

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1
Faktoriális
 - ◆ 1.2
Tökéletes
számok
 - ◆ 1.3
Válogatás
 - ◆ 1.4
CloudCoder
 - ◇ 1.4.1
lista_elso
 - ◇ 1.4.2
lista_veg
 - ◇ 1.4.3
szin_komponens
 - ◇ 1.4.4
sorszamok
 - ◇ 1.4.5
naplobol
 - ◇ 1.4.6
legnagyobb
 - ◇ 1.4.7
udvarias
 - ◇ 1.4.8
szorzo
 - ◇ 1.4.9
neptun_elobb

Feladatok

Faktoriális

Adjuk meg a bemeneti szám faktoriálisát! (Bemenet:n, kimenet n!)

```
n = input()
p = 1
for i in range(1, n + 1):
    p = p * i
print p
```

Tökéletes számok

Írjunk programot, mely bekér egy pozitív egész számot és leellenőrzi, hogy tökéletes szám-e.

```
n = input()
s = 0
for i in range(1, n):
    if n % i == 0:
        s += i
if n == s:
    print "Tokelotes"
else:
```

```
print "Nem tokeletes"
```

Válogatás

Adott egy számokat tartalmazó L listánk, írjunk programot, mely két listába válogatja L elemeit, negatívakat az egyikbe, nem negatívakat a másikba.

```
L = [-1, 2, 5, -2, 3, -4, -5, 2, -2, 0, 5, 5, 6, 3, -3]
```

```
L = [-1, 2, 5, -2, 3, -4, -5, 2, -2, 0, 5, 5, 6, 3, -3]
```

```
neg = []
nemneg = []
for e in L:
    if e < 0:
        neg.append(e)
    else:
        nemneg.append(e)
print neg, nemneg
```

CloudCoder

lista_elso

```
def lista_elso(l):
    if len(l) > 0:
        return l[0]
    else:
        return None
```

lista_veg

```
def lista_veg(l, db):
    if len(l) >= db:
        return l[len(l) - db:]
    else:
        return None
```

szin_komponens

```
def szin_komponens(szin, komponens):
    if komponens == "piros":
        return szin[0]
    if komponens == "zold":
        return szin[1]
    if komponens == "kek":
        return szin[2]
```

sorszamok

```
def sorszamok(indulok):
    return range(1, len(indulok) + 1)
```

naplobol

```
def naplobol(nevek, sorszam):
    sorban = sorted(nevek)
    return sorban[sorszam]
```

legnagyobb

```
def legnagyobb(l):  
    nagy = l[0]  
    for e in l:  
        if e > nagy:  
            nagy = e  
    return nagy
```

udvarias

```
def udvarias(l):  
    uj = []  
    for nev in l:  
        uj.append("Ms. " + nev)  
    return uj
```

SZORZO

```
def szorzo(l, k):  
    uj = []  
    for e in l:  
        uj.append(e * k)  
    return uj
```

neptun_elobb

```
def neptun_elobb(mienk, aktualis):  
    if mienk < aktualis:  
        return True  
    else:  
        return False
```