

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



python

#3

LIGA PROGRAMIRANJA U PYTHONU ZA

OSNOVNE ŠKOLE – 4. RADIONICA

Tomo Sjekavica, Mario Miličević *Informatički klub FUTURA*
Dubrovnik, 21. siječnja 2017.



Dubrovnik

Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/>

Creative Commons



- **slobodno smijete:**
 - **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
 - **remiksirati** — prerađivati djelo
- **pod slijedećim uvjetima:**
 - **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
 - **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
 - **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

Raspored Lige programiranja

- ...
- 21.01.2017 – **4. radionica**
- 04.02.2017 – **3. kolo Lige programiranja**
- 18.02.2017 – 5. radionica
- 04.03.2017 – 4. kolo Lige programiranja
- ...

- Web stranica Lige programiranja:

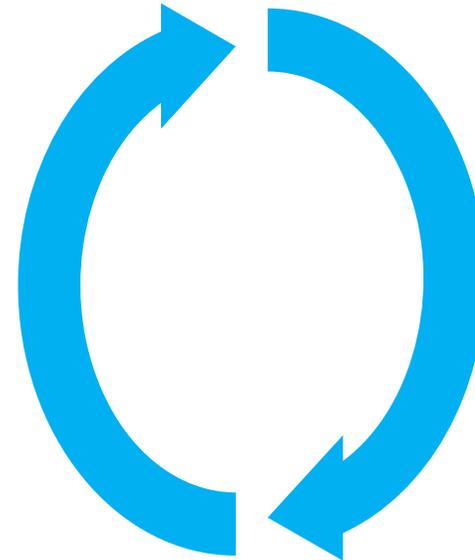
www.futura.com.hr/liga-programiranja-u-pythonu-2016-2017/



Ponavljanje gradiva s 3. radionice

- Ponavljanje bloka naredbi:

```
...  
for i in range(n):  
    naredba_1  
    ...  
    naredba_z  
...
```



Programska petlja

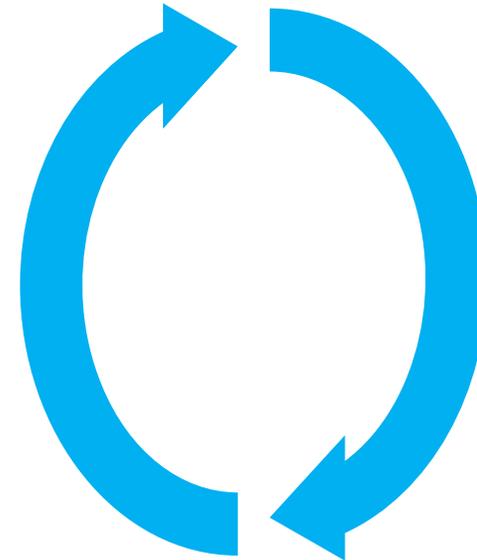
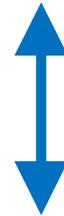
Blok naredbi će se izvesti **n puta**, za vrijednosti varijable **i** od **0** do **n-1**.



Ponavljanje gradiva s 3. radionice

- Ponavljanje bloka naredbi:

```
...  
for i in range(n,m):  
    naredba_1  
    ...  
    naredba_z  
...
```



Programska petlja

Blok naredbi će se izvesti **m-n puta**, za sve vrijednosti varijable **i** od **n** do **m-1**.



Ponavljjanje gradiva s 3. radionice

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
>>> for i in range(5):
      print(i)

0
1
2
3
4
>>>
```

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
>>> for i in range(2,7):
      print(i)

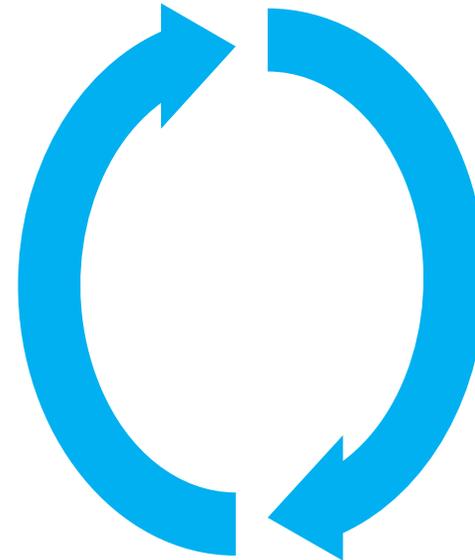
2
3
4
5
6
>>>
Ln: 29 Col: 4
```



Ponavljanje gradiva s 3. radionice

- Ponavljanje bloka naredbi:

```
...  
for i in range(n,m,k):  
    naredba_1  
    ...  
    naredba_z  
...
```



Programska petlja

Blok naredbi će se izvesti **m-n puta**, za vrijednosti varijable **i** od **n** do **m-1**, ali će se **i** svaki put uvećati za **k**.



Ponavljjanje gradiva s 3. radionice

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
>>> for i in range(5,30,5):
      print(i)

5
10
15
20
25
>>>
```

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
>>> for i in range(10,0,-2):
      print(i)

10
8
6
4
2
>>>
```

Negativni korak -
odbrojavanje
unatrag

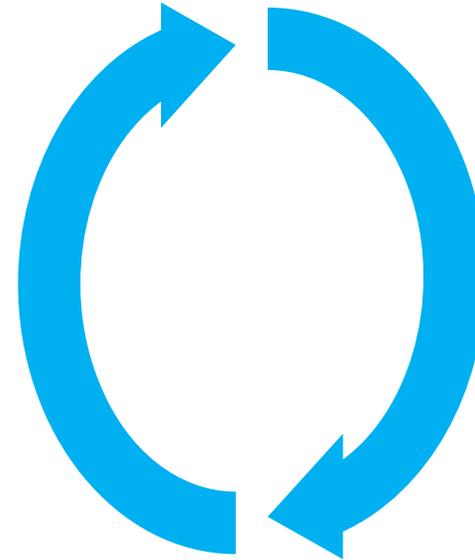
Ln: 49 Col: 4



Ponavljanje gradiva s 3. radionice

- Uvjetno ponavljanje bloka naredbi:

```
...  
while uvjet:  
    naredba_1  
    ...  
    naredba_i  
...
```



Uvjetna programska petlja

Blok naredbi će se izvoditi dok je **uvjet** ispunjen (daje vrijednost **True**). Potrebno **inicijalizirati** brojač prije petlje, a u petlji ga **povećavati** (**smanjivati**).



Ponavljjanje gradiva s 3. radionice

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
>>> a = 1
>>> while a < 10:
    print(a)
    a = a + 2

1
3
5
7
9
>>>
```

Ln: 62 Col: 4

Zadatak: Odbrojavanje



- Napisati program u kojem se unosi početno vrijeme u minutama i sekundama, te se ispisuje odbrojavanje do nule samo u sekundama (5./6. razredi), ili u formatu ukupne sekunde – minute : sekunde (7./8. razredi). Npr.

Vrijeme



| (5./6. razredi) | (7./8. razredi) |
|------------------|------------------|
| Unesi minute: 3 | Unesi minute: 3 |
| Unesi sekunde: 2 | Unesi sekunde: 2 |
| 182 | 182 – 3 : 2 |
| 181 | 181 – 3 : 1 |
| 180 | 180 – 3 : 0 |
| 179 | 179 – 2 : 59 |
| ... | ... |
| 2 | 2 – 0 : 2 |
| 1 | 1 – 0 : 1 |
| 0 | 0 – 0 : 0 |



Odbrojavanje – rješenje (5./6.)

```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
while sekunde >= 0:
    print(sekunde)
    sekunde = sekunde-1
```



odbrojavanje5-6.py

| Window 1 (Input) | Window 2 | Window 3 | Window 4 | Window 5 | Window 6 (Output) |
|--------------------------|----------|----------|----------|----------|-------------------|
| Unesite broj minuta: 1 | 64 | 51 | 38 | 25 | 12 |
| Unesite broj sekundi: 15 | 63 | 50 | 37 | 24 | 11 |
| 75 | 62 | 49 | 36 | 23 | 10 |
| 74 | 61 | 48 | 35 | 22 | 9 |
| 73 | 60 | 47 | 34 | 21 | 8 |
| 72 | 59 | 46 | 33 | 20 | 7 |
| 71 | 58 | 45 | 32 | 19 | 6 |
| 70 | 57 | 44 | 31 | 18 | 5 |
| 69 | 56 | 43 | 30 | 17 | 4 |
| 68 | 55 | 42 | 29 | 16 | 3 |
| 67 | 54 | 41 | 28 | 15 | 2 |
| 66 | 53 | 40 | 27 | 14 | 1 |
| 65 | 52 | 39 | 26 | 13 | 0 |



Odbrojavanje – rješenje (7./8.)

```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
while sekunde >= 0:
    print(sekunde, '-', sekunde//60, ':',
          sekunde%60)
    sekunde = sekunde-1
```



odbrojavanje7-8.py

| Python 3.4.1 Shell |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Unesite broj | 67 - 1 : 7 | 57 - 0 : 57 | 47 - 0 : 47 | 37 - 0 : 37 | 27 - 0 : 27 | 17 - 0 : 17 | 7 - 0 : 7 |
| Unesite broj | 66 - 1 : 6 | 56 - 0 : 56 | 46 - 0 : 46 | 36 - 0 : 36 | 26 - 0 : 26 | 16 - 0 : 16 | 6 - 0 : 6 |
| 75 - 1 : 15 | 65 - 1 : 5 | 55 - 0 : 55 | 45 - 0 : 45 | 35 - 0 : 35 | 25 - 0 : 25 | 15 - 0 : 15 | 5 - 0 : 5 |
| 74 - 1 : 14 | 64 - 1 : 4 | 54 - 0 : 54 | 44 - 0 : 44 | 34 - 0 : 34 | 24 - 0 : 24 | 14 - 0 : 14 | 4 - 0 : 4 |
| 73 - 1 : 13 | 63 - 1 : 3 | 53 - 0 : 53 | 43 - 0 : 43 | 33 - 0 : 33 | 23 - 0 : 23 | 13 - 0 : 13 | 3 - 0 : 3 |
| 72 - 1 : 12 | 62 - 1 : 2 | 52 - 0 : 52 | 42 - 0 : 42 | 32 - 0 : 32 | 22 - 0 : 22 | 12 - 0 : 12 | 2 - 0 : 2 |
| 71 - 1 : 11 | 61 - 1 : 1 | 51 - 0 : 51 | 41 - 0 : 41 | 31 - 0 : 31 | 21 - 0 : 21 | 11 - 0 : 11 | 1 - 0 : 1 |
| 70 - 1 : 10 | 60 - 1 : 0 | 50 - 0 : 50 | 40 - 0 : 40 | 30 - 0 : 30 | 20 - 0 : 20 | 10 - 0 : 10 | 0 - 0 : 0 |
| 69 - 1 : 9 | 59 - 0 : 59 | 49 - 0 : 49 | 39 - 0 : 39 | 29 - 0 : 29 | 19 - 0 : 19 | 9 - 0 : 9 | >>> |
| 68 - 1 : 8 | 58 - 0 : 58 | 48 - 0 : 48 | 38 - 0 : 38 | 28 - 0 : 28 | 18 - 0 : 18 | 8 - 0 : 8 | >>> |

Funkcija split



- Funkcija **split** vraća listu riječi iz zadanog niza znakova (standardni razdjelnik je praznina ' ')

```
>>> tekst = 'Liga programiranja u Pythonu'  
>>> tekst.split()  
['Liga', 'programiranja', 'u', 'Pythonu']
```

- Korisnik može kod poziva funkcije **split** postaviti razdjelnik po želji

```
>>> tekst = 'Liga programiranja u Pythonu'  
>>> tekst.split('a')  
['Lig', ' progr', 'mir', 'nj', ' u Pythonu']
```

Varijable i višestruko pridruživanje



- Višestruko pridruživanje vrijednosti varijablama

```
>>> broj1, broj2 = 1, 5
>>> print('broj1 =', broj1, 'broj2 =', broj2)
broj1 = 1 broj2 = 5
```

- Višestruko pridruživanje vrijednosti varijablama sa funkcijom **split**

```
>>> tekst = 'Liga programiranja u Pythonu'
>>> t1, t2, t3, t4 = tekst.split()
>>> print('t1', t1, 't2', t2, 't3', t3, 't4', t4)
t1 Liga t2 programiranja t3 u t4 Pythonu
```

Zadatak: Odbrojavanje split



- Nadograditi zadatak odbrojavanje tako da se vrijeme unosi u jednoj liniji u formatu mm:ss. Npr. 3:2
- Kod prethodnog zadatka Odbrojavanje:

Vrijeme



```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
```

```
while sekunde >= 0:
    print(sekunde, '-', sekunde//60, ':',
          sekunde%60)
    sekunde = sekunde-1
```



odbrojavanje7-8.py



Odbrojavanje split – rješenje

```
vrijeme = input('Unesite vrijeme: ')\nminute, sekunde = vrijeme.split(':')
```

```
minute = int(minute)\nsekunde = int(sekunde)
```

```
sekunde = minute*60 + sekunde
```

```
while sekunde >= 0:\n    print(sekunde, '-', sekunde//60, ':',\n          sekunde%60)\n    sekunde = sekunde-1
```

Uneseno vrijeme se sprema u varijable minute i sekunde, razdjelnik je ':'.

Pretvaranje minuta i sekunda iz niza znakova u cijele brojeve.



odbrojavanje_split.py

Liste



- Kako stvoriti niz od 10 cijelih brojeva?
- 1. pokušaj (10 varijabli s 10 vrijednosti):

```
>>> br1, br2, br3, br4, br5, br6, br7, br8, br9, br10 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

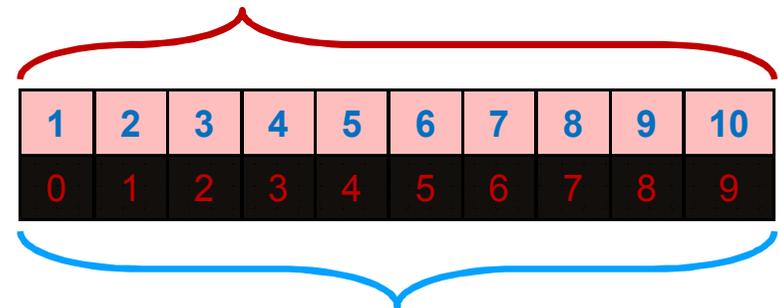


- 2. pokušaj:

```
>>> brojevi = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> print(brojevi)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



Brojevi (elementi polja).



Indeksi elemenata polja.
Indeksi idu od 0 do n-1
(u ovom primjeru do 10-1 = 9).

Pristup elementima liste



- Elementima liste se može pristupiti preko indeksa elementa liste

```
>>> brojevi[0]
1
>>> brojevi[1]
2
>>> brojevi[9]
10
>>> brojevi[10]
```

GREŠKA: u listi **brojevi** ne postoji element s indeksom **10**, zadnji element liste s 10 brojeva ima indeks **9**.

```
Traceback (most recent call last):
  File "<pyshell#89>", line 1, in <module>
    brojevi[10]
IndexError: list index out of range
```

Unos elemenata liste s tipkovnice

```
lista = [0] * 5
```

Deklaracija liste koja sadrži 5 elemenata i inicijalizacija svih elemenata na vrijednost 0.

```
for i in range(5):  
    lista[i] = float(input('Unesite broj: '))
```



lista.py

```
for i in range(5):  
    print(i, 'element liste:', lista[i])
```

Unos brojeva s tipkovnice i spremanje u listu pomoću for petlje.

i = 5
kraj petlje

Pristup elementima liste pomoću for petlje i ispis vrijednosti na ekran.

| | | | | |
|-----|-----|-----|-----|-----|
| 1.1 | 2.2 | 3.3 | 4.4 | 5.5 |
| 0 | 1 | 2 | 3 | 4 |

```
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Unesite broj: 1.1  
Unesite broj: 2.2  
Unesite broj: 3.3  
Unesite broj: 4.4  
Unesite broj: 5.5  
0 element liste: 1.1  
1 element liste: 2.2  
2 element liste: 3.3  
3 element liste: 4.4  
4 element liste: 5.5  
Ln: 926 Col: 4
```

Zadatak: Zbroj neparnih elem.



- Napisati program koji zbraja svaki drugi element liste cijelih brojeva, počevši od prvog elementa. Broj elemenata liste i vrijednosti elemenata liste se unose s tipkovnice.

| | | | | |
|--------------|--------------|--------------|--------------|--------------|
| <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> |
| 5 | 4 | 5 | 3 | 4 |
| 3 | 32 | 74 | 48 | 971 |
| 15 | 43 | 18 | 264 | 453 |
| 4 | 17 | 5 | 530 | 415 |
| 8 | 5 | 27 | | 544 |
| 9 | | 22 | | |
| <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> |
| 16 | 49 | 101 | 578 | 1368 |

Vrijeme





Zbroj neparnih elem. – rješenje

```
n = int(input('Unesi broj elemenata liste: '))
```

```
lista = [0] * n
```

Deklaracija liste koja sadži **n** elemenata i inicijalizacija svih elemenata na vrijednost 0.

```
for i in range(n):
```

```
    lista[i] = int(input('Unesite broj: '))
```

```
zbroj = 0
```

```
for i in range(0, n, 2):
```

```
    zbroj = zbroj + lista[i]
```

```
print(Zbroj neparnih elemenata liste: ', zbroj)
```



lista_zbr_neparnih.py

Zadatak: Košarka



- Igrači školske košarkaške ekipe imaju brojeve od 4 do 15.
- Na utakmici se u svakom uspješnom napadu bilježi broj igrača koji je postigao koš, kao i koliko poena vrijedi taj pogodak (1, 2 ili 3). Podaci se unose tako da je broj igrača i broj poena odvojen znakom '-'.
- Unos podataka se prekida (kraj utakmice) kad se unese prazan string.

Vrijeme



Zadatak: Košarka



- Ispisati broj igrača koji je postigao najviše koševa.
- Testni podaci:

| | | |
|--|--|--|
| <u>Ulaz</u> 5-2 7-3 15-2 11-2 5-2 11-3 12-2 | <u>Ulaz</u> 4-1 6-2 7-3 14-2 4-1 4-1 7-3 4-2 | <u>Ulaz</u> 8-3 11-2 13-3 10-1 10-2 12-3 11-2 |
| <u>Izlaz</u> Igrač br. 11 | <u>Izlaz</u> Igrač br. 7 | <u>Izlaz</u> Igrač br. 11 |

Vrijeme





Košarka – rješenje

Deklaracija liste s 16 elemenata - koji će imati indekse od 0 do 15

```
kosevi = [0] * 16  
ulaz = input('Unesi broj igrača i broj koševa: ')
```

```
while ulaz != '':  
    igr, kos = ulaz.split('-')  
    igr = int(igr)  
    kos = int(kos)  
    kosevi[igr] = kosevi[igr] + kos  
    ulaz = input('Unesi broj igrača i broj koševa: ')
```

Unos podataka u listu - broj igrača odgovara indeksu elementa

```
naj_kos, naj_igr = 0, 0  
for i in range(4, 16):  
    if kosevi[i] > naj_kos:  
        naj_kos = kosevi[i]  
        naj_igr = i
```

Čitanje liste - broj igrača odgovara indeksu elementa. Elementi s indeksima od 0 do 3 se ne čitaju jer nema igrača s tim brojevima



kosarka.py

```
print('Najviše koševa je postigao igrač br.', naj_igr)
```

Zadatak: Rođendan



- Ivo i Tea zajedno slave rođendan. Dogovorili su se da će svaki od njih pozvati po N prijatelja.
- Potrebno je prvo unijeti prirodni broj N . Zatim prvo za Iva, a onda za Teu unijeti imena pozvanih prijatelja. Koristiti dvije liste.
- Ispisati koliko ukupno prijatelja su pozvali Ivo i Tea, ali tako da se isto ime ne broji 2 puta.

Vrijeme



Zadatak: Rođendan



□ Testni podaci:

| | | | |
|---|--------------------------------------|---|---|
| <u>Ulaz</u> 3 Ane Mira <u>Ante</u> Pero <u>Ante</u> Kate | Ivo Tea | <u>Ulaz</u> 4 Mia Lea Ivan Jure Lea Jure Ane Mia | <u>Ulaz</u> 4 Pavo Teo Ane Iva Ane Teo Kate Luce |
| <u>Izlaz</u> Dolazi 5 prijatelja. | <u>Izlaz</u> Dolazi 5 prijatelja. | <u>Izlaz</u> Dolazi 6 prijatelja. | |

Vrijeme





Rođendan – rješenje



rodjendan.py

```
N = int(input('Unesi N: '))
ivo = [''] * N
tea = [''] * N
```

Deklaracija dviju lista s po **N** elemenata - posebno za Ivove, odnosno za Teine goste

```
for i in range(N):
    ivo[i] = input('Ivov prijatelj: ')
for i in range(N):
    tea[i] = input('Tein prijatelj: ')
```

Unos prvo Ivovih **N** prijatelja, a zatim Teinih **N** prijatelja

```
broj = 2 * N
for i in range(N):
    for j in range(N):
        if ivo[i] == tea[j]:
            broj = broj - 1
```

Kreće se od pretpostavke da će ukupno biti **2*N** gostiju

```
print('Dolazi', broj, 'prijatelja.')
```

Za svako ime s Ivove liste provjerava se postoji li isto ime na Teinoj listi. Ako postoji broj gostiju se umanjuje za 1

Zadatak: Reflektori



- Na igralištu se nalazi n reflektora. Svaki reflektor može biti u stanju ugašen (broj **0**) ili upaljen (broj **1**). Na početku su svi reflektori ugašeni.
- Napisati program u kojem se unosi broj reflektora, pa zatim parovi vrijednosti broj reflektora i stanje (**0** ili **1**), sve dok se ne unese kombinacija **0 0** ili **0 1**.
- Ispisati koliko je reflektora upaljeno, a koliko ih je ugašeno

Vrijeme



Zadatak: Reflektori



□ Testni podaci:

| | | | | |
|--------------|--------------|--------------|--------------|--------------|
| <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> | <u>Ulaz</u> |
| 10 | 15 | 5 | 10 | 20 |
| 1 1 | 13 1 | 1 1 | 2 1 | 19 1 |
| 5 0 | 9 0 | 2 1 | 8 1 | 5 1 |
| 6 1 | 7 1 | 3 1 | 9 1 | 3 1 |
| 3 0 | 2 1 | 5 1 | 8 0 | 20 1 |
| 4 1 | 6 1 | 5 0 | 3 0 | 13 0 |
| 8 1 | 7 0 | 4 1 | 4 1 | 14 1 |
| 0 0 | 10 1 | 5 1 | 9 0 | 8 0 |
| | 14 1 | 0 1 | 2 0 | 4 1 |
| | 0 1 | | 4 0 | 11 1 |
| | | | 0 0 | 5 0 |
| | | | | 0 0 |
| <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> | <u>Izlaz</u> |
| 4 6 | 6 9 | 5 0 | 0 10 | 6 14 |

Vrijeme





Reflektori– rješenje

```
n = int(input('Unesi broj reflektora: '))
lista = [0] * (n + 1)
rbr = 1
onoff = 0
on = 0
while rbr != 0:
    unos = input('Unesite broj reflektora i 0/1: ')
    rbr, onoff = unos.split()
    rbr = int(rbr)
    onoff = int(onoff)
    lista[rbr] = onoff
for i in range (1, n+1):
    if lista[i] == 1:
        on = on + 1
print('Upaljeno', on, 'a ugašeno', n-on, 'reflektora.')
```



reflektori.py

Ne zaboravite!

- Za 15 dana – u subotu **04.02.2017**
3. kolo Lige programiranja
- Početak : 10:00**
- 3 zadatka rješavate 75 minuta**
- nemojte kasniti!**

