

Források:

[https://www.nkp.hu/tankonyv/digitalis\\_kultura\\_9\\_nat2020/](https://www.nkp.hu/tankonyv/digitalis_kultura_9_nat2020/)

lecke\_05\_001

<https://sulipy.hu/>

## Tartalomjegyzék

1. Alapfogalmak.....	1
2. A programozás építőkövei:.....	1
2.1. Vezérlési szerkezetek:.....	2
2.2. Adatszerkezetek:.....	2
2.2.1. Változók:.....	2
2.3. Utasítások, szekvenciák.....	3
2.4. Szelekció – azaz feltételes elágazás.....	5
2.4.1. Kétágú elágazás.....	6
2.4.2. Többágú elágazás.....	6
2.5. Gyakorló feladat készítése:.....	7
2.6. Ciklusok.....	8
2.6.1. A számláló típusú ciklus.....	8
2.6.2. Példa a számláló ciklusra:.....	9
2.6.3. Feltételes ciklus:.....	10
2.6.4. Jelszó megadásának modellálása – Hozzáférés csak cuncimókusnak.....	11

### 1. ALAPFOGALMAK

**A program olyan egyszerű utasítások, műveletek logikus sorozata, amelyekkel a számítógépet irányítjuk.**

**számítógépes program azon utasításoknak a sorozata, amelyeket a számítógépnek egymás után végre kell hajtania valamely feladat elvégzése céljából, jellemző módon azt, hogy az adatokkal milyen műveleteket végezzen.**

A programokat háttértárolón tároljuk, ha éppen nem futnak. Ha egy programot elindítunk, az operációs rendszer a háttértárolóról betölti a programot a memóriába. A CPU számára átadja a program kezdetének címét, majd a program ezután átveszi a számítógép vezérlését és futni, működni kezd.

**Algoritmuson vagy eljáráson** olyan megengedett lépésekből álló módszert, utasítás(sorozat), részletes útmutatást, receptet értünk, amely valamely felmerült probléma megoldására alkalmas. (Megoldási terv, mely előírja, hogy meghatározott lépéseket adott sorrendben kell elvégezni.) Pl.

*Rántottakészítés algoritmus (megbeszélni!)*

## 2. A PROGRAMOZÁS ÉPÍTŐKÖVEI:

## 2.1. VEZÉRLÉSI SZERKEZETEK:

Egy program algoritmusa **háromféle** típusú **vezérlési szerkezet** segítségével írható le: )

Utasítás: A végrehajtandó feladat. Pl. *Kapcsold be a számítógépet!*

**A szekvencia** :egyszerű tevékenységek, **utasítások sorozata**. Pl. *Kapcsold be a számítógépet, írd be a felhasználónevedet, add meg a jelszavadat!*

**Szelekció: feltétel(ek)től függő tevékenység(ek)** végrehajtása, több irányban folytatva az előrehaladást. (Elágazás, feltételes elágazás) Pl. Elme-  
gyek a vásárolni. *Ha esik az eső, esernyőt viszek,és gyalog megyek; ha nem esik, biciklivel megyek.* Irodalomban: „Ha még egyszer azt üzeni, mindnyájunknak el kell menni...”

**Iteráció:** ismétlési szerkezet, elemi **utasítások ismétlődő végrehajtása**. Van olyan ciklus, amelyik **feltétel** alapján történik; van olyan, amely **meghatározott számú** alkalommal történik. (**Ciklus** )

Feltételes ciklus például: *Amíg jól nem laktam, addig ebédelek. Addig kortyolom a hideg sörömet, amíg józan vagyok.*

Számláló ciklus például : *Már ötször mondtam el, mit kell csinálnod, fiam! Százszor leírod, hogy az órán a tanuló figyel!*

## 2.2. ADATSZERKEZETEK:

Alapvetően **kétfajta** adattípus van: az **állandó( konstans) adatok** – értékük a program futása során nem változik – és a **változók** – a változó értékét felül lehet írni, illetve a program futása közben megváltoztathatjuk- . Konstans például a gravitációs állandó a Földön, változó az ember életkora.

## 2.2.1. VÁLTOZÓK:

A programjainknak gyakran kell adatokat tárolniuk. Az adatokat a gép memóriájába teszik el. Az eltett adatokat úgy tudjuk ismét elővenni, ha megadjuk azt a nevet, amit az eltett adathoz hozzárendelünk. Ezt a hozzárendelt nevet változónak hívjuk, és az eltárolást programozóul úgy mondjuk, hogy értéket adunk a változónak. (Az értékadás egy művelet, ugyanúgy, mint az összeadás vagy az osztás, és van műveleti jele is, ami sok programozási nyelvben – a Pythonban is – az egyenlőségjel.)

pl.

<b>X = input(„Add meg a neved!”)</b>	Jelentése: Létrehoztam egy X nevű változót,
	Melynek értékét a felhasználó adja meg.

Az adatok típusai:

- egész számok (**integer**): Egész számokat tudunk tárolni ilyenkor. A számokkal számolni tudunk, tehát értelmetlen karaktereket elosztani egy számmal.

Fontos: Ha a felhasználó a gépbe valamit beír, azt a Python karakterként értelmezi, azaz a típusa <str> lesz. Ha számolni akarok egy számmal, akkor először közvetlenül a beolvasás (input) után át kell alakítani számmá. Ennek menete, ha a változót B néven hoztuk létre:

**B = int(input(„Melyik évben születél?:”))**

- karaktersorozat (**string**) : Ez a típus szövegek vagy egy betű tárolására szolgál. A karaktersorozatot mindig idézőjelek ( „”) vagy aposztrófok ( ’ ’ ) közé kell tenni.

Például így íratunk ki egy karaktersorozatot:

**print(„Ezzel az adattípussal értelmetlen számolni.”)**

- tört szám (**float**): Ha tört számokkal kell számolnunk, akkor azt az adattípust használjuk. A kör területének vagy kerületének kiszámolásához a Pi értékét kell használni (3.14 ), akkor használjuk a float típust.

Ha például úgy osztunk két számot, hogy maradéka maradjon, akkor a python automatikusan float adattípussá változtatja a számot.

Példaprogram a float adattípusra:

a=5	Létrehoztam egy a nevű változót, melynek értéke 5 lesz.
b=2	Létrehoztam egy b nevű változót, melynek értéke 5 lesz.
c=a/b	Létrehoztam egy c nevű változót, melynek értéke az a és a b hányadosa lesz.
print(c)	Kiíratom a c nevű változó értékét a konzolra.
print(type(c))	Kiíratom a c változó adattípusát a konzolra. A kiírás eredménye a float szó lesz.

### 2.3. UTASÍTÁSOK, SZEKVENCIÁK

Utasítás az tehát, hogy megmondom a gépnek, mit tegyen. Szekvencia utasítások sorozata. Például:

<code>print(„A program kiszámolja az áfa összegét.”)</code>	Egyszerű utasítás – A szöveget kiíratom a konzolra.
<code>x=int(input(„Add meg a születési évét !: »))</code>	Ez szekvencia – x nevű változóba bekérjük a felhasználó születési évét, és a beírt szöveget azonnal számmá alakítjuk át.
<code>kor=2024-x</code>	Létrehozunk egy kor nevű változót, mely az adott évből kivonja a x változó tartalmát.
<code>print(kor)</code>	Kiíratom a kor változó tartalmát, azaz a felhasználó életkorát.

### 2.4. SZELEKCIÓ – AZAZ FELTÉTELES ELÁGAZÁS

Bizonyos esetekben a program többfelé ágazhat, azaz futása az adott feltétel teljesítése esetén eltérő módon történhet. Például: *Ha elfogyott a söröm, elmegyek a boltba a szomszéd kedvenc sörjét beszerezni.*

A python nyelvben ehhez az **if**, **elif**, **else** szavakat használjuk fel. Az **utasításokat** a python szabályai szerint **4 helyel beljebb kell kezdeni**. A feltételvizsgálat után **mindig kettőspontot** írunk!

Legalább két része van a feltételes elágazásnak:

**if ( feltételvizsgálat):**

....utasítás1

**else:**

....utasítás2

De lehetséges az is, hogy több része van az elágazásnak:

**if ( feltételvizsgálat):**

utasítás1

**elif ( feltételvizsgálat):**

utasítás2

**else:**

....utasítás3

## 2.4.1. KÉTÁGÚ ELÁGAZÁS

Ez esetben a program kétfajta módon folytatódhat. Lássunk rá példát.

K= input(„Add meg a jelszót! :”)	Egy K nevű változóba letároljuk a felhasználó által beírt szót.
If K == „cuncimókuska”:	Megvizsgáljuk ( == jelenti az egyenlőségvizsgálatot), hogy K változóban a cuncimókuska szó van-e
....print(„Szia, édesem!”)	Az if – azaz a feltételvizsgálat – után 4 hellyel beljebb írunk, kiírjuk a Szia, édesem!-szöveget. Ez a <b>feltételes vizsgálat igaz ága</b> .
else:	A sor elejére megyünk vissza, majd az else szóval jelezzük, hogy a <b>hamis ág</b> következik
....print(„Ön eszerint nem a feleségem!”)	A <b>hamis</b> ágat is 4 hellyel beljebb kezdjük.

## 2.4.2. TÖBBÁGÚ ELÁGAZÁS

A fenti példát kiegészítjük a gyermek megjelenésével:

K= input(„Add meg a jelszót! :”)	Egy K nevű változóba letároljuk a felhasználó által beírt szót.
If K == „cuncimókuska”:	Megvizsgáljuk ( == jelenti az egyenlőségvizsgálatot), hogy K változóban a cuncimókuska szó van-e
....print(„Szia, édesem!”)	Az if – azaz a feltételvizsgálat – után 4 hellyel beljebb írunk, kiírjuk a Szia, édesem!-szöveget. Ez a <b>feltételes vizsgálat igaz ága</b> .
elif K==”kicsilányom”:	A sor elejére megyünk vissza, majd az <b>elif szóval jelezzük, hogy egy másik vizsgálatot tartunk</b> .
....print(„Szia, leányom!”)	Ezt az ágat is 4 hellyel beljebb kezdjük.
else:	A sor elejére megyünk vissza, majd

	az else szóval jelezzük, hogy a hamis ág következik
....print(„Ön eszerint nem hozzátartozóm!”)	A hamis ágot is 4 hellyel beljebb kezdjük.

The screenshot shows the 'ONLINE PYTHON BETA' web interface. At the top, there are navigation links for 'programoz', 'KRETA', 'Tőzsdeárnyék', and 'MARKETWATCH'. Below the header, there's a toolbar with icons for file operations. The main area displays a Python script in a file named 'Untitled2.py'. The script is as follows:

```

1 K=input("Add meg a jelszót! :")
2 if K=="cuncimókuska":
3     print("Sza, édesem!")
4 elif K=="kicsilányom":
5     print("Szia, leányom!")
6 else:
7     print("Ön eszerint nem hozzátartozóm!")
8

```

Below the code editor, there are buttons for 'Run' and 'Share', and a field for 'Command Line Arguments'. The output window shows the execution results:

```

Add meg a jelszót! :
cuncimókuska
Sza, édesem!

** Process exited - Return Code: 0 **
Press Enter to exit terminal

```

## 2.5. GYAKORLÓ FELADAT KÉSZÍTÉSE:

Írj egy olyan programot, ahol a gép számokat kér be egy és hét között (1-7)!

A megfelelő – tehát a szám egy és hét között van – szám esetén a konzolra írasd ki a hét megfelelő napját ( például: 1 beírása esetén a gép a hétfő szót írja ki), s ha a szám nem egy és hét között van, akkor azt írasd ki, hogy: „Ön nem megfelelő számot adott meg!”.

## 2.6. CIKLUSOK

A ciklus, vagy **iteráció** a számítógép-programozás és az algoritmusok egyik alapvető eszköze, amely az ismétlődő (azonos vagy hasonló) tevékenységek megvalósítására szolgál. Egy olyan *programrészletet* értünk rajta, amely valahányszor megismétlődik.

A Pythonban kétféle ciklus létezik: a feltételes ciklus és a számláló típusú.

## 2.6.1. A SZÁMLÁLÓ TÍPUSÚ CIKLUS

A számláló ciklusban az utasítások meghatározott számszor hajtódnak végre. Pl. **Háromszor** *veri ezt kenden Ludas Matyi vissza!* (Fazekas Mihály)

Ezt a ciklust a **for** és a **range**(terjedelem, meddig is?) szavakkal valósítjuk meg.

The screenshot shows the 'ONLINE PYTHON BETA' web interface. The code editor contains a Python script in 'main.py':

```
1 for a in range(3):
2     a += 1
3     print(a, " verés")
```

Below the code editor, there are buttons for 'Run' and 'Share', and a field for 'Command Line Arguments'. The output area shows the result of running the code:

```
1 verés
2 verés
3 verés
```

At the bottom, a status message indicates: '\*\* Process exited - Return Code: 0 \*\* Press Enter to exit terminal'.

1. Ábra: a számláló - vagy lépéses ciklus

for a in range(3):	A ciklus <b>feje</b> : az <b>a</b> változó a <i>ciklus forgásának számát</i> mutatja ( <b>ciklusváltozó</b> ) Jelen esetben 3-szor ismételjük meg a ciklusmagban lévő utasítást.
....a += 1 / vagy így is jó: a=a+1	A <b>ciklusmag</b> : itt két utasítás található. A ciklus utasítását is 4 <i>karakterrel beljebb</i> kell kezdeni. Ebben a sorban az <b>a változó értékét növeljük eggyel</b> .
....print(a, " verés")	A ciklus belsejében 3-szor ismételjük az előző utasítást. (Háromszor íratjuk ki, hogy 1-2-3- verés). Az a <b>változó</b> szerepe az, hogy a <b>ciklus pörgésének számát</b> írjuk ki.

## 2.6.2. PÉLDA A SZÁMLÁLÓ CIKLUSRA:

Írjunk egy programot, amely megszámolja, hány karakter van egy adott szóban. Az adott szó a **hatodik** szó.

Mi is az algoritmus? (Terve?) Kell létrehozunk egy ciklusváltozót (számlálót, zeiger), egy számot, amely megmutatja, hányadik betűnél vagyunk.

Számláló (ciklusváltozó) <b>cv</b>	1	2	3	4	5	6	7
A szó betűi	h	a	t	o	d	i	k

The screenshot shows an online Python IDE interface. At the top, there's a blue header with the Python logo and 'ONLINE PYTHON BETA'. Below it are icons for file operations. The main area shows a file named 'main.py' with the following code:

```
1 szo="hatodik"
2 for cv in range(len(szo)):
3     cv += 1
4 print(cv, " betűből áll a szó.")
```

Below the code editor, there are buttons for 'Run', 'Share', and 'Command Line Arguments'. The output area shows the result of running the code: '7 betűből áll a szó.' followed by a message indicating the process exited successfully.

Az 1. sorban a **szo** nevű **változóba** tároltuk le a szót.

A második sorban a **cv** változót indítjuk el: **nullától** számol a **szo** változóban tárolt érték **hosszáig** (`len(szo)`). [A **len()** kifejezés jelentése: a zárójelben levő változó hosszaiig – ahány betű van a szóban]

A harmadik sor jelenti azt, hogy a **cv változó értékét növeljük eggyel**, azaz jön a ciklus következő köre – első alkalom, második alkalom, harmadik..... a **szo** változó hosszaiig.

A negyedik sor már a ciklus lefutása után következik: Miután a **cv** értéke elérte a **szo** változó hosszát ( 7 betűhossz), a 4. sorban **kiíratom a cv változót**.



## 2.6.3. FELTÉTELES CIKLUS:

A **feltételes ciklus** olyankor használatos, amikor nem tudjuk előre, hogy hányszor kell a ciklusnak lefutnia. Az viszont ismert, hogy milyen feltétel teljesülése esetén kell a programnak tovább futnia vagy leállnia.

A ciklus „pörgése” tehát nem egy konkrét számhoz, hanem feltételhez van kötve.

A köznyelvben ismert mondás: „*Addig jár a kórsó a kútra, amíg el nem törik.*” Tehát a kórsó folyamatosan, többször, ciklikusan jár a kútra, de ha eltört, már nem jár. Ekkor befejezte a járási folyamatot.

A feltételes ciklus a *while* szóval valósul meg.

Általánosan:

**while** *amíg a feltétel igaz:*

....utasítás(1)

....utasítás(n)

utasítás vagy feltételvizsgálat

The screenshot shows the 'ONLINE PYTHON BETA' web interface. The code editor displays a Python script named 'main.py' with the following content:

```

1 valasz="igen"
2 while valasz=="igen":
3     valasz=input("Ép vagy még? :")
4     if valasz=="igen":
5         print("Akkor még járj a kútra!")
6     print("Mivel már nem vagy ép, többet már nem jársz a kútra!")

```

Below the code editor, the 'Run' button is highlighted. The output console shows the following interaction:

```

Ép vagy még? :
igen
Akkor még járj a kútra!
Ép vagy még? :
igen
Akkor még járj a kútra!
Ép vagy még? :
nem
Mivel már nem vagy ép, többet már nem jársz a kútra!

```

Ez a kis program a fenti szólás megvalósítása. A felhasználónak az igen vagy a nem szót kell beírni. Amennyiben az igen szót gépeli be, a program az *Akkor még járj a kútra!*-mondattal engedi a további járást. Ha a nem szót írja be a felhasználó, a program kilép a ciklusból, és a program futása a ciklus után, azaz a 6. sorban folytatódik a *Mivel már nem vagy ép, ..* - sor kiírásával.

A **ciklus fejében**( 2. sor) azt a **vizsgálatot** tesz-

szük meg, hogy a **felhasználó az igen szót adta-e meg**, s ha ez igaz, akkor a ciklus magja (3-5sor) hajtódik végre. Ha a felhasználó a **nem** szót adta meg, akkor a ciklus fejében a **vizsgálat**( igent írt-e be a felhasználó?) **hamissá válik**, tehát a ciklus futása befejeződik, és a ciklus utáni első utasítást hajtja végre a gép.

## 2.6.4. JELSZÓ MEGADÁSÁNAK MODELLÁLÁSA – HOZZÁFÉRÉS CSAK CUNCIMÓKUSNAK

Valósítsuk meg azt a modellt, hogy a gép csak annak engedi a belépést, aki a cuncimokus jelszót írja be.

The screenshot shows the Online Python Beta IDE interface. The code editor contains the following Python code:

```
main.py
1  valasz:str=""
2  while valasz != "cuncimokus":
3      valasz=input("Add meg a jelszót! :")
4  print("Kedves Felhasználó, most már hozzáférhetsz a ki
```

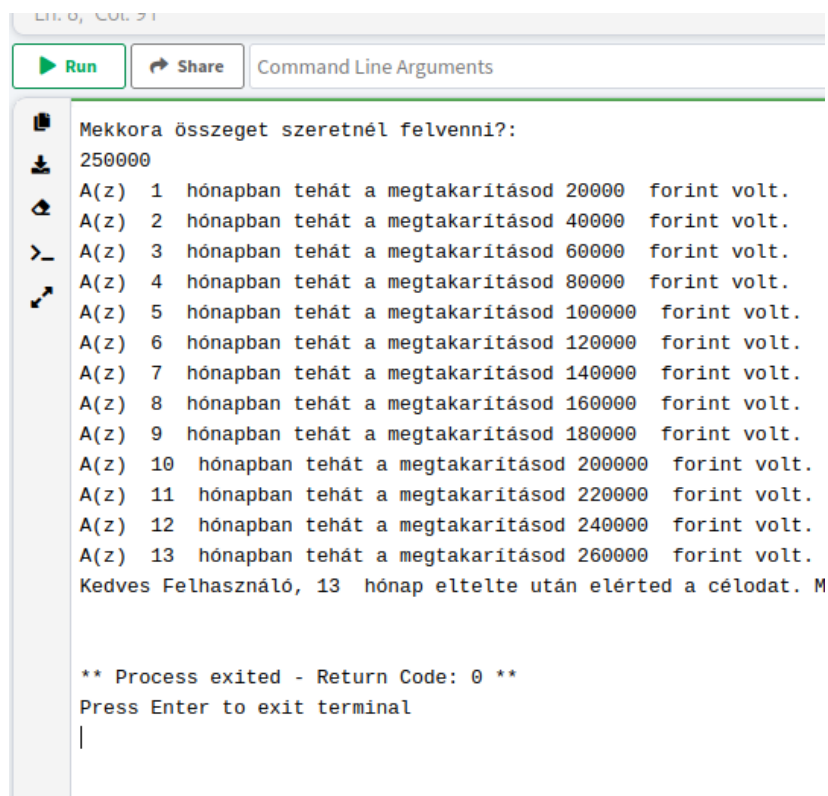
Below the code editor, the 'Run' button is highlighted. The output console shows the following interaction:

```
Add meg a jelszót! :
dhlf
Add meg a jelszót! :
dfd
Add meg a jelszót! :
dfdf
Add meg a jelszót! :
dgfdgd
Add meg a jelszót! :
cuncimokus
Kedves Felhasználó, most már hozzáférhetsz a kincseimhez!
```

<code>valasz:str=""</code>	Elsőként <b>létrehozom a változót</b> , amelyben tárolom a felhasználó által megadott karaktersorozatot. Az értékét üresre állítom a idézőjelek beírásával.
<code>while valasz !="cuncimokus":</code>	<b>A ciklus feje.</b> A <i>feltétel</i> az, hogy <i>addig „pörögjön a ciklus, amíg a valasz változó nem lesz egyenlő a cuncimokus szóval.</i> Tehát a <i>feltétel</i> az, hogy:”A valasz nem egyenlő a cuncimokus szóval?” - Amíg ez igaz (tehát nem egyenlő), addig a ciklus magja végrehajtódik. Ha a valasz tartalma a cuncimokus karaktersorozat, akkor a feltétel hamis lesz, s így a ciklus futása megszakad.
<code>valasz=input("Add meg a jelszót! :")</code>	<b>A ciklus magja.</b> Egyetlen utasítást tartalmaz: a valasz változóban tárolja le a felhasználó által megadott karaktersorozatot.
<code>print("Kedves Felhasználó, most már hozzáférhetsz a kincseimhez!")</code>	A ciklus utáni első sor – a cikluson kívül vagyunk. Ezt a sort csak akkor mutatja gép, ha kiléptünk a ciklusból, mert a ciklus feltételvizsgálata hamis értéket adott. Jelen esetben a valasz változó tartalma a cuncimokus karaktersorozat lett.

## 2.6.5. GYAKORLÓ FELADAT KÉSZÍTÉSE

Készíts egy olyan programot, mely kiszámolja, hogy havi húszezer forint félretételével hány hónap alatt éred el az általad megadott összeget! Például: Szeretnél hozzájutni 250.000 Ft-hoz. Hány hónapig kell havi 20.000 Ft-ot félretenned, hogy elérd a 250.000 Ft összeget?

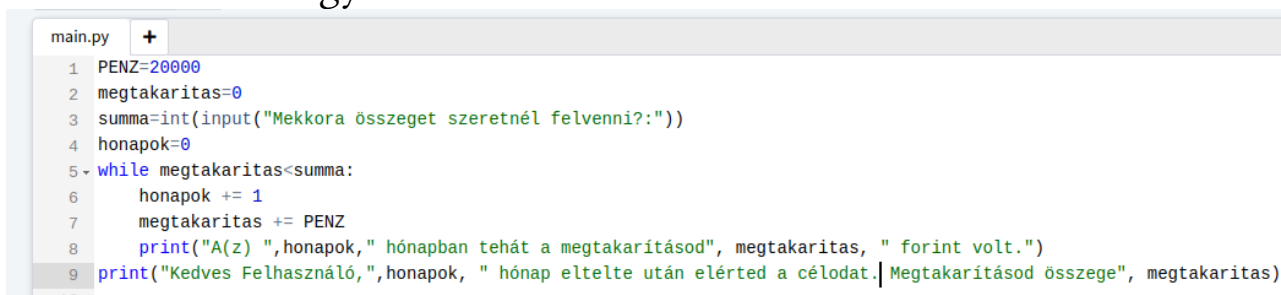


```
Run Share Command Line Arguments

Mekkora összeget szeretnél felvenni?:
250000
A(z) 1 hónapban tehát a megtakarításod 20000 forint volt.
A(z) 2 hónapban tehát a megtakarításod 40000 forint volt.
A(z) 3 hónapban tehát a megtakarításod 60000 forint volt.
A(z) 4 hónapban tehát a megtakarításod 80000 forint volt.
A(z) 5 hónapban tehát a megtakarításod 100000 forint volt.
A(z) 6 hónapban tehát a megtakarításod 120000 forint volt.
A(z) 7 hónapban tehát a megtakarításod 140000 forint volt.
A(z) 8 hónapban tehát a megtakarításod 160000 forint volt.
A(z) 9 hónapban tehát a megtakarításod 180000 forint volt.
A(z) 10 hónapban tehát a megtakarításod 200000 forint volt.
A(z) 11 hónapban tehát a megtakarításod 220000 forint volt.
A(z) 12 hónapban tehát a megtakarításod 240000 forint volt.
A(z) 13 hónapban tehát a megtakarításod 260000 forint volt.
Kedves Felhasználó, 13 hónap eltelte után elérted a célodat. M

** Process exited - Return Code: 0 **
Press Enter to exit terminal
|
```

Esetleg kíváncsi vagy a megoldásra? Nézd meg, és mondd el, hogy mit csinálnak az egyes sorok!



```
main.py +
1 PENZ=20000
2 megtakaritas=0
3 summa=int(input("Mekkora összeget szeretnél felvenni?"))
4 honapok=0
5 while megtakaritas<summa:
6     honapok += 1
7     megtakaritas += PENZ
8     print("A(z) ",honapok," hónapban tehát a megtakarításod", megtakaritas, " forint volt.")
9 print("Kedves Felhasználó,",honapok, " hónap eltelte után elérted a célodat. | Megtakarításod összege", megtakaritas)
```