolvas() oadat=beFajl.read() tAdatLista=oadat.split(", ") szamosadat=list(map(int, tAdatLista)) van=False for i in range(1,len(szamosadat)): if szamosadat[i-1]==szamosadat[i] and not van: van = True helye = str(i)+", "+str(i+1) #print(szamosadat) if van: print("A", hely, "helyen lévő elemek egyeznek meg.") else: print("Nincsenek megegyező szomszédos elemek.")