

Prepovedani intervali

Najprej moram povedati: naloga ni moja. Dobil sem jo na Advent of code iz leta 2016.

Imamo seznam prepovedanih intervalov števil, na primer

```
prepovedani = [  
    (12, 18),  
    (2, 5),  
    (3, 8),  
    (0, 4),  
    (15, 19),  
    (6, 9),  
    (13, 17),  
    (4, 8)  
]
```

To pomeni, da so prepovedana vsa števila od 12 do 18 (vključno z 12 in 18), vsa števila od 2 do 5 in tako naprej.

1. Napiši program, ki izpiše največjo zgornjo mejo iz seznama. V gornjem primeru torej izpiše 19.
2. Nato pa napiši program, ki za vsa števila od 0 do največje zgornje meje izpiše prvi interval, ki to število prepoveduje. Če število ni prepovedano, pa naj to pove. Konkretno, program naj izpiše tole:

```
0 je vsebovan v (0, 4)  
1 je vsebovan v (0, 4)  
2 je vsebovan v (2, 5)  
3 je vsebovan v (2, 5)  
4 je vsebovan v (2, 5)  
5 je vsebovan v (2, 5)  
6 je vsebovan v (3, 8)  
7 je vsebovan v (3, 8)  
8 je vsebovan v (3, 8)  
9 je vsebovan v (6, 9)  
10 je dovoljeno  
11 je dovoljeno  
12 je vsebovan v (12, 18)  
13 je vsebovan v (12, 18)  
14 je vsebovan v (12, 18)  
15 je vsebovan v (12, 18)  
16 je vsebovan v (12, 18)  
17 je vsebovan v (12, 18)  
18 je vsebovan v (12, 18)  
19 je vsebovan v (15, 19)
```

Pazi: vsako število je izpisano le ob prvem intervalu, ki ga prepoveduje, četudi je morda prepovedano v večih.

Dodatna naloga

Napiši program, ki izpiše prvo dovoljeno število. V gornjem primeru izpiše 10.

Seznam intervalov spremeni tako, da dodaš še interval (9, 12). Program naj v tem primeru izpiše 20, saj so vsa števila do 19 prepovedana.

Rešitev

Iskanje največje zgornje meje v seznamu intervalov je podobno iskanju največjega števila v seznamu števil. Le da imamo tu pare števil.

```
meja = 0

for od, do in prepovedani:
    if do >= meja:
        meja = do
```

```
print(meja)
```

19

Zdaj pa moramo čez vsa števila od 0 do 19 (oz. do `meja` in za vsako izpisati interval, ki ga prepoveduje (ali pa, če ne najdemo nobenega, povedati, da je dovoljeno. Od 0 do `meja` znamo šteti.

```
i = 0
while i <= meja:
    ...
    i += 1
```

Namesto ... pa moramo poiskati prvi interval, ki prepoveduje to število, ali pa povedati, da ni prepovedano. To pa je praktično enako iskanju prvega lihega števila v seznamu.

```
i = 0
while i <= meja:
    for od, do in prepovedani:
        if od <= i <= do:
            print(i, "je vsebovan v", (od, do))
            break
    else:
        print(i, "je dovoljeno")
    i += 1
```

Dodatna naloga v bistvu zahteva, da si zapomnimo prvo dovoljeno število. To bomo storili znotraj `else`. Ena možnost je, da dopolnimo prejšnjo zanko.

```

i = 0
prvo_dovoljeno = meja + 1

while i <= meja:
    for od, do in prepovedani:
        if od <= i <= do:
            print(i, "je vsebovan v", (od, do))
            break
        else:
            print(i, "je dovoljeno")
            if prvo_dovoljeno == meja + 1:
                prvo_dovoljeno = i
    i += 1

print("Prvo dovoljeno število:", prvo_dovoljeno)

```

Za začetek se pesimistično delamo, da bo prvo dovoljeno število šele onstran meje. Če kasneje naletimo na kaj boljšega, zgrabimo.

Če ne želimo spreminjati gornjega programa, pa lahko navozlamo nekaj bolj zanimivega.

```

i = 0
while i <= meja:
    for od, do in prepovedani:
        if od <= i <= do:
            break
        else:
            prvo_dovoljeno = i
            break
    i += 1
else:
    prvo_dovoljeno = meja + 1

print("Prvo dovoljeno število:", prvo_dovoljeno)

```

Prvo dovoljeno število: 10

Zanka **while** se *prekine* z **break** prvič, ko se zanka **for** *ne prekine* z **break**. Če pa se zanka **while** nikoli ne prekine, je prvo dovoljeno število onstran meje.

Ta postopek za iskanje prvega dovoljenega števila ni prav hiter. V dodatnem izzivu imamo primer, ko imamo veliko intervalov in tudi prvo dovoljeno število precej veliko. Nalogo se da rešiti lepše, hitrejše.