



Programozás
gyakorlat – 05. hét

October 3

2022

Iteráció: Előltesztelési ciklus, Hátultesztelési ciklus, Véletlenszámok

Feladatlap

Iterációk

A program legfontosabb tulajdonságai közé tartozik, hogy képes utasításokat ismételten végrehajtani. Az iteráció ismétlést jelent, ilyenkor egy vagy több utasítás újra és újra végrehajtódik.

Három mechanizmust kínál a python nyelv az iterációk megvalósítására:

- elől tesztelő ciklus: a belépési feltételét fogalmazzuk meg. `while`
- hátul tesztelő ciklus: a bennmaradás feltételét fogalmazzuk meg.
- léptető ciklus: a ciklusváltozó minden egyes ciklus végrehajtásakor automatikusan lép egyet. `for`

Elöl tesztelő Ciklus – While

A program a ciklusba való belépés előtt megvizsgál egy feltételt – ezt belépési feltételnek nevezzük.

- Ha a belépési feltétel teljesül, akkor a ciklusmag végrehajtásra kerül, egyébként nem.
- Amíg a belépési feltétel teljesül, a ciklusmag ismételten végrehajtódik.
- Ha már a belépési feltétel nem teljesül, akkor a program a ciklus utáni utasítással folytatódik.

Tipikusan olyan feladatok megoldására javasolható, amelyekben az induló feltételek határozzák meg a ciklusmag végrehajtásának szükségességét.

Elöl tesztelő ciklus kódja

`while` feltétel:
utasítás

`while` feltétel:
utasítás1
utasítás2
...
utasításN

Véletlen számok generálása

Ahhoz, hogy véletlen számokat tudjunk generálni, használnunk kell a `random` modult. A `math` modul használatát követően megtanulták, hogy ahhoz, hogy használni tudjuk, importálnunk kell. Ezután tudjuk felhasználni az abban a modul irt metódusokat.

`import random` – a `random` modul importálása-

`random.randint(also határ, felső határ)` egész számokat generál. Két paramétert vár: mettől meddig generálja a számokat.

`random.randint(0,3)` → a lehetséges értékek: 0,1,2,3

`random.random()` – 0-1 közötti float számot generál

1. Feladat

Generáljunk véletlen számokat, és írassuk is ki őket:

[0,10] között

[10,15]

[0,100]

[-100,100]

10 db 0-1 közötti véletlen számot

4 db -10 30 közötti véletlen számot

2. feladat

Írj programot, amivel fej vagy írás játékot lehet játszani.

Először egy a képernyőre kiírt menü segítségével kérjük be a felhasználó tippjét, ami legyen egy egész szám. A 0 fejet, az 1 írást jelentsen.

Ezután generáljunk egy véletlen egész számot a $[0,1]$ azaz $[0,2[$ intervallumból, ez lesz a pénzfeldobás eredménye.

Ezt írjuk is ki a képernyőre, majd írjuk ki, hogy a felhasználó jól tippelt-e. Ha igen, gratuláljunk neki, ha nem, bíztassuk, hogy játsszon újra.

3. Feladat

Készítsünk programot, amely bekér egy egész számot (1-től 100-ig), majd kiírja az adott számot betűkkel.

- A szó kiírásához előbb nézzük meg hogy a szám tízzel osztható-e, ha igen, akkor írjuk ki: tíz, húsz, harminc, stb.
- Ha a szám nem osztható tízzel, nézzük meg mi áll a tízesek helyén a számban (div függvénnyel) és ez szerint előbb írjuk ki, hogy: tízen, húszon, harminc, stb. (ha nulla van a tízesek helyén, akkor semmit ne írjunk ki).
- Végül nézzük meg hogy mi áll az egyesek helyén (mod függvénnyel) és ez alapján írjuk ki mellé egy másik elif segítségével, hogy: egy, kettő, három, stb.

4. Feladat

Írjunk programot, ami bekér a felhasználótól egy számot. Ennek a számnak az értékét addig szorozzuk bekért értékekkel, amíg a felhasználó nullát nem ad.

5. Feladat

Írjunk programot, ami az "alma" szóhoz hozzáfűz további szavakat szóközzel elválasztva, egészen addig, amíg a felhasználó üres Stringet (entert) nem ad.

6. Feladat

Írjunk ki a képernyőre bekért darabszámú, bekért karaktert!

7. Feladat

Kérjünk be két egész számot! Írjuk ki a számokat az első számtól a másodikig! Figyeljünk arra, hogy nem biztos, hogy az első szám a kisebb! Oldjuk meg a feladatot úgy is, hogy kérjük be a lépésközt is!

8. Feladat

Határozzuk meg és írassuk ki az összes hárommal és öttel egyaránt osztható, 1000-nél kisebb természetes számot

9. Feladat

Készítsünk programot, amely bekér egy összeget (minimum 1, maximum 1000 Ft), majd kiírja, hogy azt hogyan lehet a lehető legkevesebb pénzértméből összeállítani.

10. Feladat

Készítsünk programot, mely meghatároz egy számtani sorozatot (adott az első elem, a differencia és az elemszám).

11. Feladat

Kérjünk be egy N pozitív egész számot, majd írjuk ki az N -nél nem nagyobb pozitív páratlan számok összegét!

12. Feladat

Határozzuk meg $n!$ értékét!

13. Feladat

Határozzuk meg két természetes szám szorzatát úgy, hogy nem használjuk a szorzás műveletét!

14. Feladat

Határozzuk meg egy valós szám pozitív egész kitevőjű hatványát úgy, hogy nem használjuk a hatványozás műveletét!

15. Feladat

Határozzuk meg két pozitív egész szám legnagyobb közös osztóját! (Euklideszi algoritmus)

16. Feladat

Határozzuk meg két pozitív egész szám legkisebb közös többszörösét!

17. Feladat

Készítsünk programot, mely egy tört számlálójának és nevezőjének megadása után kiírja az egyszerűsített törtet.

18. Feladat

Írjunk olyan programot, amely addig kér be egész számokat a billentyűzetről, amíg azok összege meg nem haladja a 100-at. A beolvasás végén írjuk ki azt, hogy a bekért számok közül hány volt páros, és hány volt páratlan.

19. Feladat

Írjunk programot, mely bekér 5 különböző karaktert, majd kiírja ezen karakterek összes permutációját.

Hátul tesztelő Ciklus – Do - While

A program a ciklusba egyszer mindenképpen belép, majd megvizsgálja a feltételt, amennyiben igaz, úgy ismételtén végrehajtja a ciklus magot.

- Ha a kilépési feltétel teljesül, akkor a ciklusmag végrehajtásra kerül, egyébként nem.
- Amíg a feltétel teljesül, a ciklusmag ismételtén végrehajtódik.
- Ha már a kilépési feltétel nem teljesül, akkor a program a ciklus utáni utasítással folytatódik.

Tipikusan olyan feladatok megoldására javasolható, ahol adatellenőrzést kell végezni.

Hátul tesztelő ciklus kódja

```
while True:
    if feltétel:
        utasítás
    else :
        break
```

20. Feladat

Írjunk programot, ami bekér a felhasználótól egy páros számot. Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismételjük meg a bekérést. Ha a szám megfelelő, írjuk ki a képernyőre: "A megadott szám megfelelő" .

21. Feladat

Írjunk programot, ami bekér a felhasználótól egy pozitív számot. Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismételjük meg a bekérést. Ha a szám megfelelő, írjuk ki a képernyőre a szám 5-tel vett osztási maradékát.

22. Feladat

Írjunk programot, ami bekér a felhasználótól egy számot az [1,7] intervallumból. Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismételjük meg a bekérést. Ha a szám megfelelő, írjuk ki a képernyőre a számnak megfelelő hét napját (hétfő, kedd szerda...).

23. Feladat

Írjunk programot, ami bekér a felhasználótól egy negatív és páratlan számot. Ha a felhasználó nem a feltételnek megfelelő számot ad meg, írjunk ki hibaüzenetet, és ismételjük meg a bekérést.

24. Feladat

Írjunk programot, ami bekér a felhasználótól egy 3-mal és 5-tel is osztható számot. Ha a felhasználó nem a feltételnek megfelelő számot ad meg, írjunk ki hibaüzenetet, és ismételjük meg a bekérést. Végül írjuk ki a képernyőre a szám harmadát és ötödét.

25. Feladat

Kérjünk be egy egész számot a felhasználótól, írjuk ki az 5-tel vett osztási maradékát! Ismételjük addig, amíg 0-t nem kapunk!

26. Feladat

Kérjünk be két valós számot a felhasználótól, írjuk ki, hogy melyik a nagyobb! Ismételjük addig, amíg egyenlő számot nem írunk be!

27. Feladat

Generáljunk egy véletlen egész számot az $[1,12[$ intervallumból, és írjuk ki a képernyőre a számot, és a szám 3-mal vett osztási maradékát. Ezek után kérdezzük meg a felhasználót, hogy szeretne-e újabb számot. Ha igen, ismételjük meg az előzőt, ha nem, lépünk ki a programból. A felhasználó választását (i/n) egy karakter típusú változóban tároljuk el!

28. Feladat

Készítsünk programot, amely bekér egész számokat mindaddig, amíg nem adjuk meg a 0-t. A program határozza meg és írja ki a beadott egész számok közül a legkisebbet és a legnagyobbat. (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi.) A számok beolvasását a 0 végjelig repeat .. until ciklus segítségével valósítsuk meg!

29. Feladat

Készítsünk játékprogramot, amely gondol egy számot 1 és 50 között. A felhasználó addig találgathat, amíg nem találja el a keresett számot. A számítógép minden rossz tipp után írja ki, hogy a gondolt szám nagyobb vagy kisebb.

30. Feladat

Égészítsük ki az előző programunkat úgy, hogy a játékos csak maximum 7-szer tippelhesen. Ha a hetedik tippre sem találja el a gondolt számot, a program írja ki a gondolt számot majd fejeződjön be.

31. Feladat

Olvassunk be pozitív egész számokat 0 végjelig. Írjuk ki a számok számát és a számok átlagát. (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi.)

32. Feladat

Írj szorzótábla programot. A program kérje be a szorzótábla alapszámát, és egy n értéket! Csak pozitív értékeket engedjen meg! Ezután írja ki a szorzótábla első n sorát, a következő formátumban:

- $1 * 5 = 5$
- $2 * 5 = 10$
- $3 * 5 = 15$
- $4 * 5 = 20$
- A példában az alapszám 5, n pedig 4

33. Feladat

Írj programot, amely alkalmas tetszőleges alapú szorzótábla gyakoroltatására! A program kérje be a szorzótábla alapszámát, illetve egy n értéket! Ezután írjon ki véletlenszerű szorzásokat, és kérje be az eredményt a felhasználótól!

- $3 * 5 = ?$
- $9 * 5 = ?$

Az első szám értéke legfeljebb a megadott n legyen! A program ellenőrizze a válasz helyességét! A gyakorlásból a 0 eredmény beírásával lehessen kilépni! A 0 választ a program ne értékelje!

34. Feladat

Készítsünk programot, amely ki fogja kérdezni a matematikát (két szám összeadását, kivonását és szorzását az $<1,10>$ intervallumból). A két számot és a műveletet a számítógép véletlenszerűen válassza ki. A program akkor fejeződjön be, ha a felhasználó 10 példát kiszámolt helyesen. Rossz válasz esetén kérdezze újra ugyanazt a példát. A program végén írjuk ki az eredményességet százalékokban.

35. Feladat

Írj programot, amely meghatározza egy megadott számtani sorozat első n elemének az összegét!

Példa számtani sorozatra: $a_1=4$ $d=2 \Rightarrow 4, 6, 8, 10, \dots$

A felhasználó adja meg a számtani sorozat első elemét, illetve differenciáját! A program határozza meg és írja ki az első n elemet és azok összegét! A program számolja ki az első n elem összegét az ismert összegképlet segítségével is! Írja ki mindkét eredményt!

36. Feladat

Írj programot, amely meghatározza egy megadott mértani sorozat első n elemének az összegét!

Példa mértani sorozatra: $a_1=4$ $q=2 \Rightarrow 4, 8, 16, 32, \dots$

A felhasználó adja meg a mértani sorozat első elemét, illetve hányadosát! A program határozza meg és írja ki az első n elemet és azok összegét! A program számolja ki az első n elem összegét az ismert összegképlet segítségével is! Írja ki mindkét eredményt!

37. Feladat

Készítsünk programot, amely kiszámolja $a/b+c/d$ (két tört összegét), majd az eredmény törzsalakú törté alakítja (amely már nem egyszerűsíthető tovább).