https://oroszl.web.elte.hu/szamprob/static_notebooks/Package01/mintapelda01.html

1., FELADAT: A kifejezések nyelvtana a szokásos: a +, -, * és / műveletek ugyanúgy működnek, mint a legtöbb nyelvben.
A zárójeleket az elvégzendő műveletek csoportosítására, a műveleti sorrend meghatározására használjuk.

```
print(2+2)                # 4
print(50 - 5*6)           # 20
print( (50 - 5*6) / 5 )   # 5.0
print(8 / 5)              # 1.6
print( 17 / 3 )           # 5.666666666666667
print( 17 // 3 )          # 5
print( 17 % 3 )           # 2
print( 5 ** 2 )           # 25
print( 2 ** 7 )           # 128
```

#2., FELADAT: Az egész számok (pl. 2, 4, 20) alkotják az int típust, a valós számok (pl 5.0, 1.6) pedig a float típust.
A Python 3-ban az osztás (/) mindig lebegőpontos értékeket ad vissza.
Lefelé kerekítő egészosztás elvégzéséhez, amellyel egész értéket kapunk,a // operátor használható;
az osztás maradékához a %-t hívhatjuk segítségül:

```
print("\n")
print( int(3.1415) )    # 3
print(float(42))        # 42.0
```

#3., FELADAT: Logikai Bool-változók
Két objektum azonosságáról dupla egyenlőségjellel, azaz a == operátor segítségével tájékozódhatunk.

```
print("\n")
print(1 == 1)           # True
print(1 == 2)           # False
print(1 != 2)           # True
print(1 > 2)            # False
print(1 < 2)            # True

b=3
print (1 < 2 == b)
```

#4., FELADAT: A relációkat összefűzhetjük and és or logikai műveletekkel is.
Reláció erdményét (vagy bármely logikai műveletét) ellentettjére változtathatjuk a not művelettel.

```
not 2*2==5              # True

b=3
((1==2) and (3==4)) or (b==3)       # True
```

#5., FELADAT: Karakterláncokat a + művelettel ragaszthatunk össze, és *-gal ismételhetünk.

```
print ( 3 * 'Kálc' + 'ium' )            # KálcKálcKálcium
print ( 2 * 'Pi' + 'tér' )              # PiPitér
```