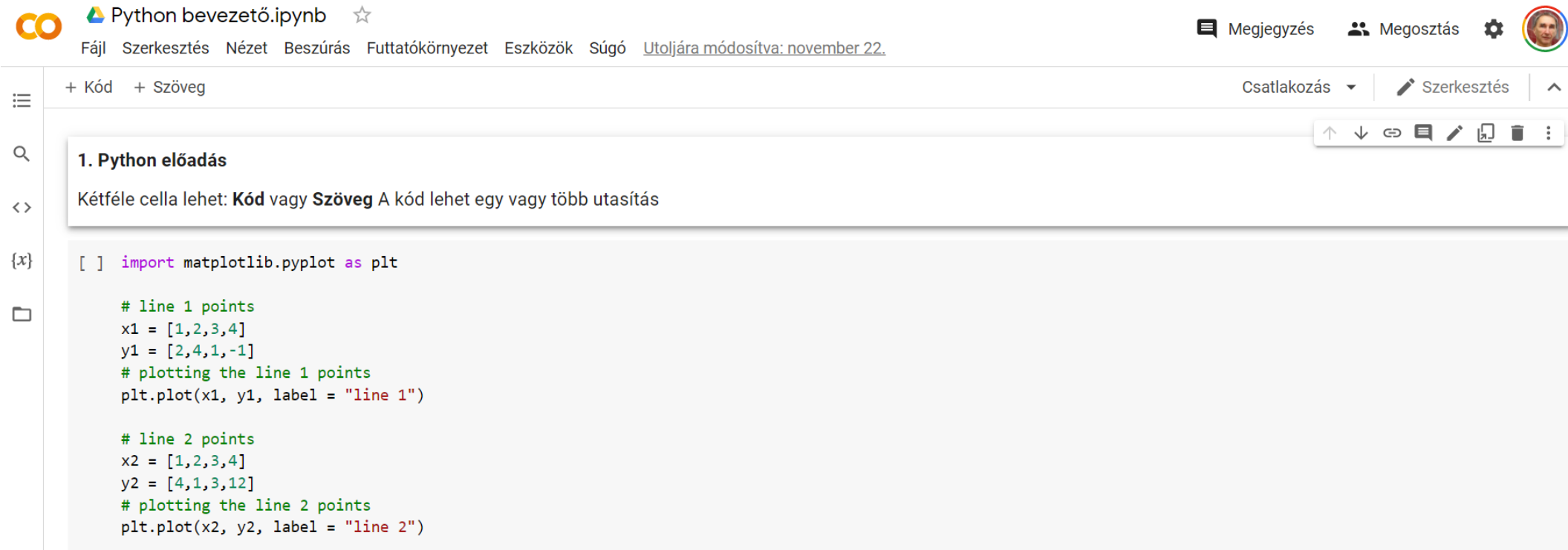


10. GYAKORLAT - PYTHON

Böngészőből használható python: <https://colab.research.google.com/> (Google fiókkal kell bejelentkezni, a Google Drive-ra fogja menteni a Jegyzetfüzetet (Notebook)-ot.)



The screenshot shows a Google Colab notebook titled "Python bevezető.ipynb". The interface includes a top navigation bar with options like "Fájl", "Szerkesztés", "Nézet", "Beszúrás", "Futtatókörnyezet", "Eszközök", "Súgó", and "Utoljára módosítva: november 22.". On the right, there are icons for "Megjegyzés", "Megosztás", and a user profile. The notebook content is divided into two sections: a text cell and a code cell. The text cell contains the instruction: "Kétféle cella lehet: **Kód** vagy **Szöveg** A kód lehet egy vagy több utasítás". The code cell contains the following Python code:

```
[ ] import matplotlib.pyplot as plt

# line 1 points
x1 = [1,2,3,4]
y1 = [2,4,1,-1]
# plotting the line 1 points
plt.plot(x1, y1, label = "line 1")

# line 2 points
x2 = [1,2,3,4]
y2 = [4,1,3,12]
# plotting the line 2 points
plt.plot(x2, y2, label = "line 2")
```

Tankönyv:

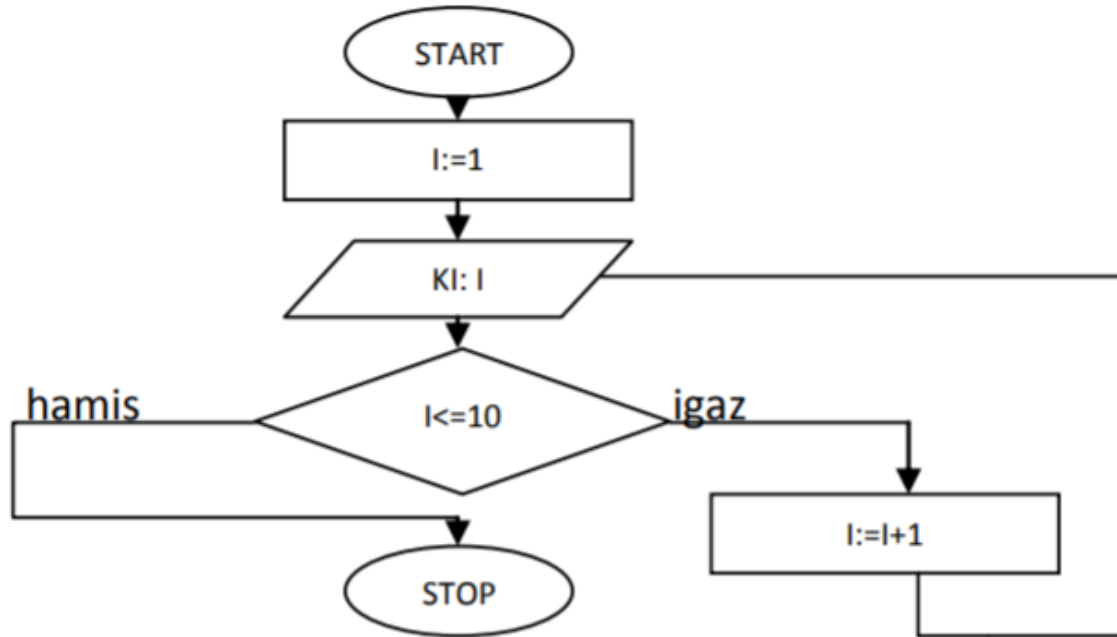
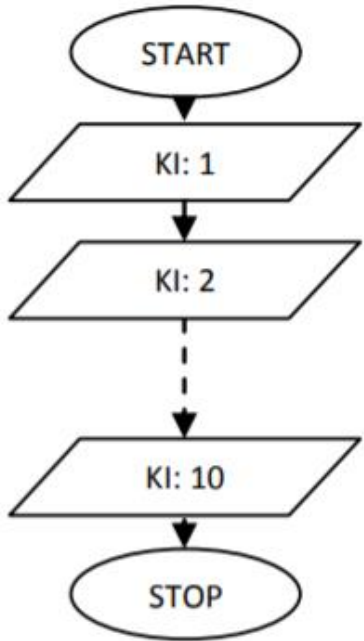
<https://people.inf.elte.hu/kiss/bevinfo/Python%20tank%f6nyv.pdf>

Tutorial, szintaxis (angolul):

<https://www.w3schools.com/python/default.asp>



ÍRJUK KI A SZÁMOKAT 1-TŐL 10-IG!



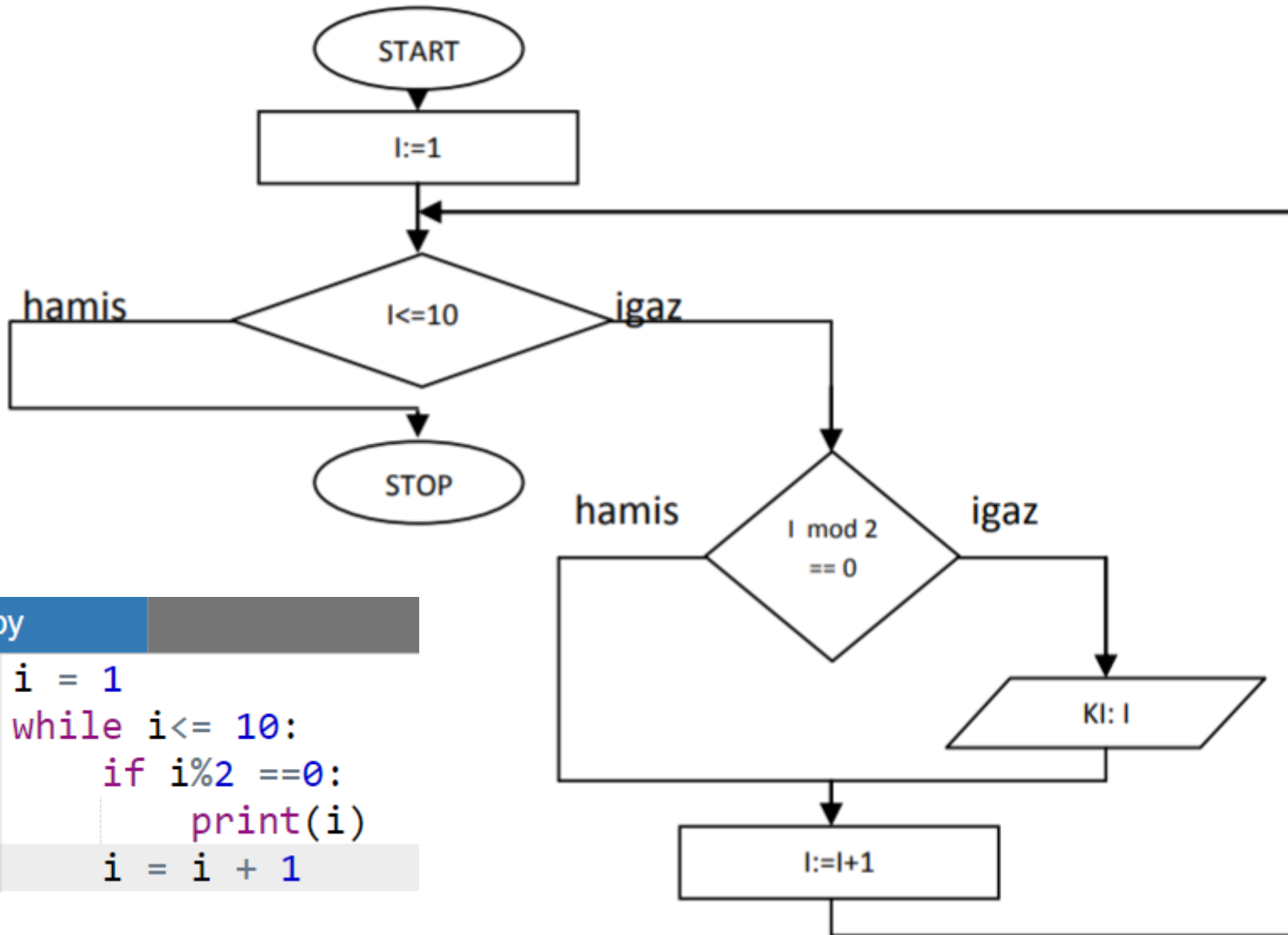
Ugyanez rövidebben (while ciklus segítségével).

```
✓ 0 mp  
▶ print(1)  
print(2)  
print(3)  
print(4)  
print(5)  
print(6)  
print(7)  
print(8)  
print(9)  
print(10)
```

```
{x}  
✓ 0 mp  
[2] i = 1  
while i <= 10:  
    print(i)  
    i = i + 1
```



ÍRJUK KI A **PÁROS** SZÁMOKAT 1-TŐL 10-IG!

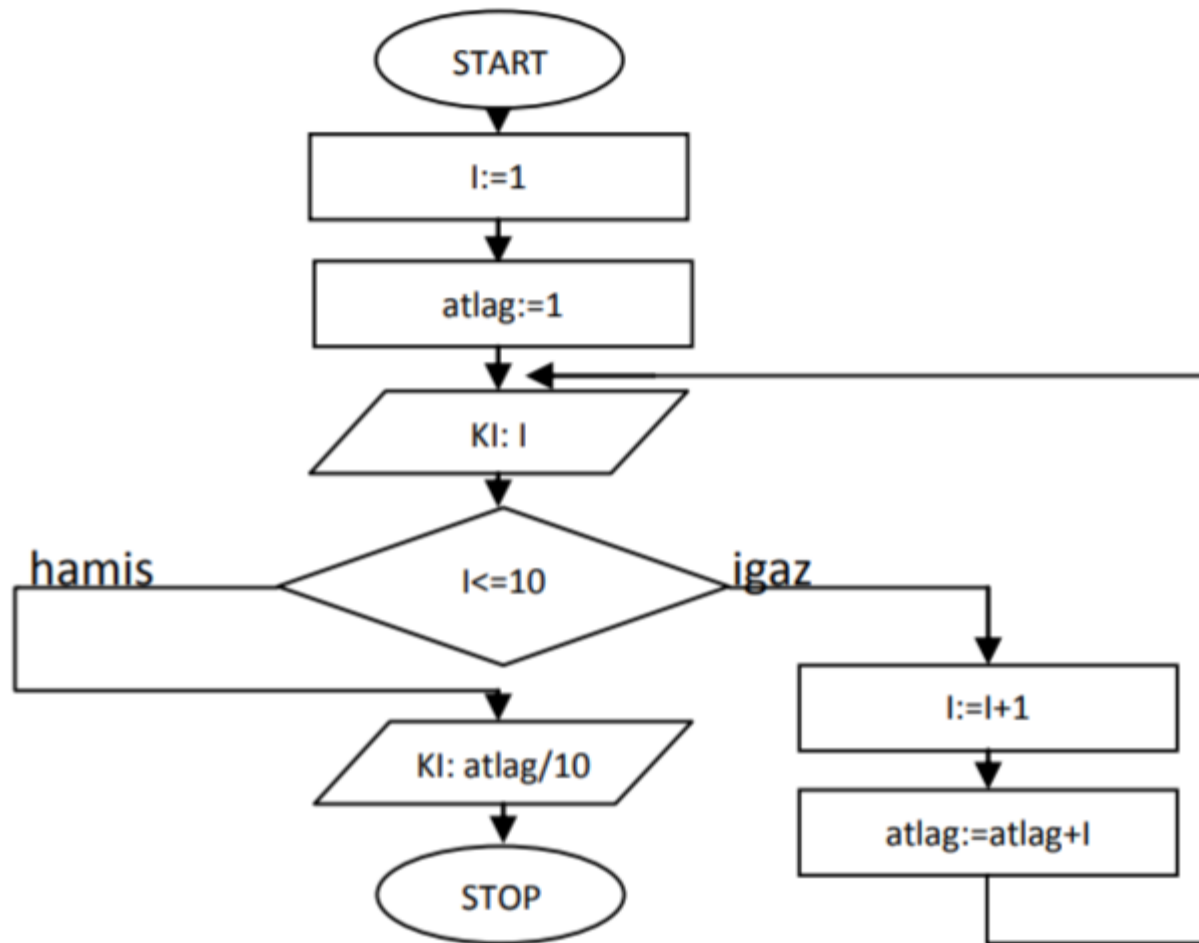


main.py

```
1 i = 1
2 while i <= 10:
3     if i%2 == 0:
4         print(i)
5     i = i + 1
```



ÍRJUK KI A SZÁMOKAT 1-TŐL 10-IG ÉS A VÉGÉN AZ ÁTLAGUKAT IS!



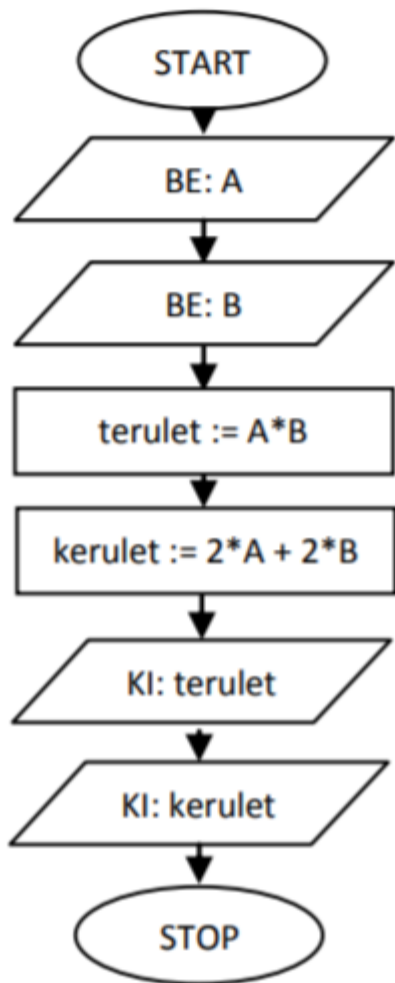
main.py

```
1 i = 1
2 összeg = 0
3 while i <= 10:
4     print(i)
5     összeg = összeg + i
6     i = i + 1
7 átlag = összeg / 10
8 print (átlag)
```

Az átlag változóban az összeg képződik. Amikor mind a tízet összeadtuk, a 10-zel osztással kapjuk meg az átlagot.

TÉGLALAP KERÜLETÉNEK, TERÜLETÉNEK

MEGHATÁROZÁSA: A PROGRAM KÉRJE BE A TÉGLALAP
A ÉS B OLDALÁT, MAJD AZ EREDMÉNYT ÍRJA KI.

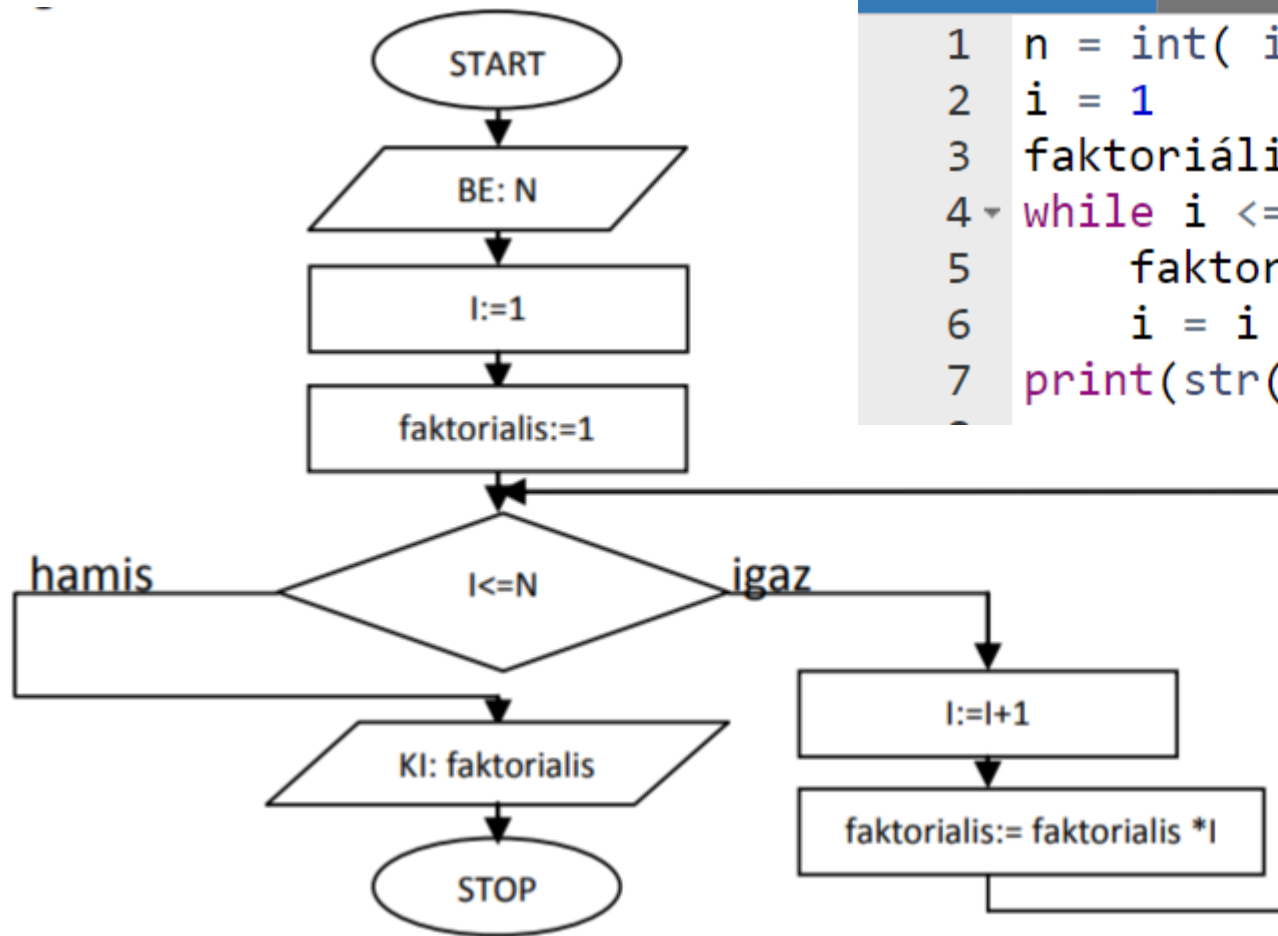


main.py

```
1 a=int( input("Add meg az egyik oldalt: ") )
2 b=int( input("Add meg a másik oldalt: ") )
3 kerület = 2 * (a + b)
4 terület = a * b
5 print (kerület, terület)
```



FAKTORIÁLIS ($N! = 1*2*3*4*...*(N-1)*N$) KIÍRÁSA



main.py

```
1 n = int( input("n:") )
2 i = 1
3 faktoriális = 1
4 while i <= n:
5     faktoriális = faktoriális * i
6     i = i + 1
7 print(str(n)+"! = ", faktoriális)
```

1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

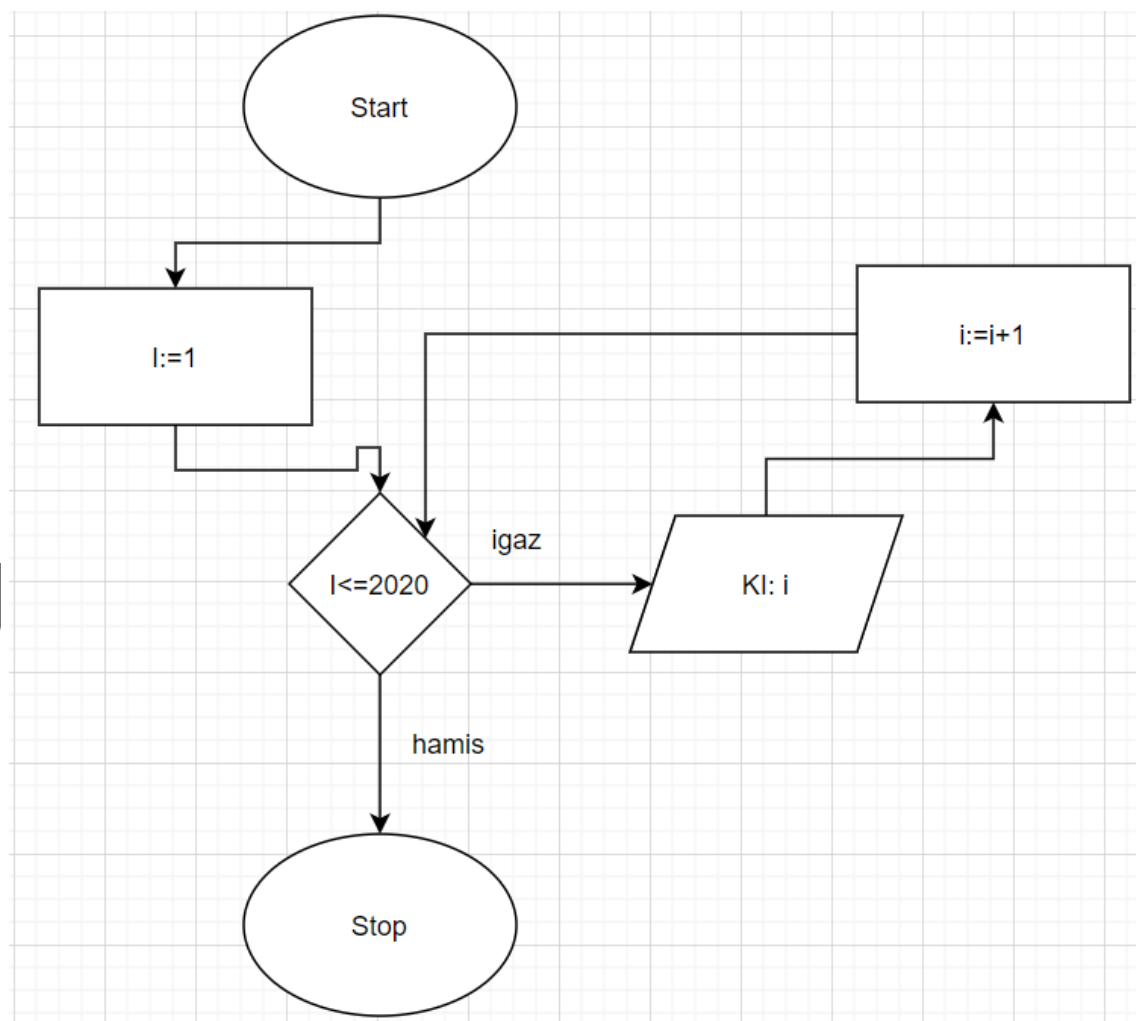
1. Írjuk ki a számokat 1-től 2020-ig.

main.py

```
1 i = 1
2 while i <= 2020:
3     print(i)
4     i = i + 1
5
```

main.py

```
1 for i in range(1,2021):
2     print(i)
```

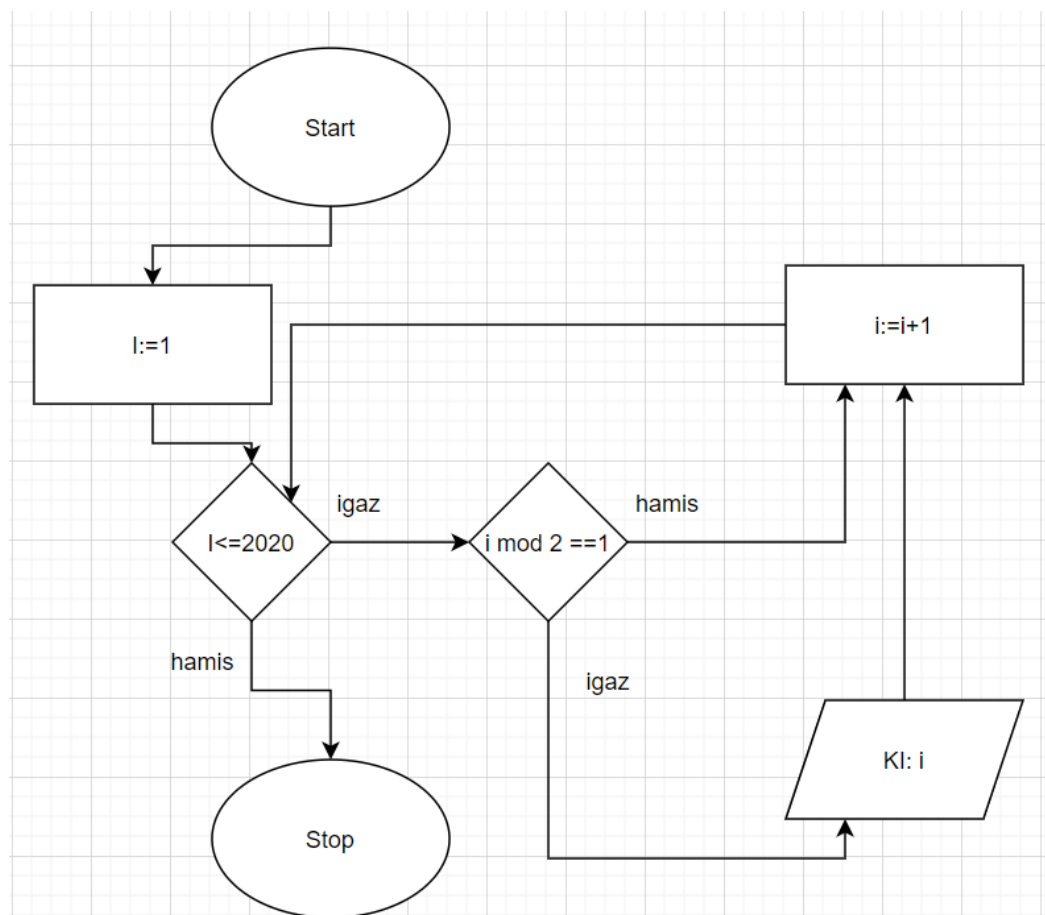


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

2. Írjuk ki a **páratlan** számokat 1-től 2020-ig.

main.py

```
1 for i in range(1,2021):  
2     if i % 2 == 1:  
3         print(i)
```

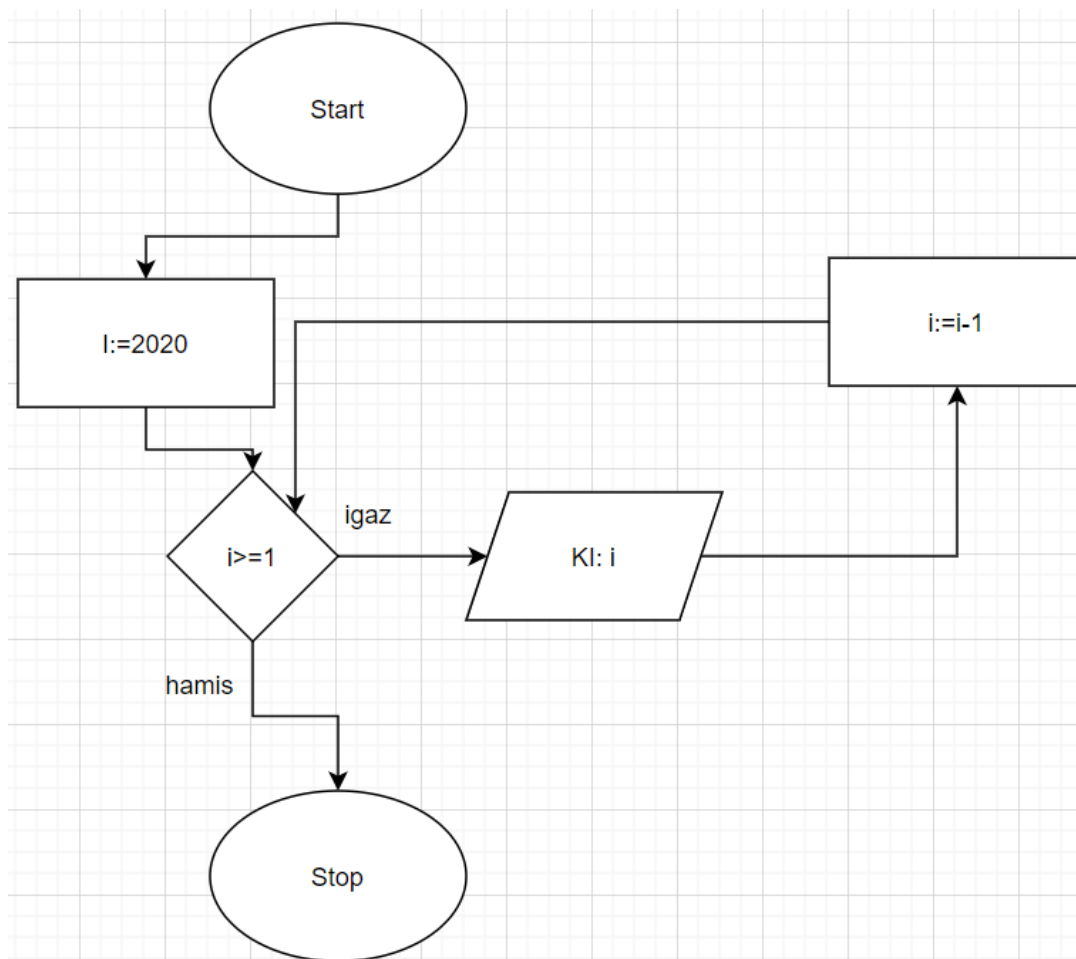


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

3. Írjuk ki a számokat **csökkenő** sorrendben 2020-tól 1-ig.

main.py

```
1 i = 2020
2 while i >= 1:
3     print (i)
4     i = i - 1
```

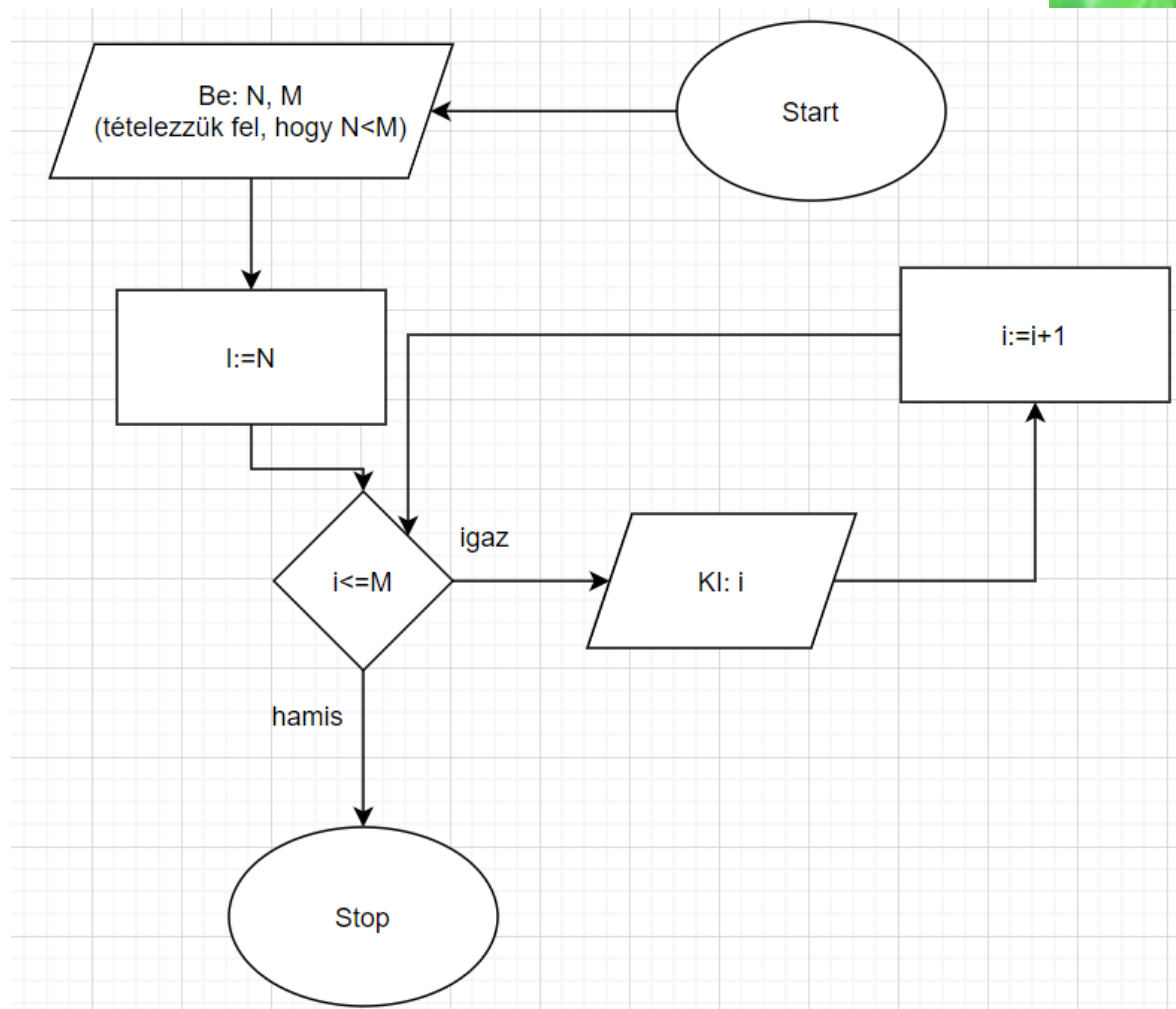


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

4. Írjuk ki a számokat egy beolvasott **N** számtól egy beolvasott **M** számig.

main.py

```
1 N = int( input("N: "))
2 M = int( input("M: "))
3 # feltesszük, hogy M a nagyobb
4 i = N
5 while i <= M:
6     print(i)
7     i = i + 1
```

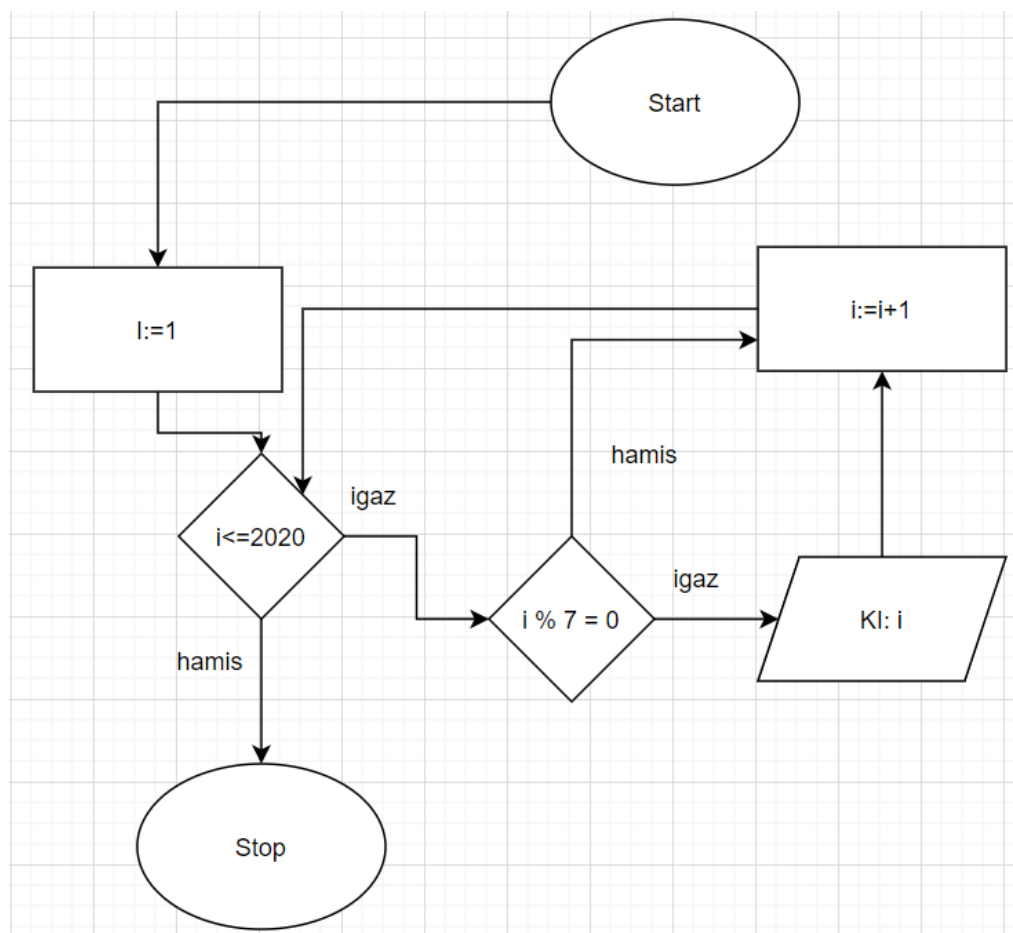


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

5. Írjuk ki a 7-tel osztható számokat 1 és 2020 között. (7-tel osztható a szám, ha 0 a 7-tel elvégzett maradékos osztás maradéka, azaz szám $\% 7 = 0$.)

main.py

```
1 i = 1
2 while i <= 2020:
3     if i % 7 == 0:
4         print(i)
5     i = i + 1
```

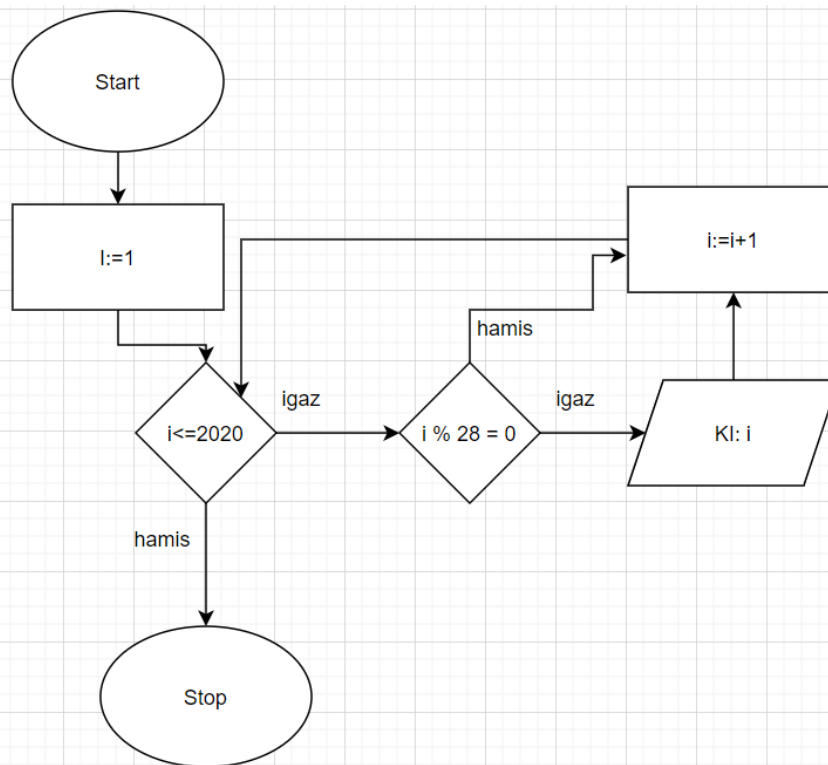
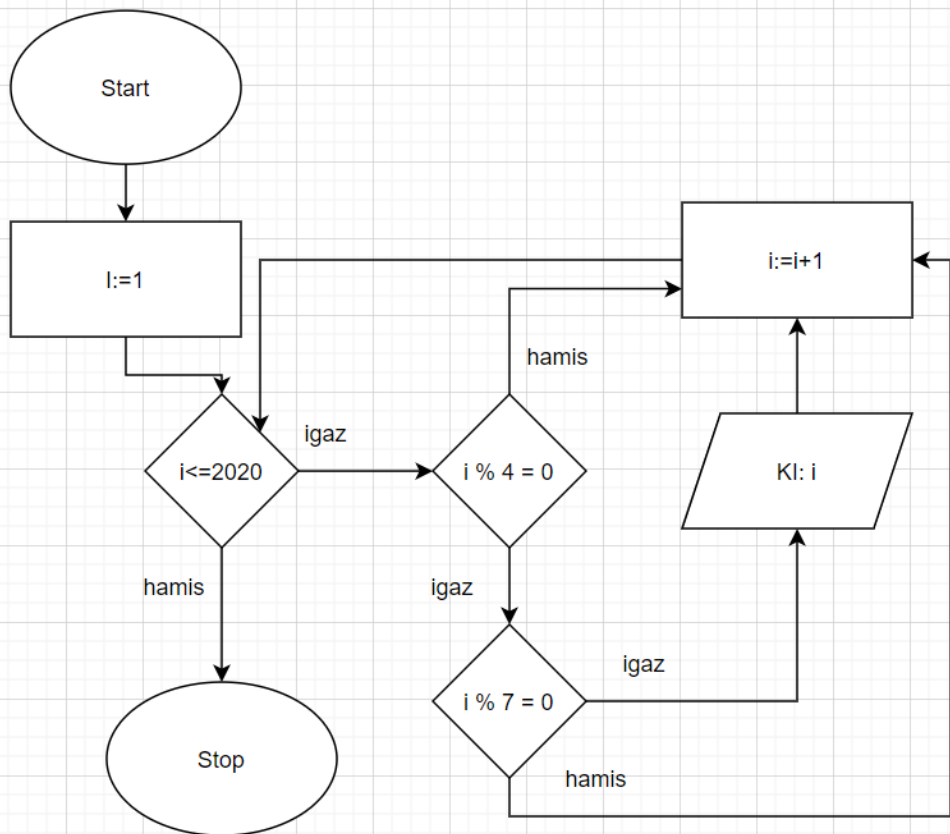


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

6. Írjuk ki a 4 és 7-tel osztható számokat 1 és 2020 között. A 2. megoldásnál kihasználtuk, hogy egy szám osztható 4-gyel és 7-tel, akkor és csak akkor, ha a legkisebb közös többszörösükkel, azaz 28-cal osztható.

main.py

```
1 i = 1
2 while i <= 2020:
3     if i % 4 == 0 and i % 7 == 0:
4         print(i)
5     i = i + 1
```

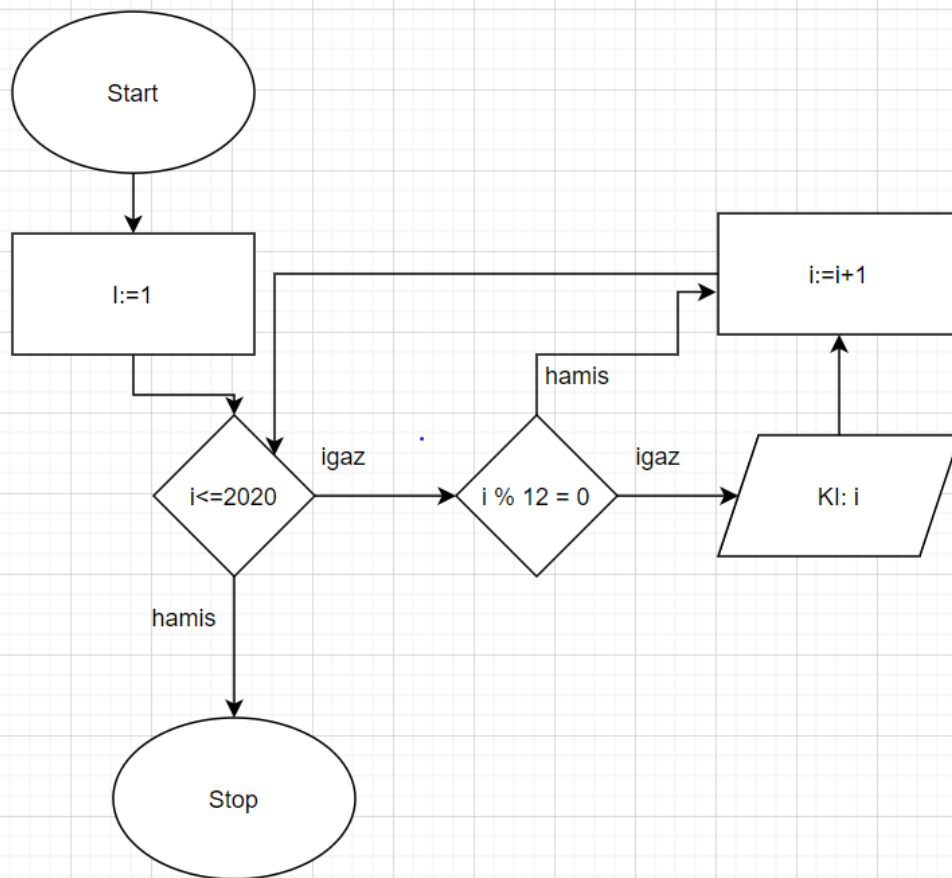
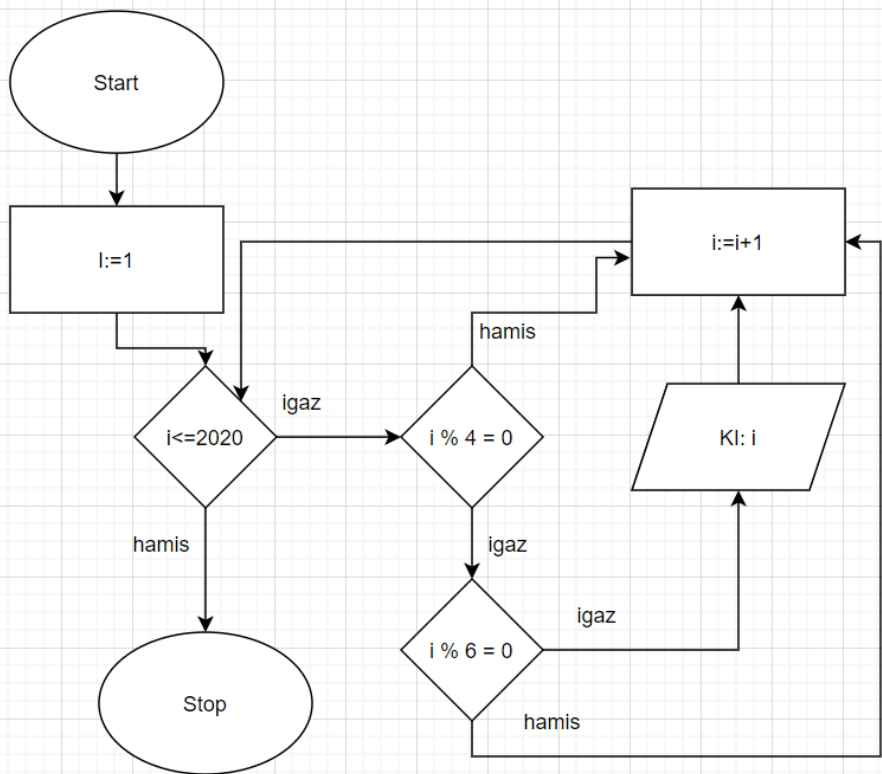


1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

7. Írjuk ki a 4 **és** 6-tal osztható számokat 1 és 2020 között. A 2. megoldásnál kihasználtuk, hogy egy szám osztható 4-gyel és 6-tal, akkor és csak akkor, ha a legkisebb közös többszörösükkel, azaz 12-vel osztható.

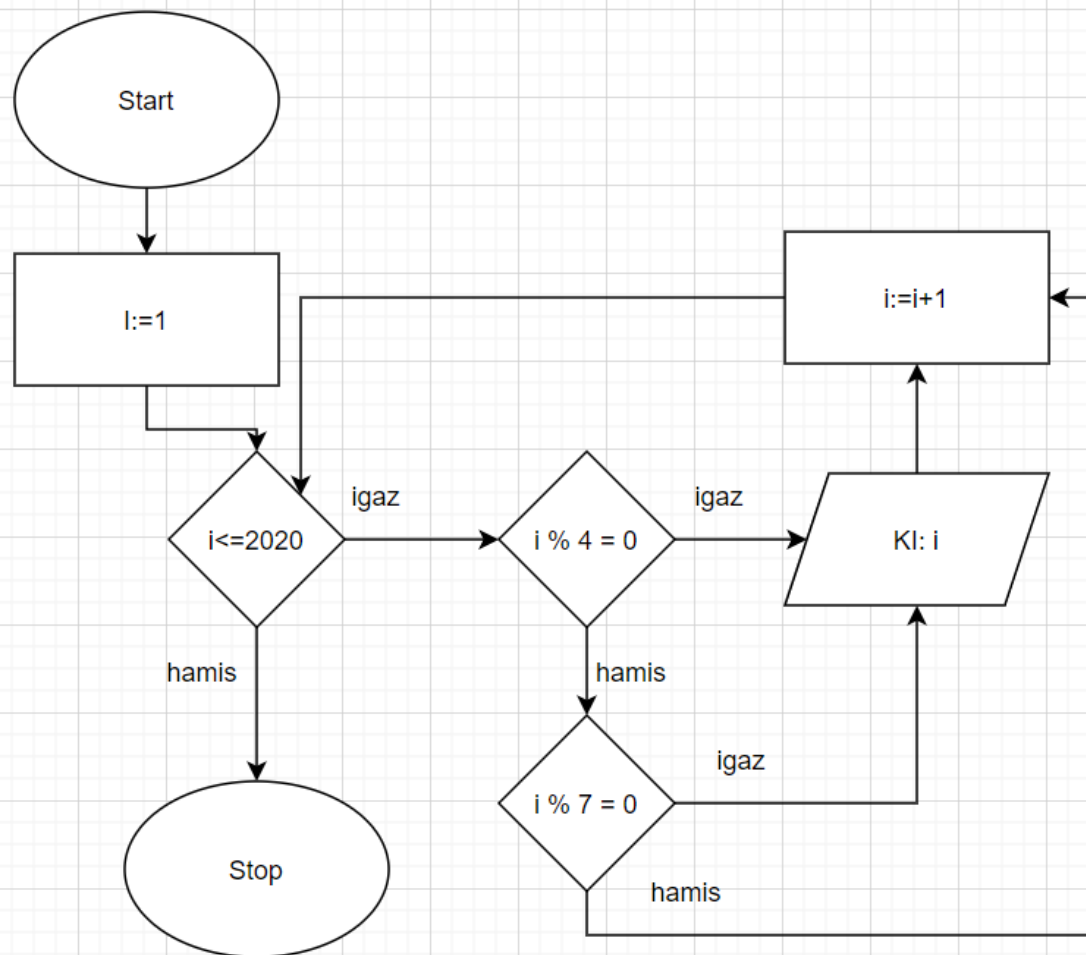
main.py

```
1 i = 1
2 while i <= 2020:
3     if i % 4 == 0 and i % 6 == 0:
4         print(i)
5     i = i + 1
```



1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

8. Írjuk ki a 4 **vagy** 7-tel osztható számokat 1 és 2020 között.

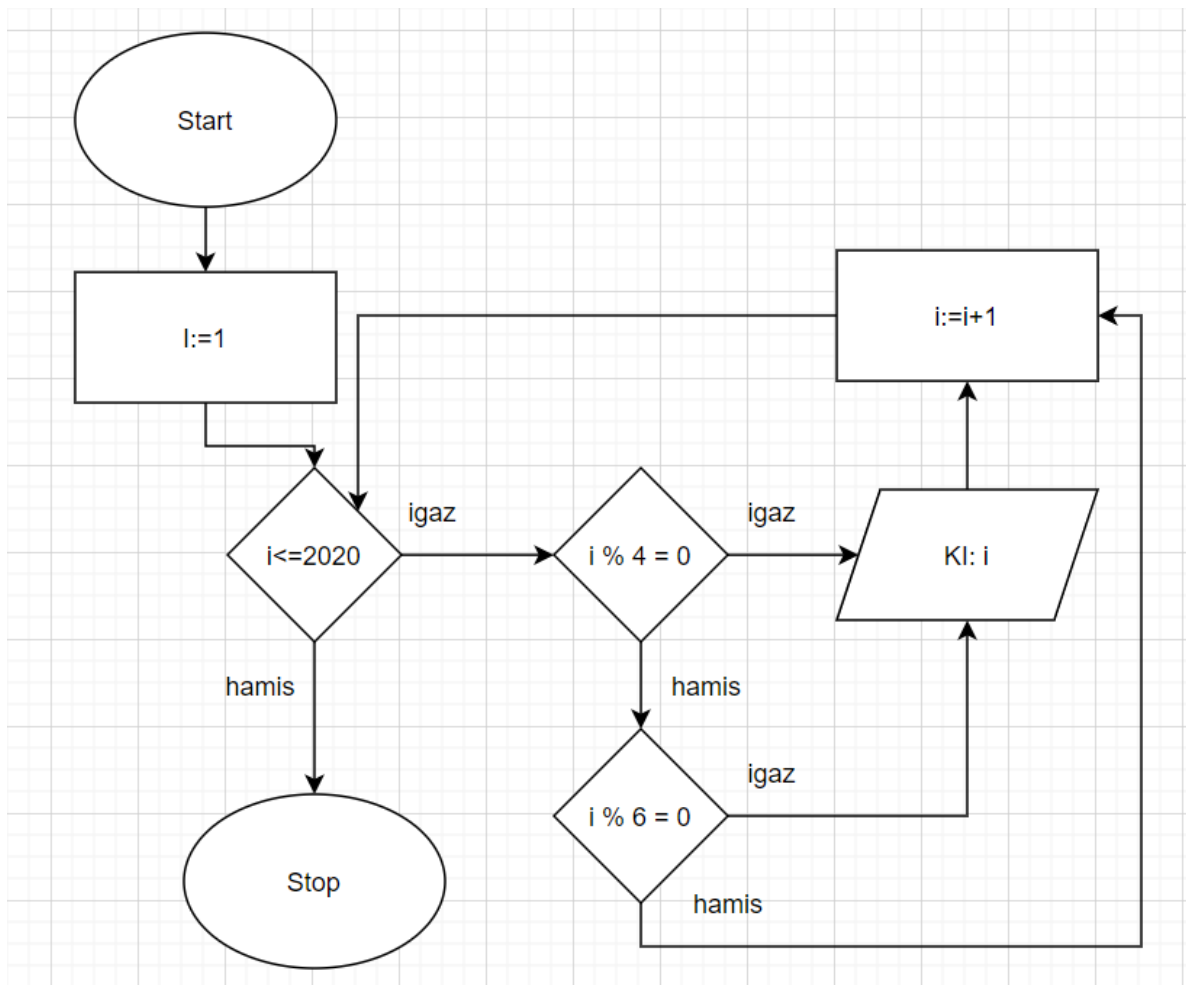


main.py

```
1 i = 1
2 while i <= 2020:
3     if i % 4 == 0 or i % 7 == 0:
4         print(i)
5     i = i + 1
```

1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

9. Írjuk ki a 4 **vagy** 6-tal osztható számokat 1 és 2020 között.



main.py

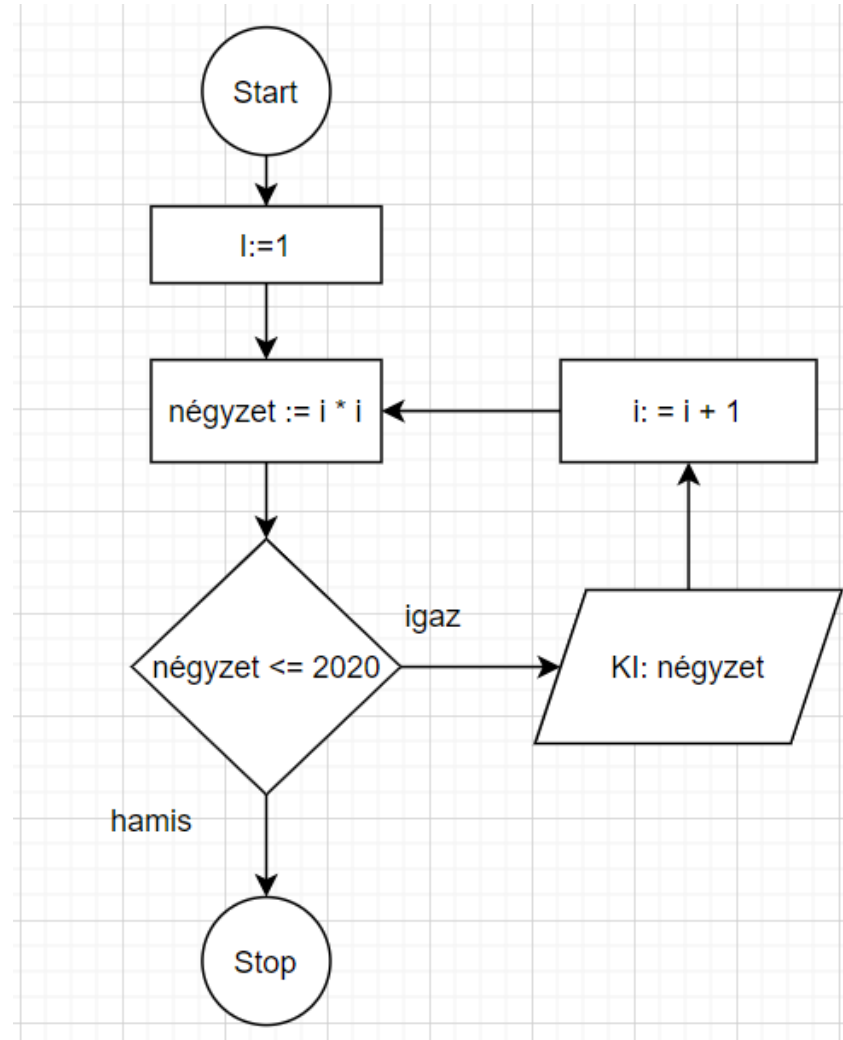
```
1 i = 1
2 while i <= 2020:
3     if i % 4 == 0 or i % 6 == 0:
4         print(i)
5     i = i + 1
```

1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

10. Írjuk ki a négyzetszámokat 1-től 2020-ig.

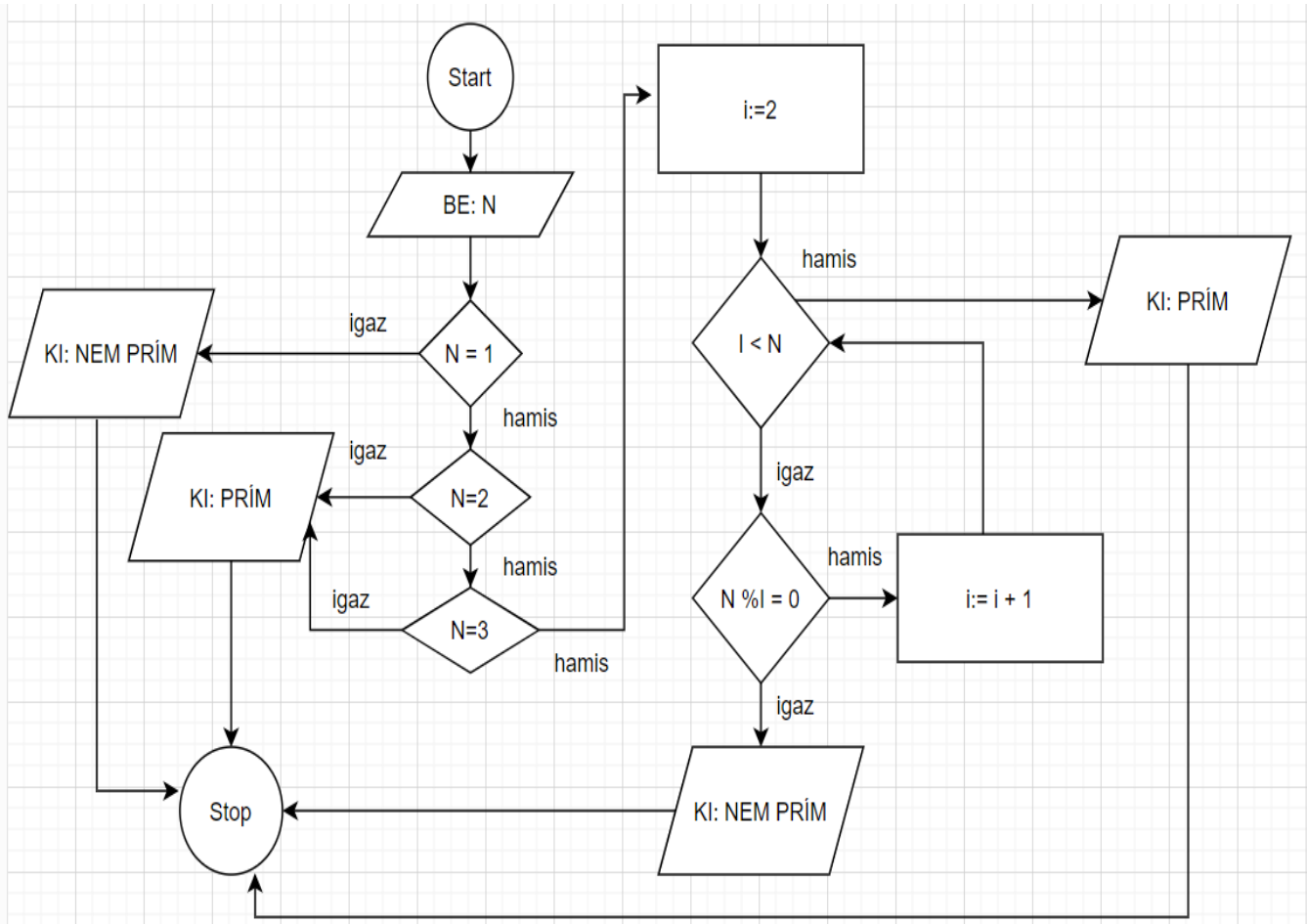
main.py

```
1 i = 1
2 négyzet = i * i
3 while négyzet <= 2020:
4     print(négyzet)
5     i = i + 1
6     négyzet = i * i
```



1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

11. Írjuk ki, hogy egy beolvasott N szám PRÍM vagy NEM PRÍM.

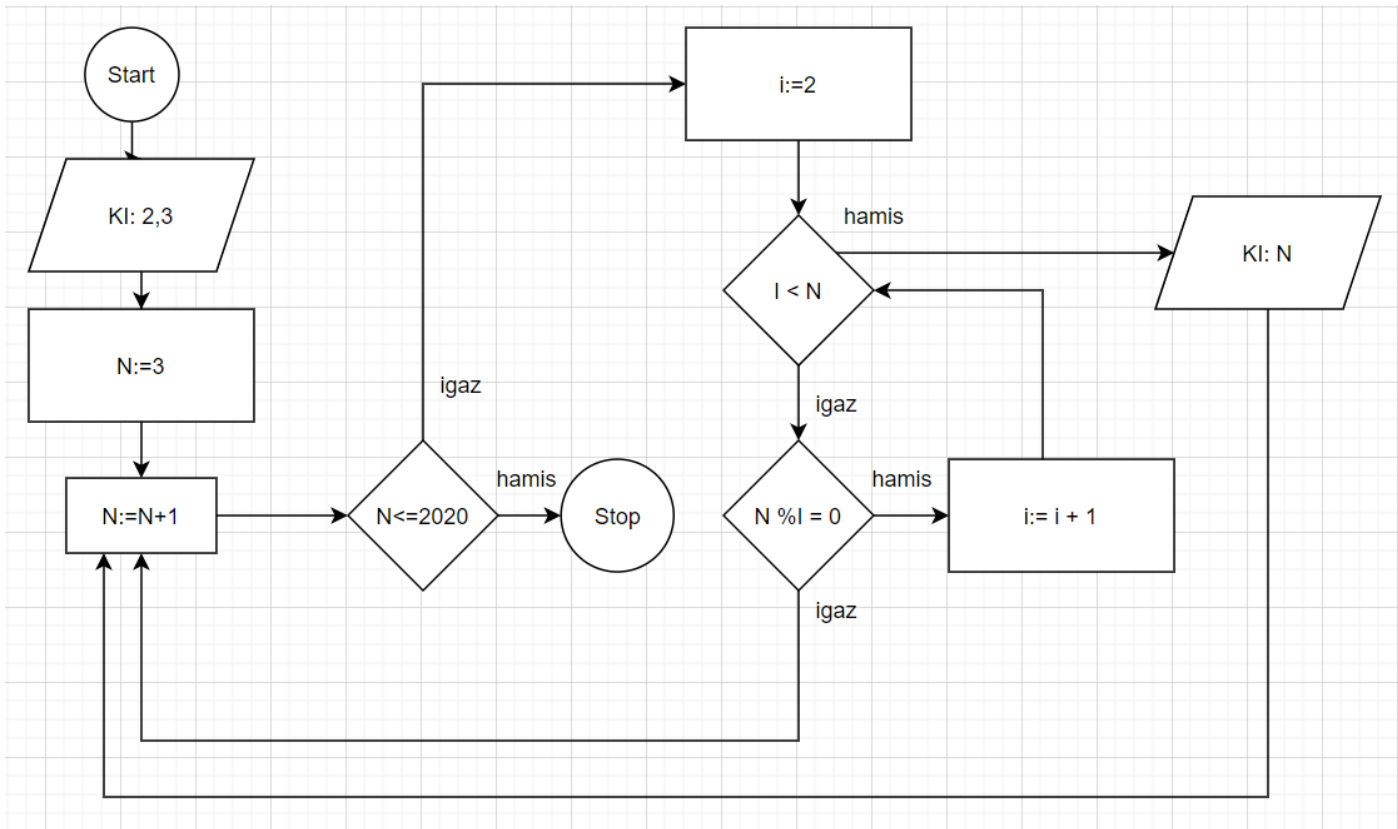


main.py

```
1 N = int(input("N: "))
2 if N == 1:
3     print("Nem prím")
4 elif N == 2:
5     print("Prím")
6 elif N == 3:
7     print("Prím")
8 else:
9     i = 2
10    while i < N:
11        if N % i == 0:
12            print("Nem prím")
13            quit()
14            i = i + 1
15    print("Prím")
```

1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

12. Írjuk ki a prímszámokat 1-től 2020-ig.

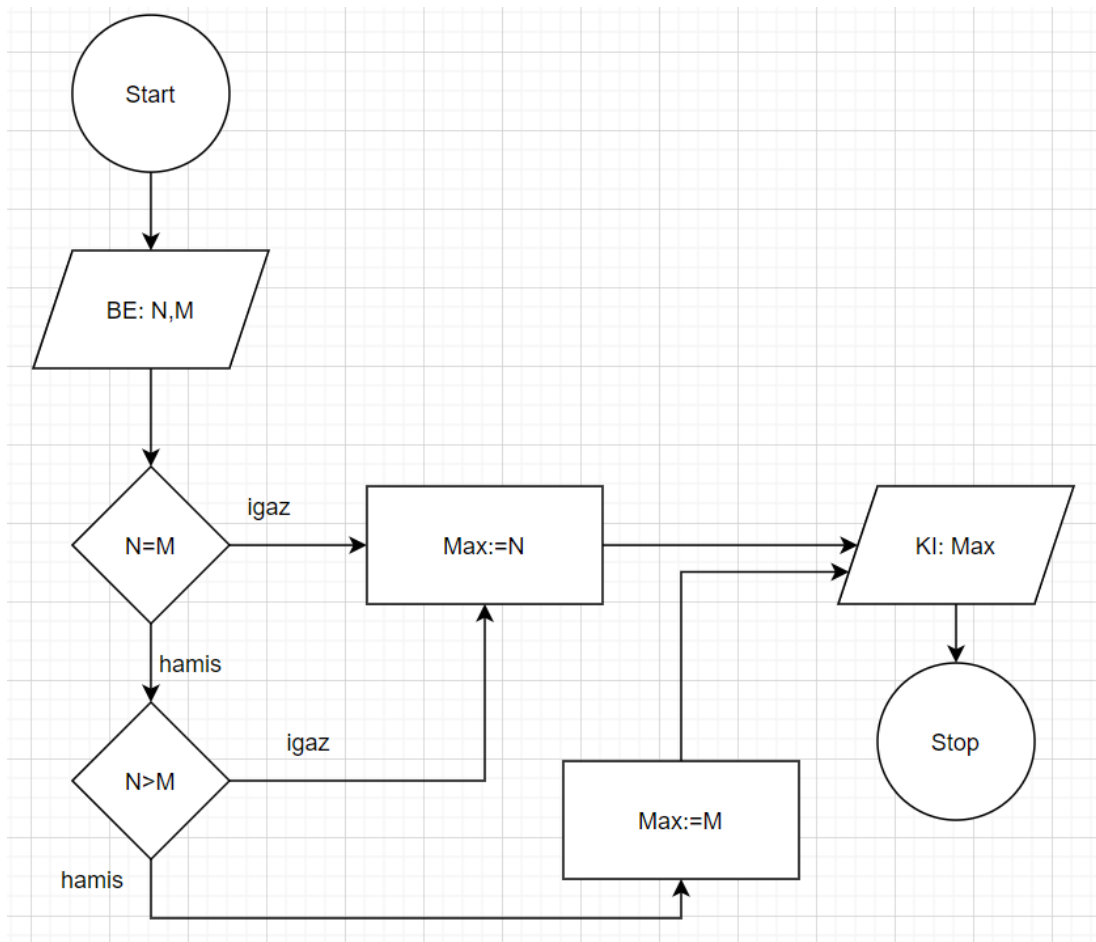


main.py

```
1 def prim(N):
2     if N == 1:
3         return False
4     elif N == 2 or N == 3:
5         return True
6     else:
7         i = 2
8         while i < N:
9             if N % i == 0:
10                return False
11                quit()
12                i = i + 1
13            return True
14
15
16 for j in range(1, 2020 + 1):
17     if prim(j):
18         print (j)
```

1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

13. Írjuk ki egy Max változóba a beolvasott N és M számok közül a nagyobbat.



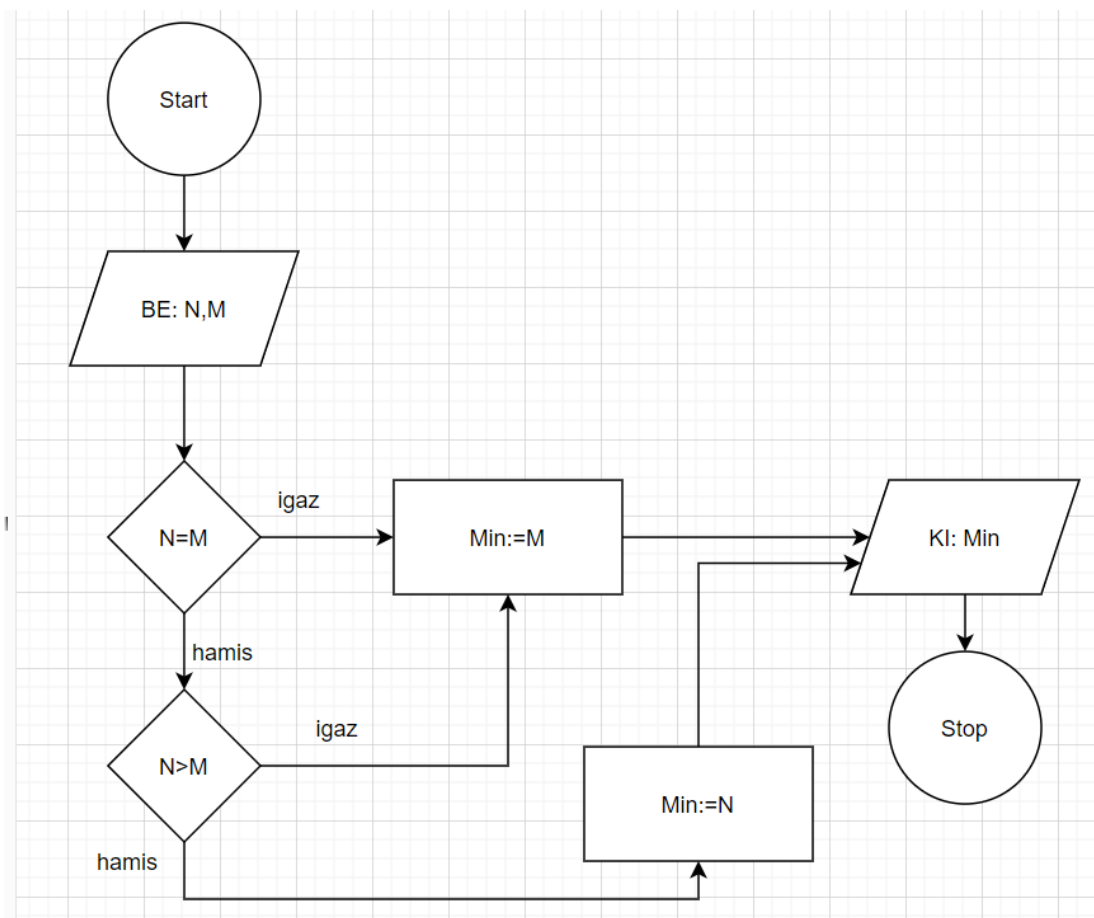
main.py

```
1 N = int(input("N: "))
2 M = int(input("M: "))
3 if N <= M:
4     Max = M
5     print(Max)
6 else:
7     Max = N
8     print(Max)
```



1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

14. Írjuk ki egy Min változóba a beolvasott N és M számok közül a kisebbet.



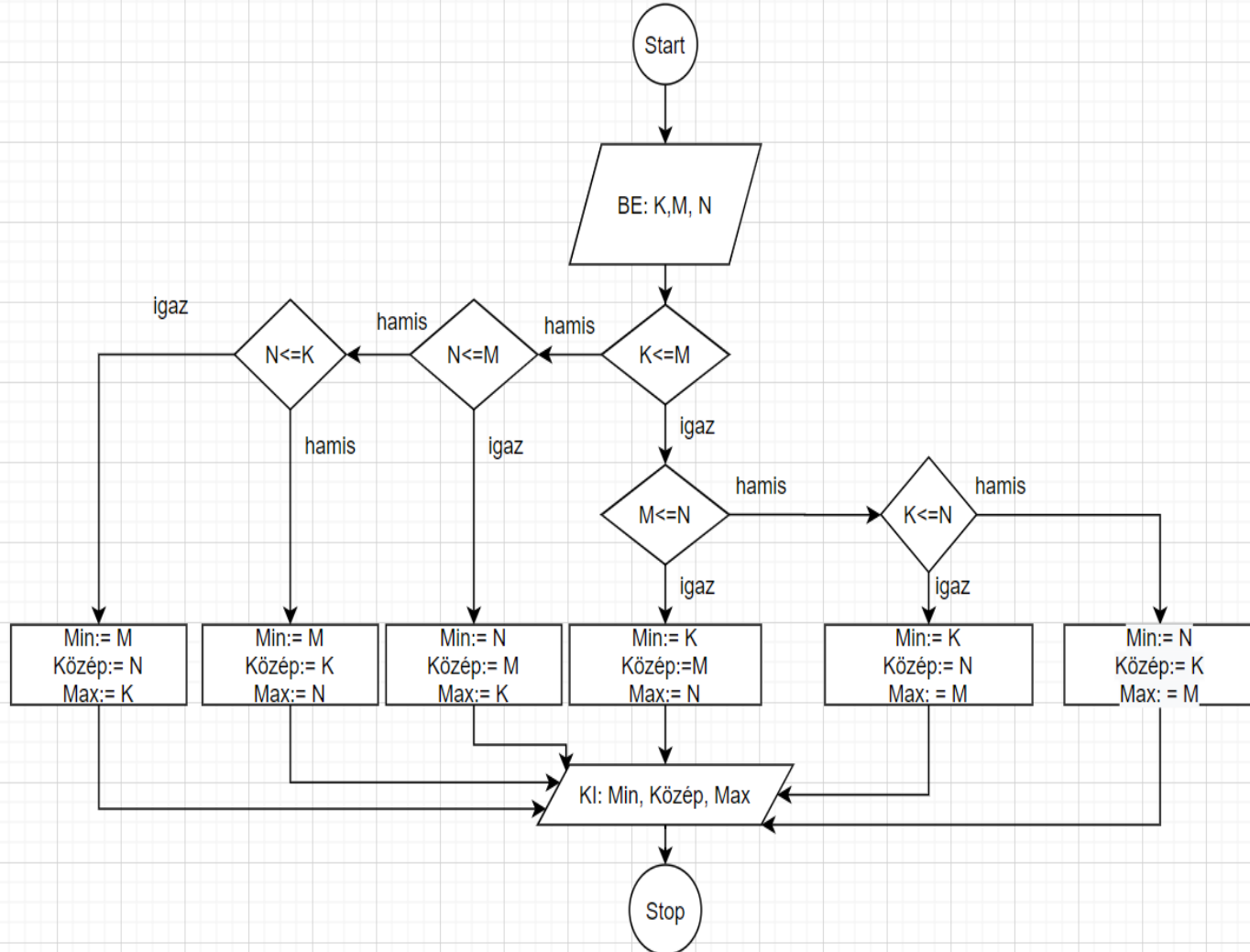
main.py

```
1 N = int(input("N: "))
2 M = int(input("M: "))
3 if N <= M:
4     Min = N
5     print(Min)
6 else:
7     Min = M
8     print(Min)
```



1. GYAKORLAT FELADATSORA ÉS MEGOLDÁSAI

15. Határozzuk meg, hogy a beolvasott K,N,M közül melyik a legnagyobb (Max), melyik a legkisebb (Min) és mi



main.py

```
1 K = int(input("K: "))
2 N = int(input("N: "))
3 M = int(input("M: "))
4 if K <= M:
5     if M <= N:
6         Min = K
7         Közép = M
8         Max = N
9     else:
10        if K <= N:
11            Min = K
12            Közép = N
13            Max = M
14        else:
15            Min = N
16            Közép = K
17            Max = M
18    else:
19        if N <= M:
20            Min = N
21            Közép = M
22            Max = K
23        else:
24            if N <= K:
25                Min = M
26                Közép = N
27                Max = K
28            else:
29                Min = M
30                Közép = K
31                Max = N
32    print(Min, Közép, Max)
```



Feladatok!

Minden feladat egy 10gyak.ipynb nevű notebookban, külön-külön cellában legyen. Minden feladat előtt szöveges cellában szerepeljen, hogy hanyadik feladat jön utána. A notebook-ot a statisztika2016@gmail.com címmel osszátok meg. A megoldott feladatot **formázva** (azaz gmailben **ne legyen** az egyszerű szöveg bekattintva) küldjétek el a statisztika2016@gmail.com címre.

10.1 feladat

statisztika2016@gmail.com

10.1 feladat

```
def prim(n):  
    if n == 1:  
        return False  
    elif n == 2 or n == 3:  
        return True  
    else:  
        i = 2  
        while i < n:  
            if n % i == 0:
```

↩ ↪ Rögzített ... 🔍

Küldés



nyezet Eszközök Súgó Minden módosítás mentve

Megjegyzés

Megosztás

RAM Lemez Szerkesztés

← Megosztás személyekkel és csoportokkal ⚙️

statisztika2016@gmail.com ×

Szerkesztő

Felhasználók értesítése

1. Írjuk ki, hogy SZERKESZTHETŐ, ha a beolvasott K,M,N pozitív számokkal megegyező hosszú oldalakkal szerkeszthető háromszög. (**Bármely két** oldal hosszainak összege > harmadik oldal hossza.) Ha nem szerkeszthető, akkor azt írjuk ki, hogy NEM SZERKESZTHETŐ.
2. Írjuk ki a **páratlan** számok **összegét** egy beolvasott **N** számtól egy beolvasott **M** számig (N<M), végpontokat is beleértve.
3. Írjuk ki **hány darab** 11-gyel osztható szám van egy beolvasott **N** számtól egy beolvasott **M** számig (N<M), végpontokat is beleértve.
4. Írjuk ki **hány darab páros négyzetszám** van egy beolvasott **N** számtól egy beolvasott **M** számig (N<M), végpontokat is beleértve.
5. Írjuk ki az 50 és 100 közötti **prímeket**, **csökkenő** sorrendben.
6. Egy beolvasott **N** számról állapítsuk meg, hogy 2^M alakú-e (kettőhatvány-e) valamilyen $M > 1$ egészre. Ha kettőhatvány, akkor írjuk ki, hogy KETTŐHATVÁNY, ha nem, akkor azt hogy NEM KETTŐHATVÁNY.