"""

1. feladat: Készítsen python programot, Kérjen be a felhasználótól két egész számot és egy alapműveletet

(+, - , *, /)! Írja ki a képernyőre a művelet eredményét!

"""

```
szam1 = int(input("Kérem adjon meg egy számot! "))
szam2 = int(input("Kérem adjon meg egy másik számot! "))
mjel = input("Kérek egy műveleti jelet! ")

if mjel == ("+"):           #elágazással oldjuk meg
    print("szam1 + szam2 = {}".format(szam1+szam2))
elif mjel == ("-"):
    print("szam1 - szam2 = {}".format(szam1-szam2))
elif mjel == ("*"):
    print("szam1 * szam2 = {}".format(szam1*szam2))
elif mjel == ("/"):
    print("szam1 / szam2 = {}".format(szam1/szam2))
else:
    print("érvénytelen műveleti jel! ")    #OK
```

"""

2. feladat: Készítsen programot a következő feladatok megoldására, amelynek a forráskódját `atlag` néven mentse el! Hozzon létre egy listát, majd töltse fel 15 darab egész véletlenszámmal a [-20,20] intervallumból! Készítsen `atlagfv()` néven egy függvényt, amely paraméterül megkapja az előzőekben feltöltött 15 elemű listát! A függvény a pozitív számok átlagával térjen vissza! Az eredmény 2 tizedesjegy pontossággal jelenjen meg!

"""

```
import random  #meghívjuk a random függvényt
lista = []
for i in range(0, 15):  #ezzel 15 db indexet állítunk elő
    lista.append(random.randint(-20, 20)) #ezzel megadjuk, hogy a 15 szám
                                          -20 és 20 közötti legyen
print(lista) #kiíratjuk a random listánkat (ez mindenkinél más-más lesz)!

def atlagfv(lista):
    osszeg = 0
    darab = 0
    for i in range(len(lista)):
        if lista[i] > 0:
            osszeg = osszeg + lista[i]
            darab = darab + 1
    return round(osszeg / darab, 2)  # a raund a kerekítő függvény a 2-es
                                       jelzi hogy hány tizedesig

print("A pozitív számok összege: {}".format(atlagfv(lista)))  #OK
```

"""

3. feladat: Összetett feladat több részfeladattal, melyeket a feladat során külön ki kell írni!

a. Olvassa be a .txt fájl adatait és tárolja el egy adatszerkezetben, ami a további feladatokmegoldására alkalmas. A fájlban legfeljebb 1000 sor lehet! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!

"""

```
class hegyek:      #létrehozunk egy osztályt, ahova beolvassuk az adatokat.
    def __init__(self, line): #speciális metódus, beolvasáshoz alkalmazzuk!!
        darabol = line.split(";")
        self.hegycsucsNeve = darabol[0]
        self.hegyseg = darabol[1]
        self.magassag = int(darabol[2].rstrip("\n"))

lista = []
with open("hegyekMo.txt", "rt", encoding="utf-8") as file: #az elérési út
                                    utáni rt az olvasásra megnyitást adja meg,
                                    majd a dekódolás
    file.readline()  #csak akkor kell ha van fejléc a beolvasott anyagban!!
    lines = file.readlines()
    for i in range(0, len(lines)):
        lista.append(hegyek(lines[i]))


#Beolvasás ellenőrzése érdemes elvégezni, de nem kötelező! Ha működik
kommenteljük ki!
for i in range(0, len(lista)):
    print("{} {} {}".format(lista[i].hegycsucsNeve,
                    lista[i].hegyseg,
                    lista[i].magassag))    #OK
```

”””

b. Határozza meg és írja ki a képernyőre hány hegy található az állományban!

”””

```
print ("b. feladat: Hegycsúcsok száma: {} db".format(len(lista)))  #OK
```

”””

c. Határozza meg és írja ki az állományban található hegyek átlagmagasságát!

”””

```
osszesMagassag = 0   # létrehozunk egy új változót
for i in range(0, len(lista)):
    osszesMagassag = osszesMagassag + lista[i].magassag
atlag = round(osszesMagassag / len(lista), 2)# 2 tizedesre kerekítve
print("c. feladat: Hegycsúcsok átlagmagassága: {} m".format(atlag))  #OK
```

”””

d. Határozza meg és írja ki a legmagasabb hegy adatait! Feltételezheti, hogy nem alakult ki holtverseny.

”””

```
maxIndex = 0   #bezetjük a maxIndex változót, aminek a kezdő értéke 0
for i in range(1, len(lista)):  # a bejáró ciklussal 1-től végig megy a lista
                                minden elemén
    if lista[i].magassag > lista[maxIndex].magassag:
        maxIndex = i
print("5. feladat: A legmagasabb hegycsúcs adatai: ")
print("\tNév: {}".format(lista[maxIndex].hegycsucsNeve)) # a \t a behúzást
                                                          adja meg
print("\tHegység: {}".format(lista[maxIndex].hegyseg))
print("\tMagasság: {}".format(lista[maxIndex].magassag))   #OK
```

”””

e. Kérjen be a felhasználótól egy magasságértéket! Döntse el, hogy a Börzsöny hegységben található-e a megadott értéknél magasabb hegycsúcs! A keresést ne folytassa, ha a választ meg tudja adni! Az eredményt írja ki!

”””

```
print("e. feladat: ")
m = int(input("Kérem adjon meg egy magasságértéket: "))
for i in range(0, len(lista)):
    if lista[i].hegyseg == "Börzsöny" and lista[i].magassag > m:
        print("Van magasabb hegység a Börzsönyben!")
        break    #ezzel kilépünk a ciklusból mert nem kell tovább vizsgálni
else:
    print("Nincs magasabb hegy a Börzsönyben!")   #OK
```

”””

f. Határozza meg és írja ki azoknak a hegycsúcsoknak a számát, melyek 3000 lábnál magasabbak! Az átváltáshoz az 1 m = 3.280839895 láb értékkel dolgozzon!

”””

```
db = 0
for i in range(0, len(lista)):
    if lista[i].magassag * 3.280839895 > 3000:
        db = db + 1
print("f. feladat: 3000 lábnál magasabb hegycsúcsok száma: {}".format(db))#OK
```