



PRAKTIKUM IZ PROGRAMIRANJA 3

VEŽBE 7

Lazar Vasović, Luka Marković,
Andreja Živić, Jovan Janićijević

CSV(comma-separated values)

-Svaka linija u csv fajlu predstavlja jedan red u tabeli, a zarezi su razdvojene vrednosti te tabele

1	"LatD",	"LatM",	"LatS",	"NS",	"LonD",	"LonM",	"LonS",	"EW",	"City",	"State"
2	41,	5,	59,	"N",	80,	39,	0,	"W",	"Youngstown",	OH
3	42,	52,	48,	"N",	97,	23,	23,	"W",	"Yankton",	SD
4	46,	35,	59,	"N",	120,	30,	36,	"W",	"Yakima",	WA
5	42,	16,	12,	"N",	71,	48,	0,	"W",	"Worcester",	MA
6	43,	37,	48,	"N",	89,	46,	11,	"W",	"Wisconsin Dells",	WI
7	36,	5,	59,	"N",	80,	15,	0,	"W",	"Winston-Salem",	NC
8	49,	52,	48,	"N",	97,	9,	0,	"W",	"Winnipeg",	MB
9	39,	11,	23,	"N",	78,	9,	36,	"W",	"Winchester",	VA
10	34,	14,	24,	"N",	77,	55,	11,	"W",	"Wilmington",	NC
11	39,	45,	0,	"N",	75,	33,	0,	"W",	"Wilmington",	DE
12	48,	9,	0,	"N",	103,	37,	12,	"W",	"Williston",	ND
13	41,	15,	0,	"N",	77,	0,	0,	"W",	"Williamsport",	PA
14	37,	40,	48,	"N",	82,	16,	47,	"W",	"Williamson",	WV
15	33,	54,	0,	"N",	98,	29,	23,	"W",	"Wichita Falls",	TX
16	37,	41,	23,	"N",	97,	20,	23,	"W",	"Wichita",	KS
17	40,	4,	11,	"N",	80,	43,	12,	"W",	"Wheeling",	WV
18	26,	43,	11,	"N",	80,	3,	0,	"W",	"West Palm Beach",	FL
19	47,	25,	11,	"N",	120,	19,	11,	"W",	"Wenatchee",	WA
20	41,	25,	11,	"N",	122,	23,	23,	"W",	"Weed",	CA
21	31,	13,	11,	"N",	82,	20,	59,	"W",	"Waycross",	GA
22	44,	57,	35,	"N",	89,	38,	23,	"W",	"Wausau",	WI
23	42,	21,	36,	"N",	87,	49,	48,	"W",	"Waukegan",	IL
24	44,	54,	0,	"N",	97,	6,	36,	"W",	"Watertown",	SD
25	43,	58,	47,	"N",	75,	55,	11,	"W",	"Watertown",	NY
26	42,	30,	0,	"N",	92,	20,	23,	"W",	"Waterloo",	IA
27	41,	32,	59,	"N",	73,	3,	0,	"W",	"Waterbury",	CT
28	38,	53,	23,	"N",	77,	1,	47,	"W",	"Washington",	DC
29	41,	50,	59,	"N",	79,	8,	23,	"W",	"Warren",	PA
30	46,	4,	11,	"N",	118,	19,	48,	"W",	"Walla Walla",	WA
31	31,	32,	59,	"N",	97,	8,	23,	"W",	"Waco",	TX
32	38,	40,	48,	"N",	87,	31,	47,	"W",	"Vincennes",	IN
33	28,	48,	35,	"N",	97,	0,	36,	"W",	"Victoria",	TX
34	32,	20,	59,	"N",	90,	52,	47,	"W",	"Vicksburg",	MS

Normal text file length: 8,530 lines: 130 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

	A	B	C	D	E	F	G	H	I	J	K	L
1	LatD	"LatM"	"LatS"	"NS"	"LonD"	"LonM"	"LonS"	"EW"	"City"	"State"		
2	41	5	59	"N"	80	39	0	"W"	"Youngsto	OH		
3	42	52	48	"N"	97	23	23	"W"	"Yankton"	SD		
4	46	35	59	"N"	120	30	36	"W"	"Yakima"	WA		
5	42	16	12	"N"	71	48	0	"W"	"Worcest	MA		
6	43	37	48	"N"	89	46	11	"W"	"Wiscons	WI		
7	36	5	59	"N"	80	15	0	"W"	"Winston	NC		
8	49	52	48	"N"	97	9	0	"W"	"Winnipe	MB		
9	39	11	23	"N"	78	9	36	"W"	"Winches	VA		
10	34	14	24	"N"	77	55	11	"W"	"Wilming	NC		
11	39	45	0	"N"	75	33	0	"W"	"Wilming	DE		
12	48	9	0	"N"	103	37	12	"W"	"Willistor	ND		
13	41	15	0	"N"	77	0	0	"W"	"Williams	PA		
14	37	40	48	"N"	82	16	47	"W"	"Williams	WV		
15	33	54	0	"N"	98	29	23	"W"	"Wichita	TX		
16	37	41	23	"N"	97	20	23	"W"	"Wichita"	KS		
17	40	4	11	"N"	80	43	12	"W"	"Wheelin	WV		
18	26	43	11	"N"	80	3	0	"W"	"West Pal	FL		
19	47	25	11	"N"	120	19	11	"W"	"Wenatch	WA		
20	41	25	11	"N"	122	23	23	"W"	"Weed"	CA		
21	31	13	11	"N"	82	20	59	"W"	"Waycros	GA		
22	44	57	35	"N"	89	38	23	"W"	"Wausau"	WI		
23	42	21	36	"N"	87	49	48	"W"	"Waukeg	IL		
24	44	54	0	"N"	97	6	36	"W"	"Waterto	SD		
◀ ▶ gradovi ⊕												



-čitanje celog fajla

```
import csv

gradoviFile = open('gradovi.csv')
gradoviReader = csv.reader(gradoviFile)
gradovi = list(gradoviReader)

print(gradovi[0])    #vraća ceo prvi red
print(gradovi[1])    #vraća ceo drugi red

print(gradovi[1][0])    #vraća prvi element drugog reda

gradoviFile.close()
```



-čitanje linija redom iz fajla

```
import csv

gradoviFile = open('gradovi.csv')

gradoviReader = csv.reader(gradoviFile)

for row in gradoviReader:
    print(gradoviReader.line_num , row , sep=" - ")
    #u promenljivoj row se smešta jedna po jedna linija fajla u
    obliku liste

gradoviFile.close()
```

-Napomena: Promenljiva *gradoviReader* nije lista već objekat kojim se čita iz fajla



```
import csv
```

```
ljudi = open('ljudi.csv', 'w', newline='')
```

```
ljudiWriter = csv.writer(ljudi, delimiter='\\t', lineterminator='\\n\\n')
```

```
ljudiWriter.writerow(["Ime", "Prezime"])
```

```
ljudiWriter.writerow(["Pera", "Peric"])
```

```
ljudi.close()
```

-Napomena: Ukoliko se prilikom otvaranja fajla ne postavi parametar `newline=""` između svaka dva reda postojaće prazan red

-ukoliko se prilikom pozivanja metode `csv.writer()` ne proslede parametri *delimiter* i *lineterminator* biće po default-u setovani na `,` odnosno `\\n` respektivno

-Najbolje praksa jeste postaviti ove parametre na default-nu vrednost, jer se prilikom čitanja može precizirati samo *delimiter*, ali ne i *lineterminator*



```
import csv
```

```
gradoviFile = open('gradovi.csv')
```

```
gradoviReader = csv.DictReader(gradoviFile, quotechar='"')
```

```
for row in gradoviReader:  
    print(row['City'], row['State'])
```

```
gradovi = list(gradoviReader)
```

```
print(gradovi[0]['City'])
```

-Pomoću metode `csv.DictReader()` svaki pročitani red vraća se u obliku rečnika. Ključevi rečnika jesu elementi prve linije csv fajla



```
import csv
```

```
ljudi = open('ljudi.csv', 'w', newline='')
```

```
polja = ['Ime', 'Prezime']
```

```
ljudiWriter = csv.DictWriter(ljudi, fieldnames=polja)
```

```
ljudiWriter.writeheader()
```

```
ljudiWriter.writerow({'Ime': 'Pera', 'Prezime': 'Peric'})
```

```
ljudiWriter.writerows([{'Ime': 'Laza',  
                        'Prezime': 'Lazic'},  
                        {'Prezime': 'Mikic', 'Ime': 'Mika'}])
```

```
ljudi.close()
```




-JSON(java-script object notation)

- jedan objekat koji sadrži attribute
- svaki atribut ima svoju vrednost koja može biti broj, string, boolean, null, niz ili drugi objekat

```
{ } primer.json x
1  {
2    "name": "John",
3    "age": 30,
4    "cars": [
5      "Ford",
6      "BMW",
7      "Fiat"
8    ],
9    "parent": {
10     "name": "Mark",
11     "age": 60
12   }
13 }
```



```
import json

jsonFajl = open('primer.json')
jsonData = json.loads(jsonFajl.read())

print(jsonData['name'])
print(jsonData['parent']['name'])

jsonFajl.close()
```



```
import json
```

```
x = {  
    "name": "Pera",  
    "age": 22,  
    "cars": ["Jugo", "Kec"],  
    "parent": {  
        "name": "Mika",  
        "age": 50  
    }  
}  
jsonObjekat = json.dumps(x, indent=4)  
  
fajl = open('primer.json', 'w')  
  
fajl.write(jsonObjekat)  
fajl.close()
```

`json.dumps()` metoda pravi json objekat od prosleđenog python objekta, ukoliko se ne setuje paramtera `indent` ceo json objekat će biti u jendom redu



Pronaći sve gradove čija su imena dužine najmanje 11 a koji imaju dva uzastopna mala slova L negde u svom imenu



```
import csv,re
```

```
gradoviFile = open('gradovi.csv')
```

```
gradoviReader = csv.DictReader(gradoviFile, quotechar='"')
```

```
gradovi = list(gradoviReader)
```

```
gradRegexLL = re.compile(r'.*ll.*')
```

```
gradRegexLen = re.compile(r'.{11,}')
```

```
for grad in gradovi:
```

```
    rezLL = gradRegexLL.search(grad['City'])
```

```
    rezLen = gradRegexLen.search(grad['City'])
```

```
    if (rezLen and rezLL):
```

```
        print(grad['City'])
```

Dat je csv fajl sa podacima o visini loptice koja odskače od zemlje u formatu Vreme,Visina. Odrediti sve maximume visine koje je loptica dostigla. Rešenje zadatka ispisati u resenje.json koji je oblika:

```
{  
    maximumi:[{"vreme":vreme1, "visina":visina1},...]  
}
```

```
import csv,json

def srednjiMax(prethodni,trenutni,sledeci):
    if (float(trenutni['Visina'])>float(prethodni['Visina'])) and
float(trenutni['Visina'])>float(sledeci['Visina'])):
        return True
    return False

podaci = open('loptica.csv')
parovi = list(csv.DictReader(podaci))
resenje = []
```



```
for par in parovi[1:-1]:
    index = parovi.index(par)
    if (srednjiMax(parovi[index-1],par,parovi[index+1])):
        resenje.append({'vreme':par['Vreme'],
        'visina':par['Visina']})
podaci.close()
resenjeFile = open('resenje.json','w')

resenjeFile.write(json.dumps(resenje,indent=4))
```