+01110

Matija Pretnar
Faculty of Mathematics and Physics

Projekt **Tomo**

Projekt **Tomo**

a web service
for teaching programming

Introduction to programming

2009

Introduction to programming

- a solution either **works** or **not**

2009

```python
#######################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
#######################################################################@@#


##################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##################################################################000536@#
def linearna(a, b):
    return -b / a

##################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
# ploščino pravilnega $n$-kotnika s stranico $a$.
##################################################################000537@#



##################################################################@000538#
# 3) Sestavite funkcijo prestopno(leto), ki vrne True, če je leto leto
# prestopno, sicer pa vrne False.
##################################################################000538@#
```

```
#######################################################################@@#
# Kode pod to črto nikakor ne spreminjajte.
##########################################################################

"TA VRSTICA JE PRAVILNA."
"ČE VAM PYTHON SPOROČI, DA JE V NJEJ NAPAKA, SE MOTI."
"NAPAKA JE NAJVERJETNEJE V ZADNJI VRSTICI VAŠE KODE."
"ČE JE NE NAJDETE, VPRAŠAJTE ASISTENTA."
```

```
            Check.equal("""ploscina(5,2)""", 6.881909602355868)

            Check.challenge([round(ploscina(n,a),2) for n in range(3,8) for a in range(1,4)])
            pass
        except:
            Check.error("Testi sprožijo izjemo\n  {0}", "\n  ".join(traceback.format_exc().split("\n"))[:-2])


    if Check.part():
        try:
            Check.equal("prestopno(2011)", False)
            Check.equal("prestopno(2000)", True)
            Check.equal("prestopno(1900)", False)
            Check.equal("prestopno(2004)", True)

            Check.challenge([prestopno(leto) for leto in range(1500, 2000)])
            pass
        except:
            Check.error("Testi sprožijo izjemo\n  {0}", "\n  ".join(traceback.format_exc().split("\n"))[:-2])



    print('Shranjujem rešitve na strežnik... ', end = "")
    post = json.dumps({
        'data': '{"timestamp": "2012-03-05 09:19:00.553187", "problem": 176, "user": 163}',
        'signature': '3ef56633a56dababae402589458f48b3',
        'preamble': _preamble,
        'attempts': Check.parts,
        'source': _source,
    }).encode('utf-8')
    try:
        r = urlopen('http://tomo.fmf.uni-lj.si:80/problem/upload/student/', post)
        response = json.loads(r.read().decode('utf-8'))
        print('Rešitve so shranjene.')
        for (k, e) in response['rejected']:
            Check.parts[k - 1]['rejection'] = e
        Check.summarize()
        if 'update' in response:
            print("Posodabljam datoteko... ", end = "")
            index = 1
            while os.path.exists('{0}.{1}'.format(_filename, index)):
                index += 1
            backup_filename = "{0}.{1}".format(_filename, index)
            shutil.copy(_filename, backup_filename)
            r = urlopen(response['update'])
            with open(_filename, 'w', encoding='utf-8') as f:
                f.write(r.read().decode('utf-8'))
            print("Stara datoteka je preimenovana v {0}.".format(os.path.basename(backup_filename)))
            print("Če se datoteka v urejevalniku ni osvežila, jo zaprite ter ponovno odprite.")
    except HTTPError as r:
        print('Pri shranjevanju je prišlo do napake.')
        Check.summarize()
        print('Pri shranjevanju je prišlo do napake. Poskusite znova.')
        Check.error = r.read().decode('utf-8')


_check()

#####################################################################@@#
```

Introduction to programming

2009

Introduction to programming

- a solution either **works** or **not**

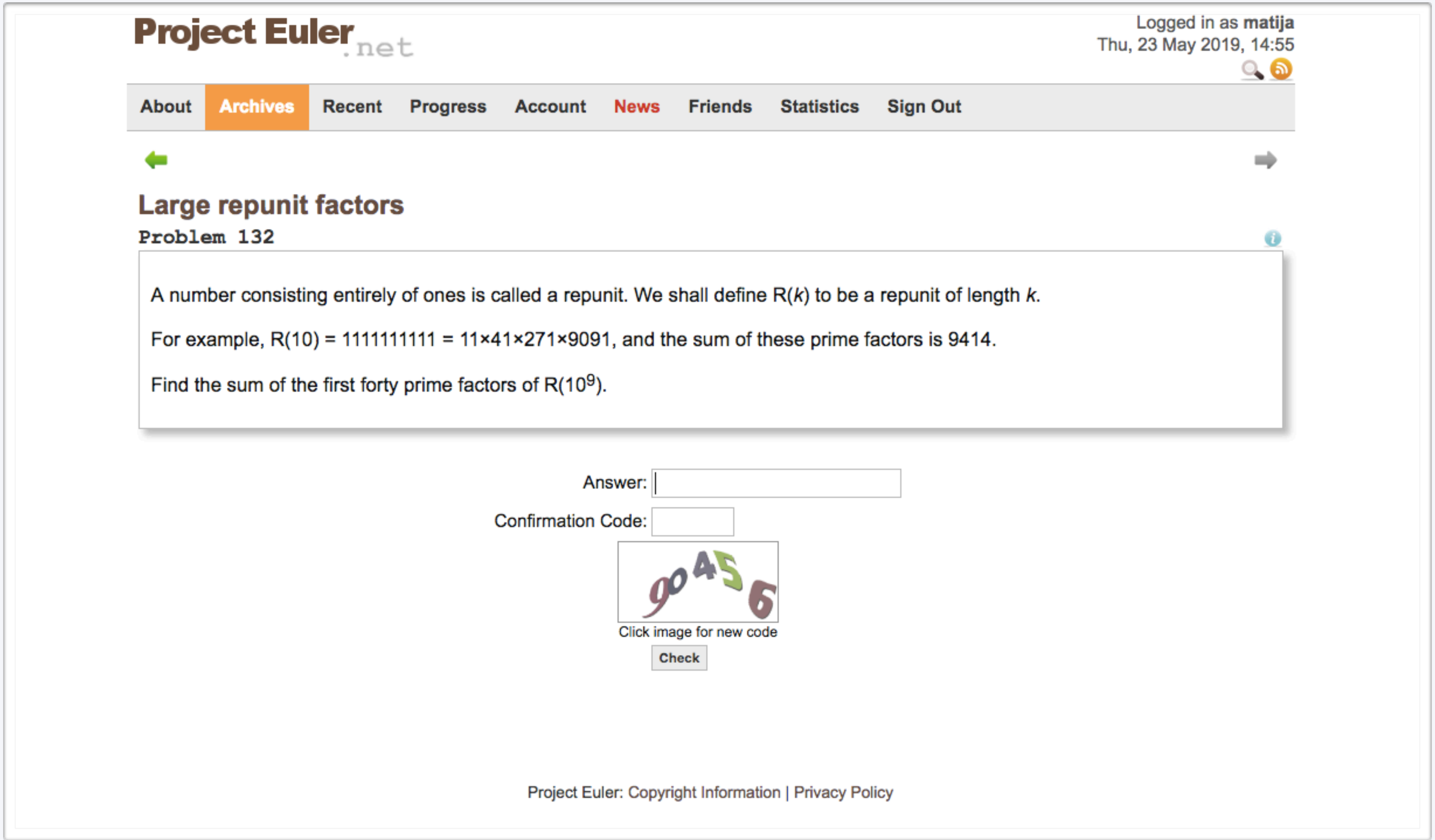
2009

Introduction to programming

- a solution either **works** or **not**
- **extra marks** for Project Euler submissions

2009

Project Euler



Logged in as **matija**
Thu, 23 May 2019, 14:55

Large repunit factors

Problem 132

A number consisting entirely of ones is called a repunit. We shall define R(*k*) to be a repunit of length *k*.

For example, R(10) = 1111111111 = 11×41×271×9091, and the sum of these prime factors is 9414.

Find the sum of the first forty prime factors of R(10^9).

Answer:

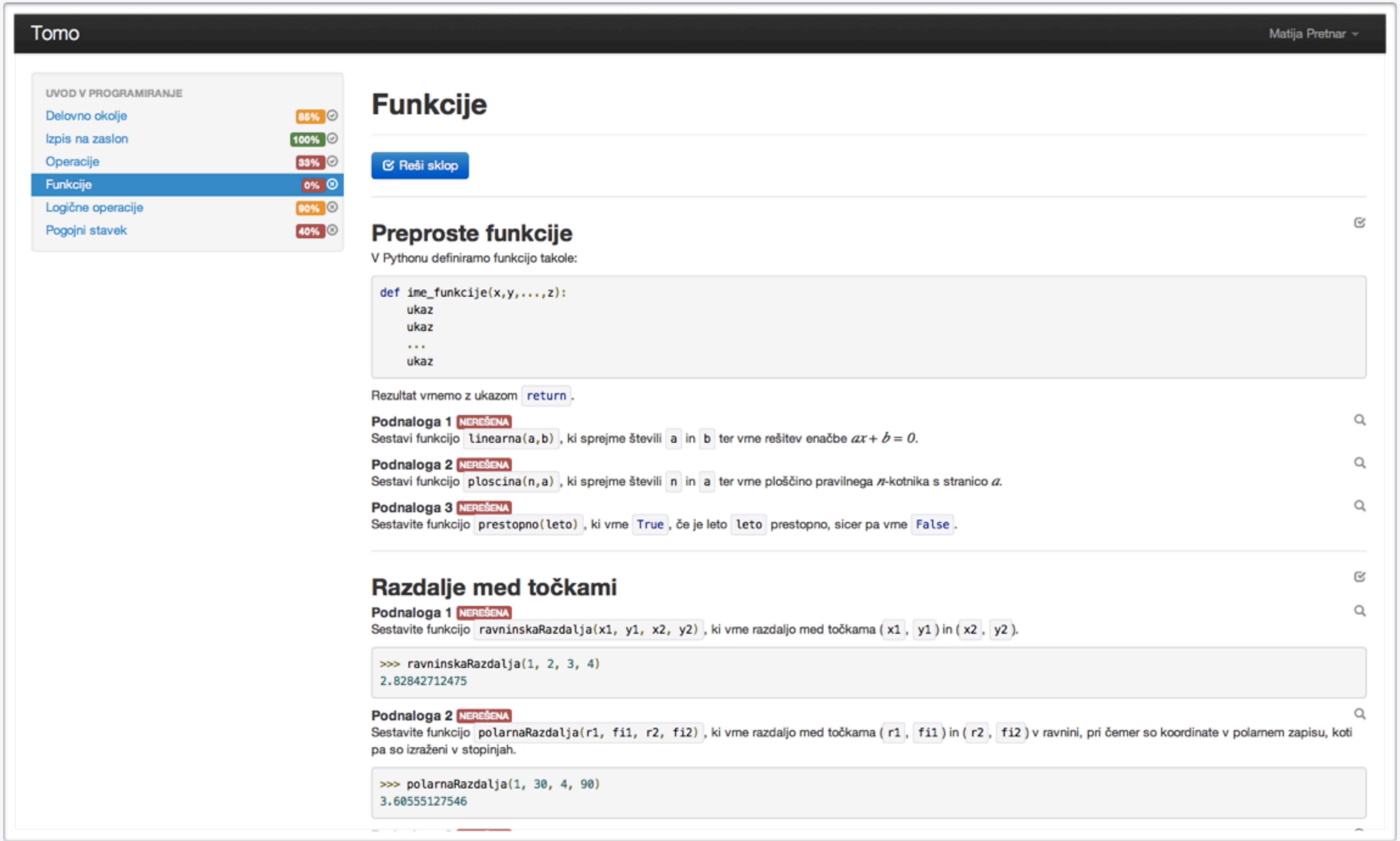
Confirmation Code:

Project Euler: Copyright Information | Privacy Policy

Projekt Tomo 1.0

2011

Projekt Tomo 2.0



Funkcije

Preproste funkcije

V Pythonu definiramo funkcijo takole:

```
def ime_funkcije(x,y,...,z):
    ukaz
    ukaz
    ...
    ukaz
```

Rezultat vrnemo z ukazom `return`.

Podnaloga 1
Sestavi funkcijo `linearna(a,b)`, ki sprejme števili `a` in `b` ter vrne rešitev enačbe $ax + b = 0$.

Podnaloga 2
Sestavi funkcijo `ploscina(n,a)`, ki sprejme števili `n` in `a` ter vrne ploščino pravilnega n-kotnika s stranico a.

Podnaloga 3
Sestavite funkcijo `prestopno(leto)`, ki vrne `True`, če je leto `leto` prestopno, sicer pa vrne `False`.

Razdalje med točkami

Podnaloga 1
Sestavite funkcijo `ravninskaRazdalja(x1, y1, x2, y2)`, ki vrne razdaljo med točkama (`x1`, `y1`) in (`x2`, `y2`).

```
>>> ravninskaRazdalja(1, 2, 3, 4)
2.82842712475
```

Podnaloga 2
Sestavite funkcijo `polarnaRazdalja(r1, fi1, r2, fi2)`, ki vrne razdaljo med točkama (`r1`, `fi1`) in (`r2`, `fi2`) v ravnini, pri čemer so koordinate v polarnem zapisu, koti pa so izraženi v stopinjah.

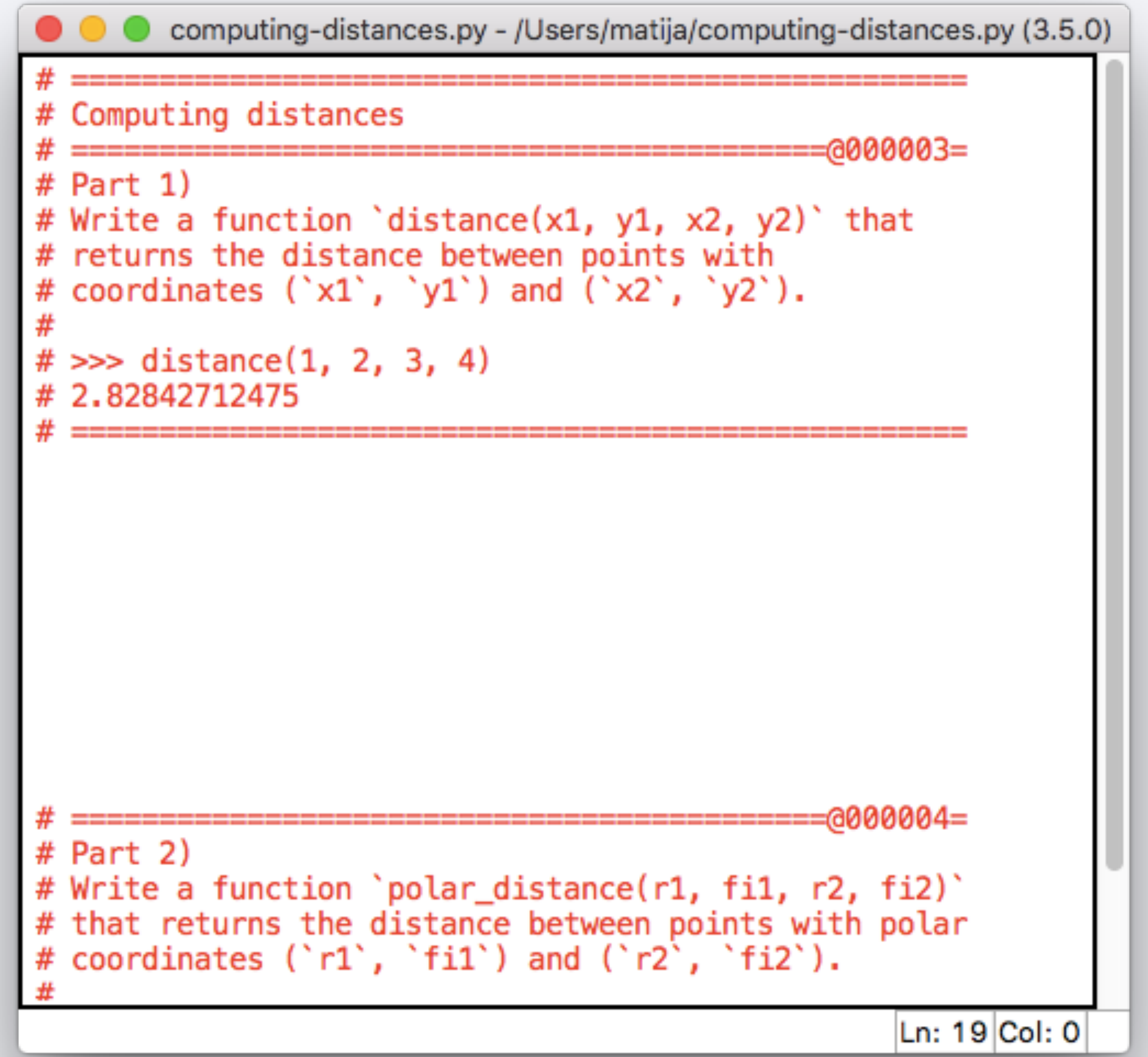
```
>>> polarnaRazdalja(1, 30, 4, 90)
3.60555127546
```

with A. Bauer

2011

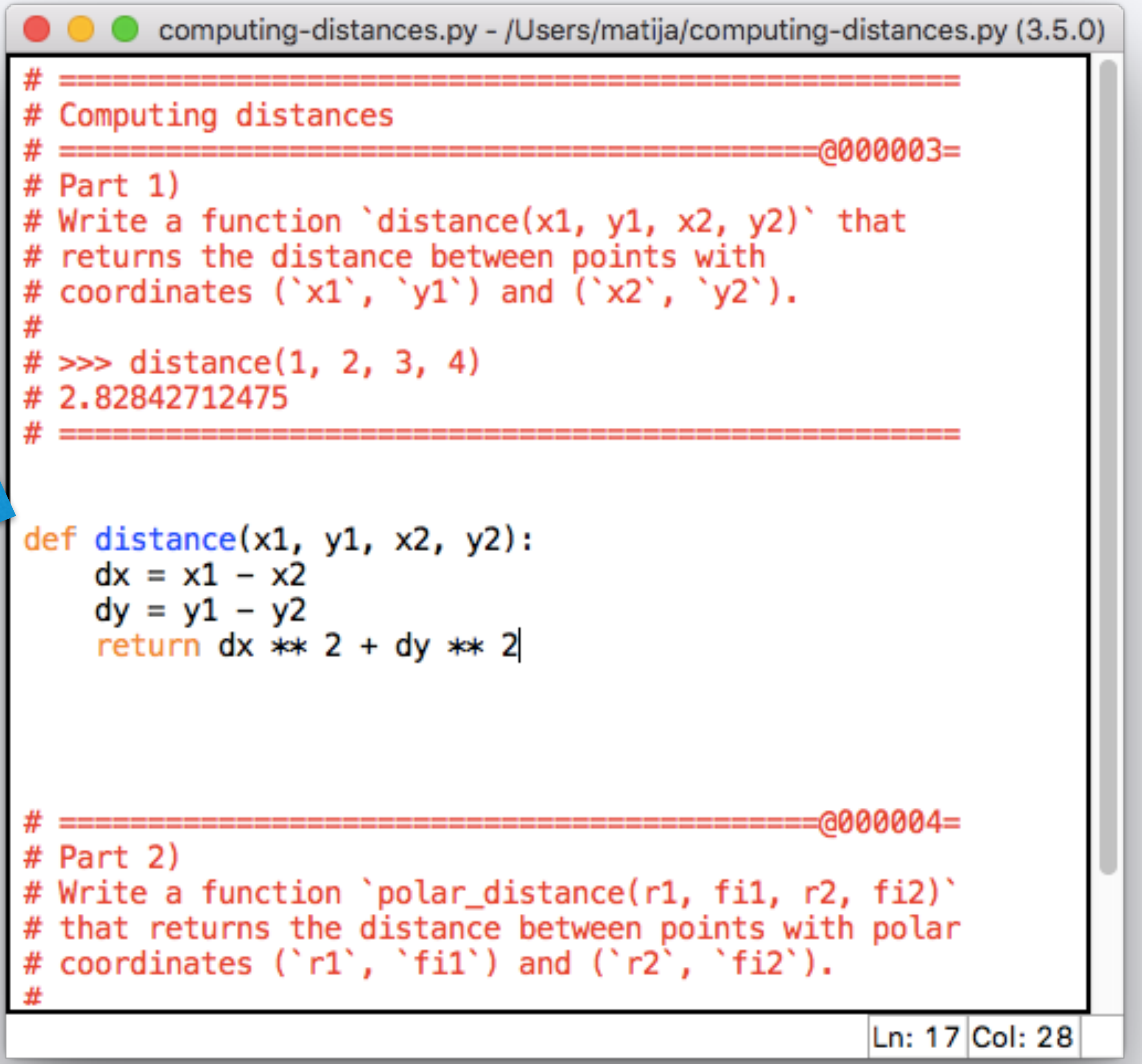
Open downloaded file

in your favourite editor


computing-distances.py - /Users/matija/computing-distances.py (3.5.0)
Ln: 19
Col: 0

```
# ===================================================
# Computing distances
# ===========================================@000003=
# Part 1)
# Write a function `distance(x1, y1, x2, y2)` that
# returns the distance between points with
# coordinates (`x1`, `y1`) and (`x2`, `y2`).
#
# >>> distance(1, 2, 3, 4)
# 2.82842712475
# ===================================================








# ===========================================@000004=
# Part 2)
# Write a function `polar_distance(r1, fi1, r2, fi2)`
# that returns the distance between points with polar
# coordinates (`r1`, `fi1`) and (`r2`, `fi2`).
#
```

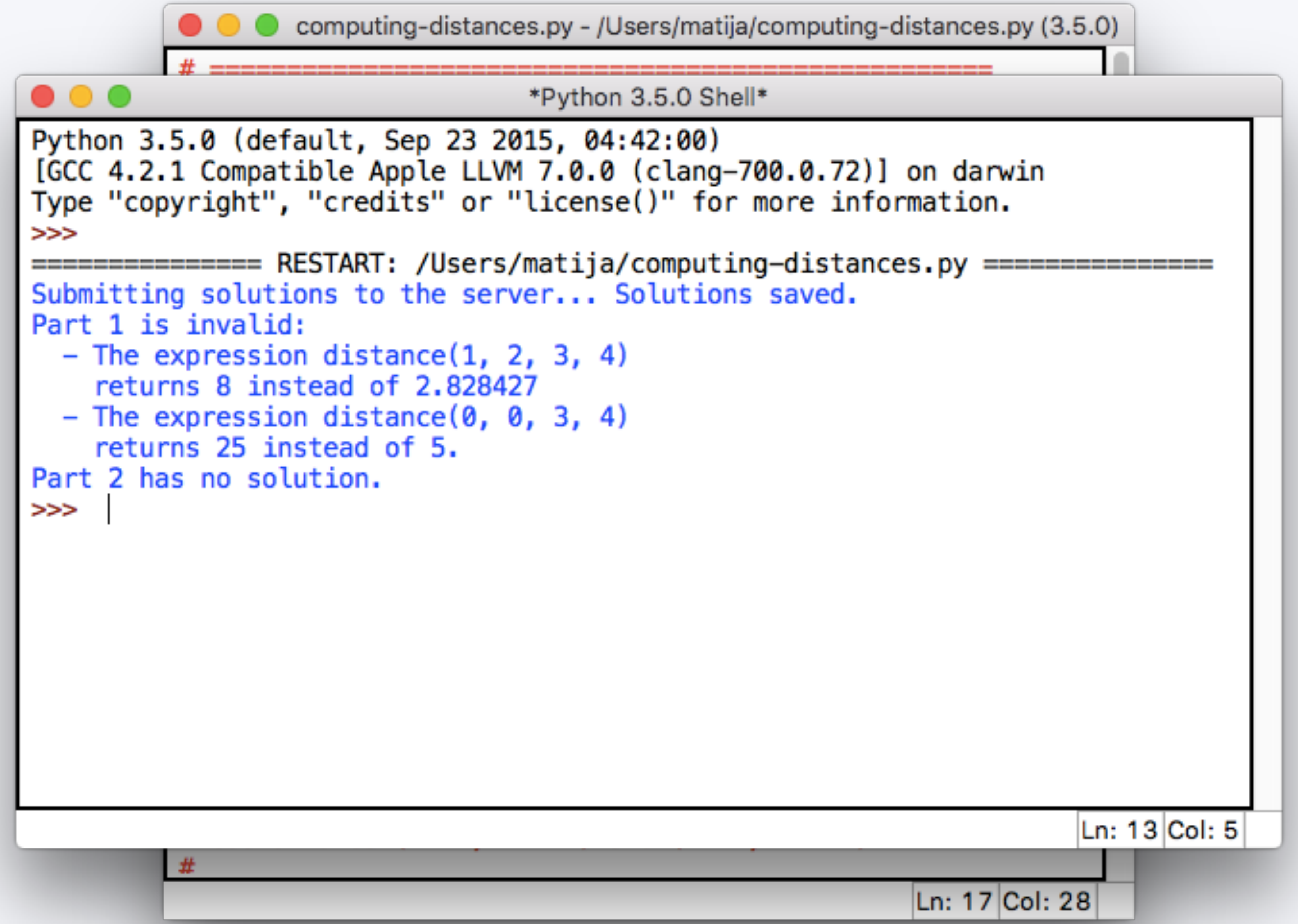
Fill in your solutions



computing-distances.py - /Users/matija/computing-distances.py (3.5.0)

```
# ===================================================
# Computing distances
# ===========================================@000003=
# Part 1)
# Write a function `distance(x1, y1, x2, y2)` that
# returns the distance between points with
# coordinates (`x1`, `y1`) and (`x2`, `y2`).
#
# >>> distance(1, 2, 3, 4)
# 2.82842712475
# ===================================================


def distance(x1, y1, x2, y2):
    dx = x1 - x2
    dy = y1 - y2
    return dx ** 2 + dy ** 2




# ===========================================@000004=
# Part 2)
# Write a function `polar_distance(r1, fi1, r2, fi2)`
# that returns the distance between points with polar
# coordinates (`r1`, `fi1`) and (`r2`, `fi2`).
#
```

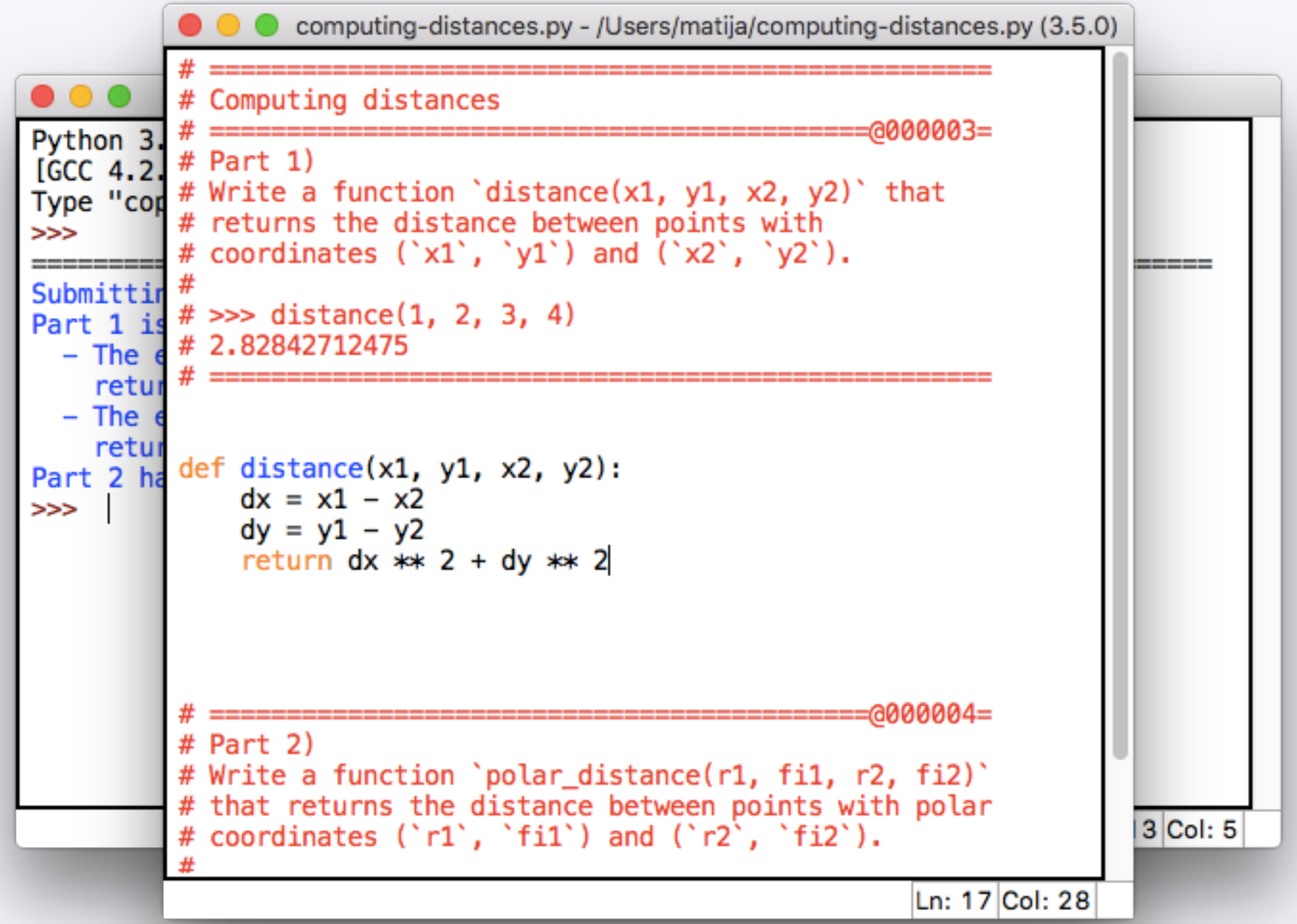
Run the file to save and check



computing-distances.py - /Users/matija/computing-distances.py (3.5.0)

```
Python 3.5.0 (default, Sep 23 2015, 04:42:00)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.72)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: /Users/matija/computing-distances.py ===============
Submitting solutions to the server... Solutions saved.
Part 1 is invalid:
  - The expression distance(1, 2, 3, 4)
    returns 8 instead of 2.828427
  - The expression distance(0, 0, 3, 4)
    returns 25 instead of 5.
Part 2 has no solution.
>>> 
```

Repeat until you solve it



computing-distances.py - /Users/matija/computing-distances.py (3.5.0)

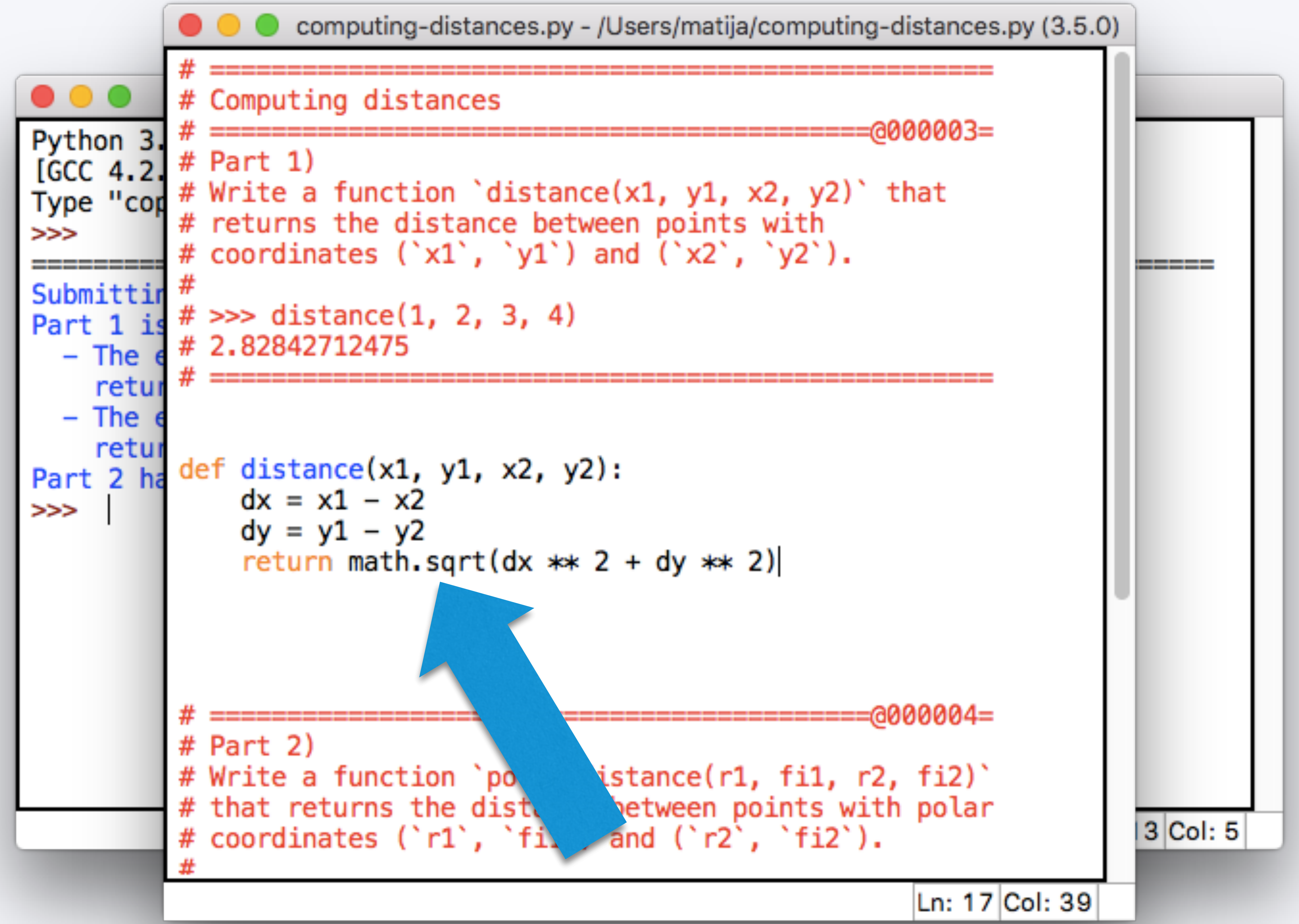
```
# =====================================================================
# Computing distances
# ==========================================================@000003=
# Part 1)
# Write a function `distance(x1, y1, x2, y2)` that
# returns the distance between points with
# coordinates (`x1`, `y1`) and (`x2`, `y2`).
#
# >>> distance(1, 2, 3, 4)
# 2.82842712475
# =====================================================================


def distance(x1, y1, x2, y2):
    dx = x1 - x2
    dy = y1 - y2
    return dx ** 2 + dy ** 2




# ==========================================================@000004=
# Part 2)
# Write a function `polar_distance(r1, fi1, r2, fi2)`
# that returns the distance between points with polar
# coordinates (`r1`, `fi1`) and (`r2`, `fi2`).
#
```

Ln: 17 Col: 28

```
Python 3.
[GCC 4.2.
Type "cop
>>>
========
Submittin
Part 1 is
  - The e
    retur
  - The e
    retur
Part 2 ha
>>> 
```

3 Col: 5

Repeat until you solve it



computing-distances.py - /Users/matija/computing-distances.py (3.5.0)

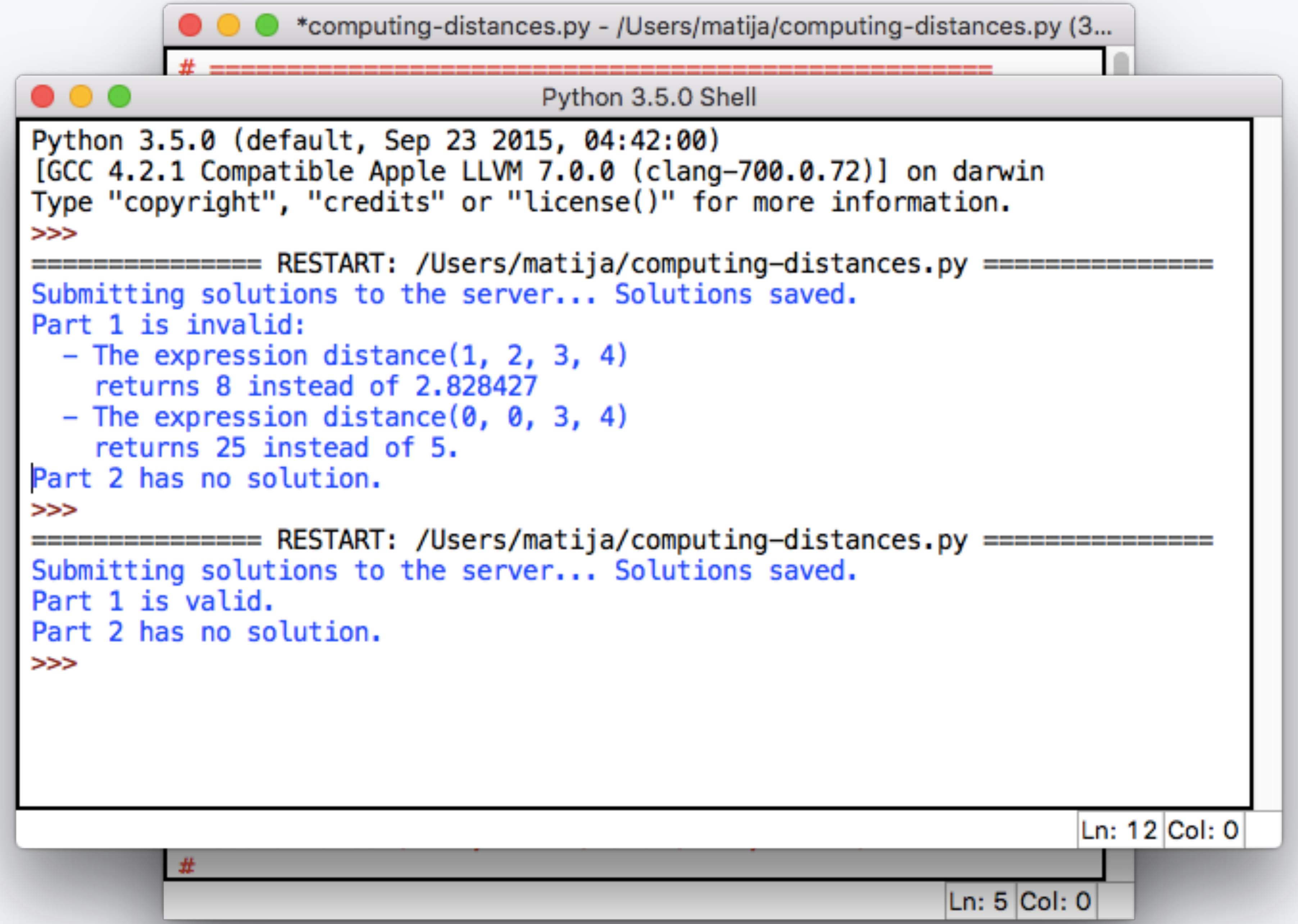
```
# ===================================================
# Computing distances
# ===========================================@000003=
# Part 1)
# Write a function `distance(x1, y1, x2, y2)` that
# returns the distance between points with
# coordinates (`x1`, `y1`) and (`x2`, `y2`).
#
# >>> distance(1, 2, 3, 4)
# 2.82842712475
# ===================================================


def distance(x1, y1, x2, y2):
    dx = x1 - x2
    dy = y1 - y2
    return math.sqrt(dx ** 2 + dy ** 2)



# ===========================================@000004=
# Part 2)
# Write a function `po[illegible]istance(r1, fi1, r2, fi2)`
# that returns the dist[illegible]etween points with polar
# coordinates (`r1`, `fi[illegible]and (`r2`, `fi2`).
#
```

Ln: 17 Col: 39

```
Python 3.
[GCC 4.2.
Type "cop
>>>
========
Submittin
Part 1 is
  - The e
    retur
  - The e
    retur
Part 2 ha
>>> |
```

3 Col: 5

Repeat until you solve it



*computing-distances.py - /Users/matija/computing-distances.py (3...
Python 3.5.0 Shell
Ln: 12 Col: 0
Ln: 5 Col: 0

```
Python 3.5.0 (default, Sep 23 2015, 04:42:00)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.72)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: /Users/matija/computing-distances.py ===============
Submitting solutions to the server... Solutions saved.
Part 1 is invalid:
  - The expression distance(1, 2, 3, 4)
    returns 8 instead of 2.828427
  - The expression distance(0, 0, 3, 4)
    returns 25 instead of 5.
Part 2 has no solution.
>>>
=============== RESTART: /Users/matija/computing-distances.py ===============
Submitting solutions to the server... Solutions saved.
Part 1 is valid.
Part 2 has no solution.
>>>
```

Projekt Tomo 3.0

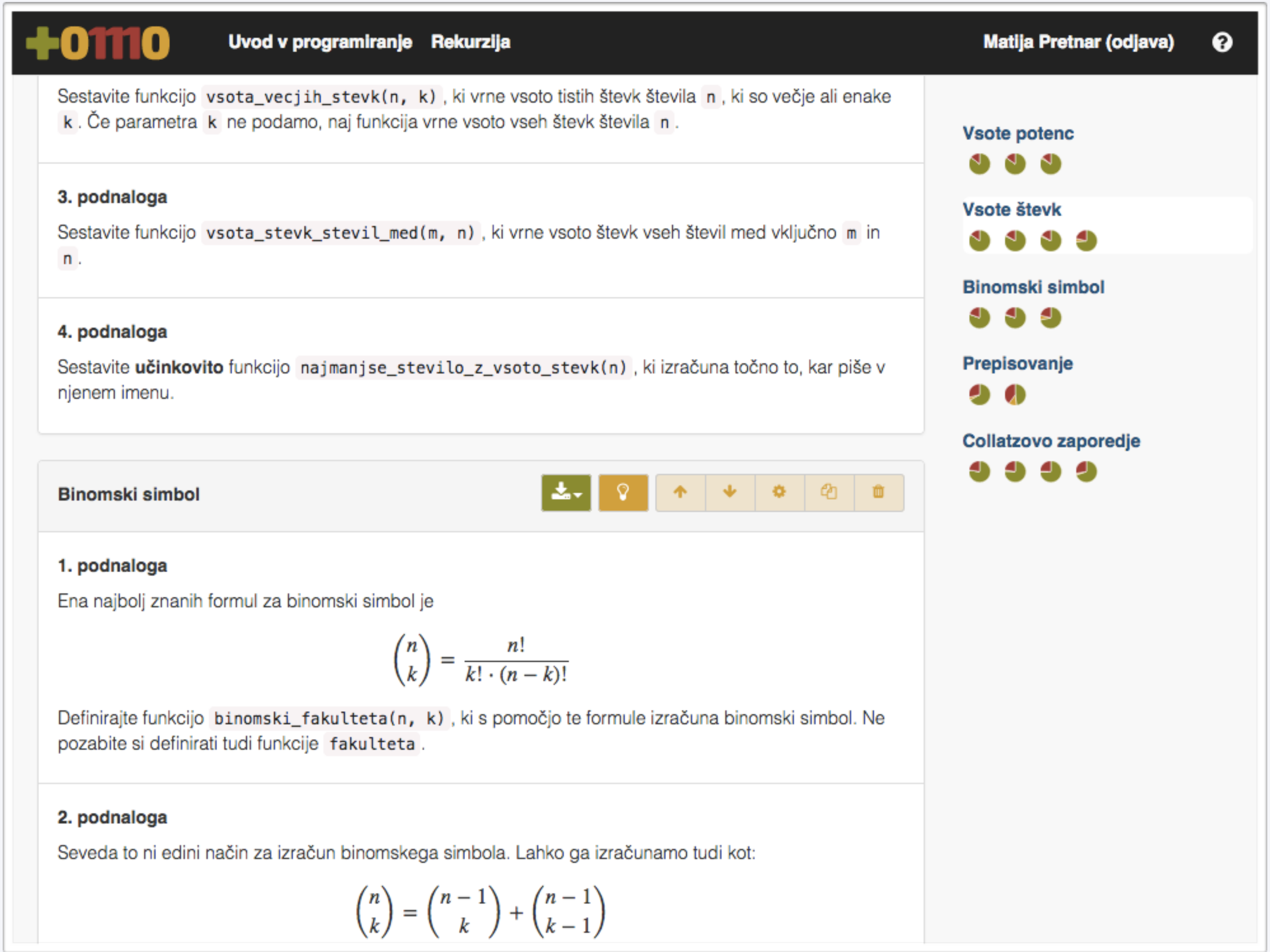
Univerza *v Ljubljani*

Naložba v vašo prihodnost
OPERACIJO DELNO FINANCIRA EVROPSKA UNIJA
Evropski sklad za regionalni razvoj

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST IN ŠPORT

with K. Berčič, G. Jerše, S. Jerše, M. Lokar

2015



Sestavite funkcijo `vsota_vecjih_stevk(n, k)`, ki vrne vsoto tistih števk števila `n`, ki so večje ali enake `k`. Če parametra `k` ne podamo, naj funkcija vrne vsoto vseh števk števila `n`.

3. podnaloga

Sestavite funkcijo `vsota_stevk_stevil_med(m, n)`, ki vrne vsoto števk vseh števil med vključno `m` in `n`.

4. podnaloga

Sestavite **učinkovito** funkcijo `najmanjse_stevilo_z_vsoto_stevk(n)`, ki izračuna točno to, kar piše v njenem imenu.

Binomski simbol

1. podnaloga

Ena najbolj znanih formul za binomski simbol je

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Definirajte funkcijo `binomski_fakulteta(n, k)`, ki s pomočjo te formule izračuna binomski simbol. Ne pozabite si definirati tudi funkcije `fakulteta`.

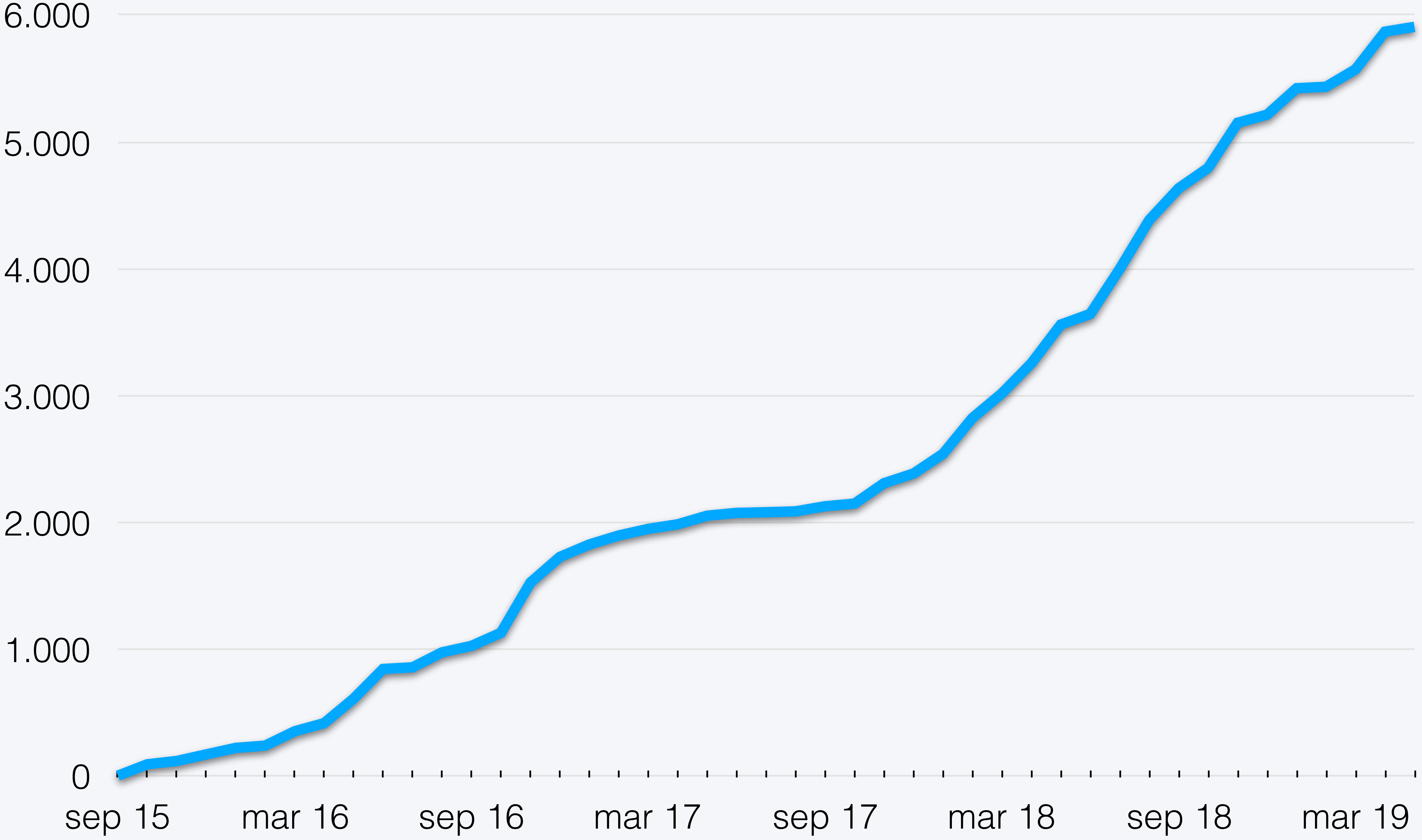
2. podnaloga

Seveda to ni edini način za izračun binomskega simbola. Lahko ga izračunamo tudi kot:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Maps ∩ Configurations ∩
Polytopes ∩ Molecules
⊆
Graphs

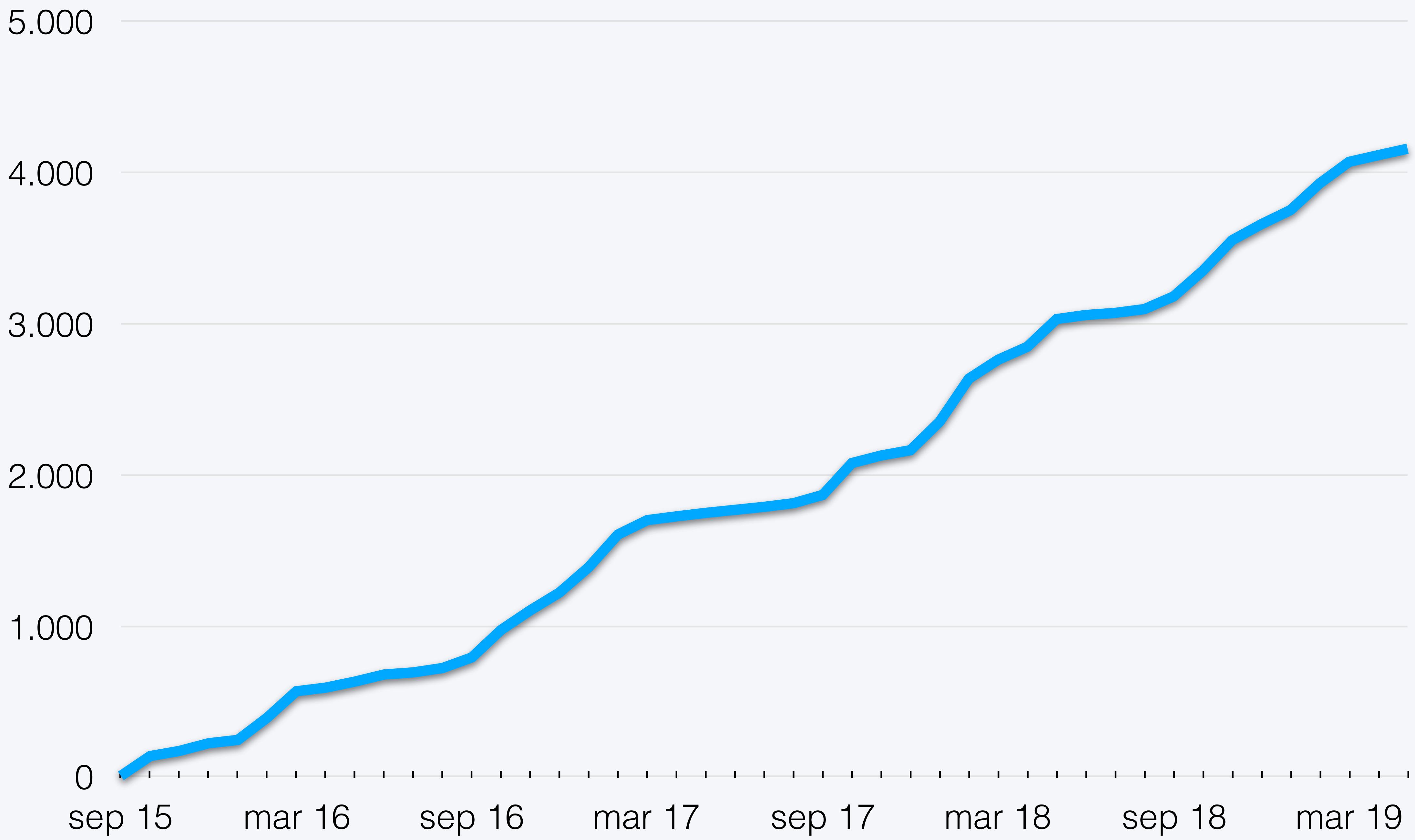
Problems


6.000
5.000
4.000
3.000
2.000
1.000
0
sep 15
mar 16
sep 16
mar 17
sep 17
mar 18
sep 18
mar 19

Languages

- Python
- R
- Octave/Matlab

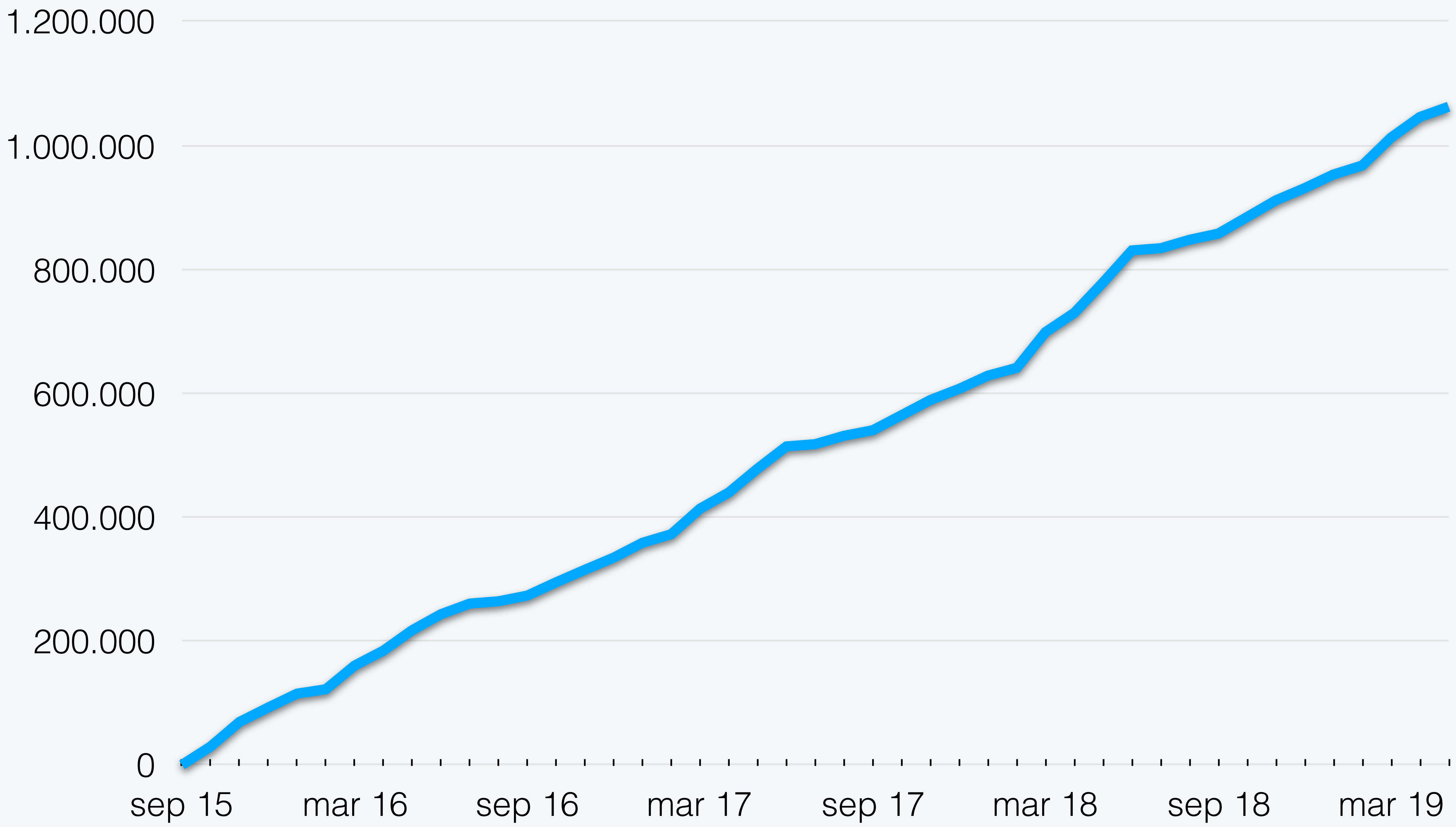
Users


5.000
4.000
3.000
2.000
1.000
0
sep 15
mar 16
sep 16
mar 17
sep 17
mar 18
sep 18
mar 19

Institutions

- 12 elementary schools
- 18 high schools
- 6 university departments
- 2 companies
- 2 research projects

Attempts


1.200.000
1.000.000
800.000
600.000
400.000
200.000
0
sep 15
mar 16
sep 16
mar 17
sep 17
mar 18
sep 18
mar 19

www.projekt-tomo.si

111