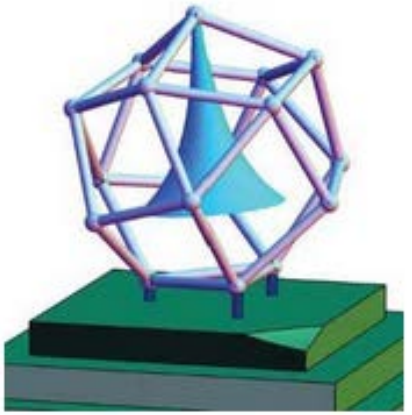
CADGME 2016



Learning and teaching programming and numerical methods with a system for automatic assessment

Gregor Jerše
Faculty of Computer and Information Science
University of Ljubljana

Matija Lokar
Faculty of Mathematics and Physics
University of Ljubljana

Situation

- Numerical methods class
- Primary form of work are lab exercises
- Students have to solve a significant amount of examples
- Students have already completed programming course
- Use their programming skills: solve lots of programming exercises connected with numerical methods

Situation

- Teachers have to prepare lots of numerical methods oriented exercises
- Students' attempts to solve them have to be supervised
- Swift teacher feedback is desired
- Teacher should have insight into students' solutions
- Problems:
  - teacher feedback can be slow, not always available
  - Requires lots of additional work from the teacher

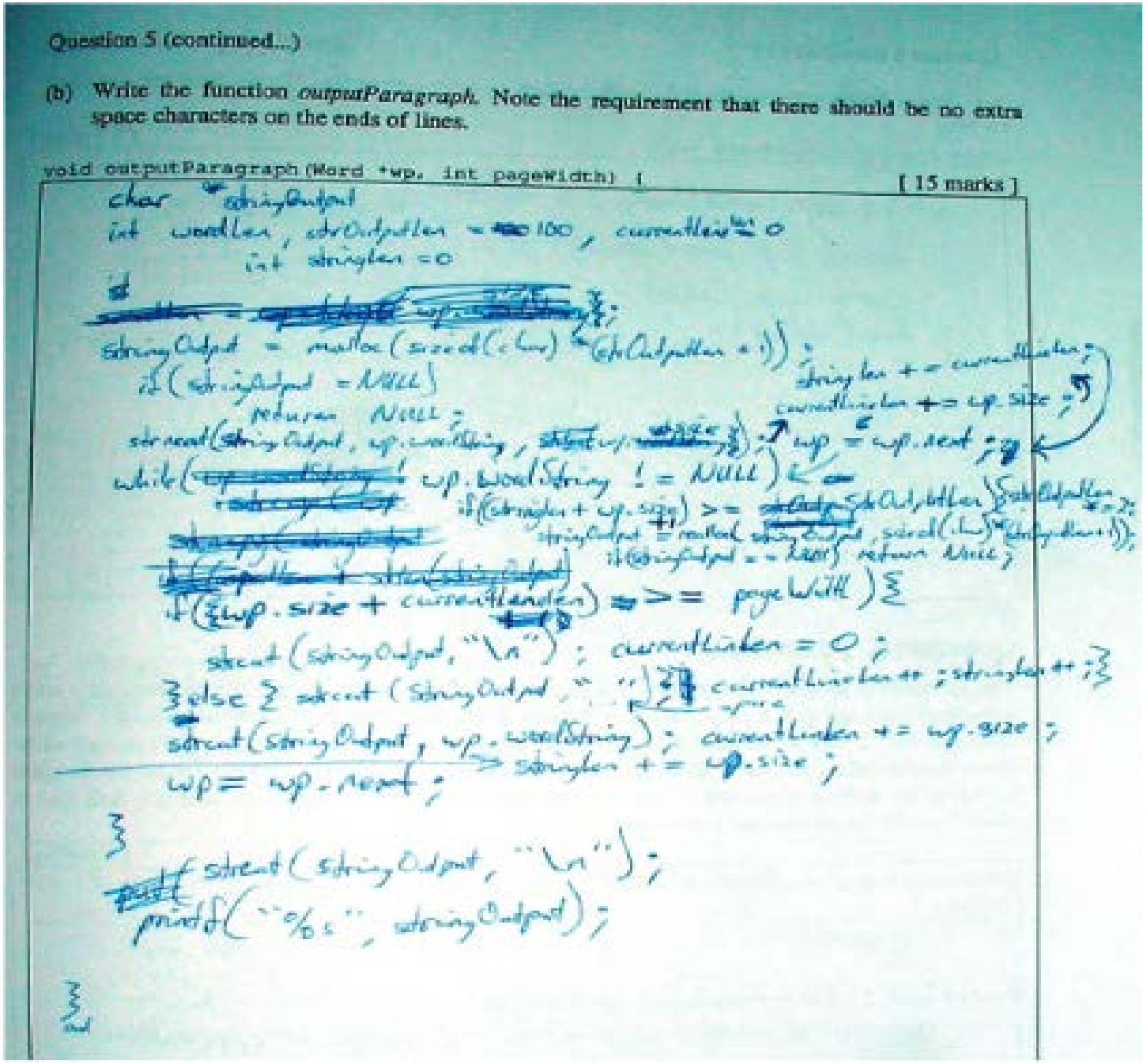


Image source: http://coderunner.org.nz/pluginfile.php/1746/mod_resource/content/2/CodeRunnerArticlePublishedForm.pdf

Solution

- Solutions have to be in a form of a computer program
  - Programs are readable by the computer
  - Can be automatically assessed

- We developed a service for automatic assessment (SAAPE).

- SAAPE: Systems for automatic assessment of programming exercises

Requirements

- Easy to use
- Requires little or no additional work from teacher and student
- Provides good and swift feedback to the students
- Stores history of their solutions
- Introduces no new programming environment
- Web based

Why yet another SAAPE?

- Existing solutions did not suit our requirements
- Usually SAAPE requires use of external server
  - Server has to be powerful, expensive
  - Requires students to upload their solutions, time consuming
  - Security issues with executing unknown code
  - Prone to DOS attacks…

Why YASAAPE?

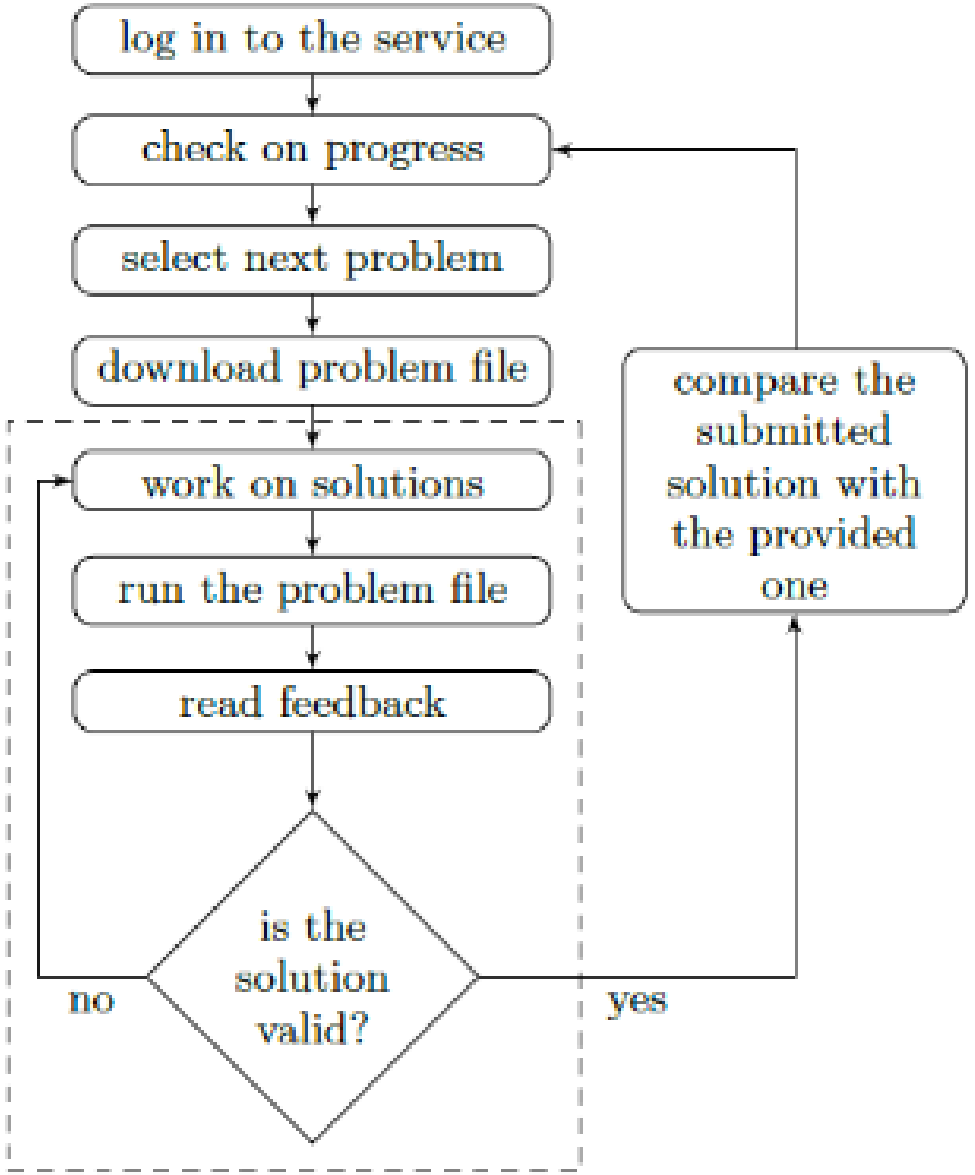
- Some run locally on the computer
  - Usually they run inside web browser
  - Instant feedback is given
  - They eliminate security issues
  - Not available when offline
  - Tied to Javascript or languages simulated by Javascript
  - Usually no insight into students' solutions

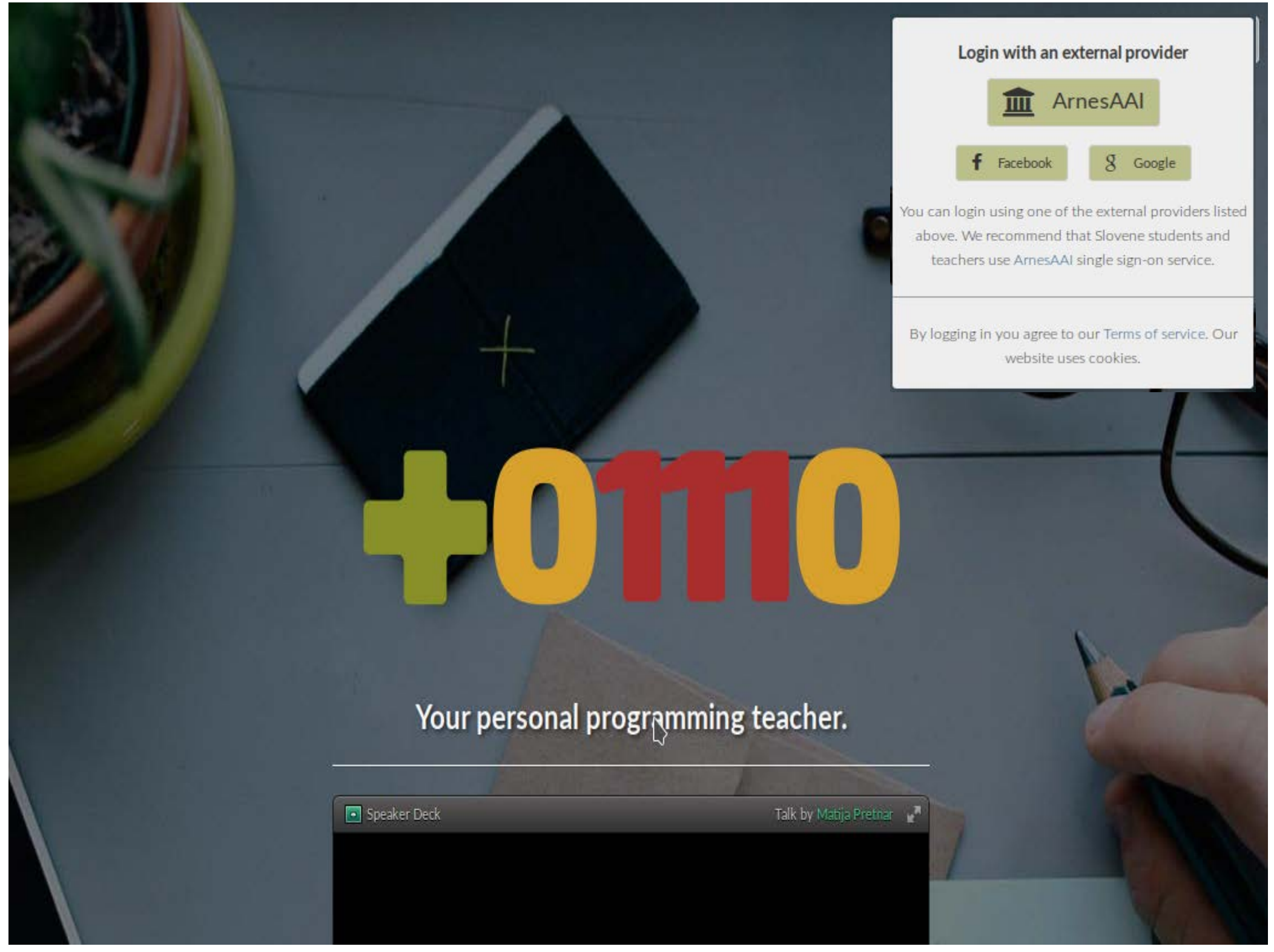
SAAPE Tomo

- Local execution (low overhead interaction)
- Instant feedback
- IDE agnostic
- Language agnostic: currently supports Python3 and Octave
- Flexibility in writing tests
- Entire students solution history is stored on the server
- Teacher has insight into students' solutions
- Works offline

http://www.projekt-tomo.si

Tomo workflow - student




Login with an external provider
ArnesAAI
Facebook
Google
You can login using one of the external providers listed above. We recommend that Slovene students and teachers use ArnesAAI single sign-on service.
By logging in you agree to our Terms of service. Our website uses cookies.
+0110
Your personal programming teacher.
Speaker Deck
Talk by Matija Pretnar

Tomo: students perspecitve

- Multiple courses are available
- Each course is divided into problem sets
- Each problem set contains several exercises

CADGME 2016

0% Simple problems

20% Workshop

0% Excercise 14

Drugi koraki v Python

Matija Lokar, UROŠ VAUPOTIČ

@{Fakulteta za matematiko in fiziko}

CADGME 2016

0% Excercise 14

20% Workshop

0% Simple problems

Gregor Jerse

Matija Lokar

Matija Lokar

Fakulteta za matematiko in fiziko

CADGME 2016 conference examples

Workshop

Problem set for ISSEP Workshop

Download this problem file

Body mass index

1. part

BMI or body mass index is a number that helps us to roughly estimate the nutritional status of human adults. If `m` is a person's weight in kilograms and `h` his height in meters, we calculate the BMI as follows: $ITM = \frac{m}{h^2}$.

A person's weight, measured in kilograms, is written in the variable `mass`. A person's height, measured in centimeters, is written in the variable `height`. Write a program that computes BMI and tells you what is your nutritional status. The computed BMI should be written to an accuracy of two decimal places, using function `two_decimal_places`:

```
def two_decimal_places(x):
    """ Function returns a string which contains a record of the decimal
     number, written to an accuracy of two decimal places. """
    return '{0:.2f}'.format(x)
```

Output should be as follows:

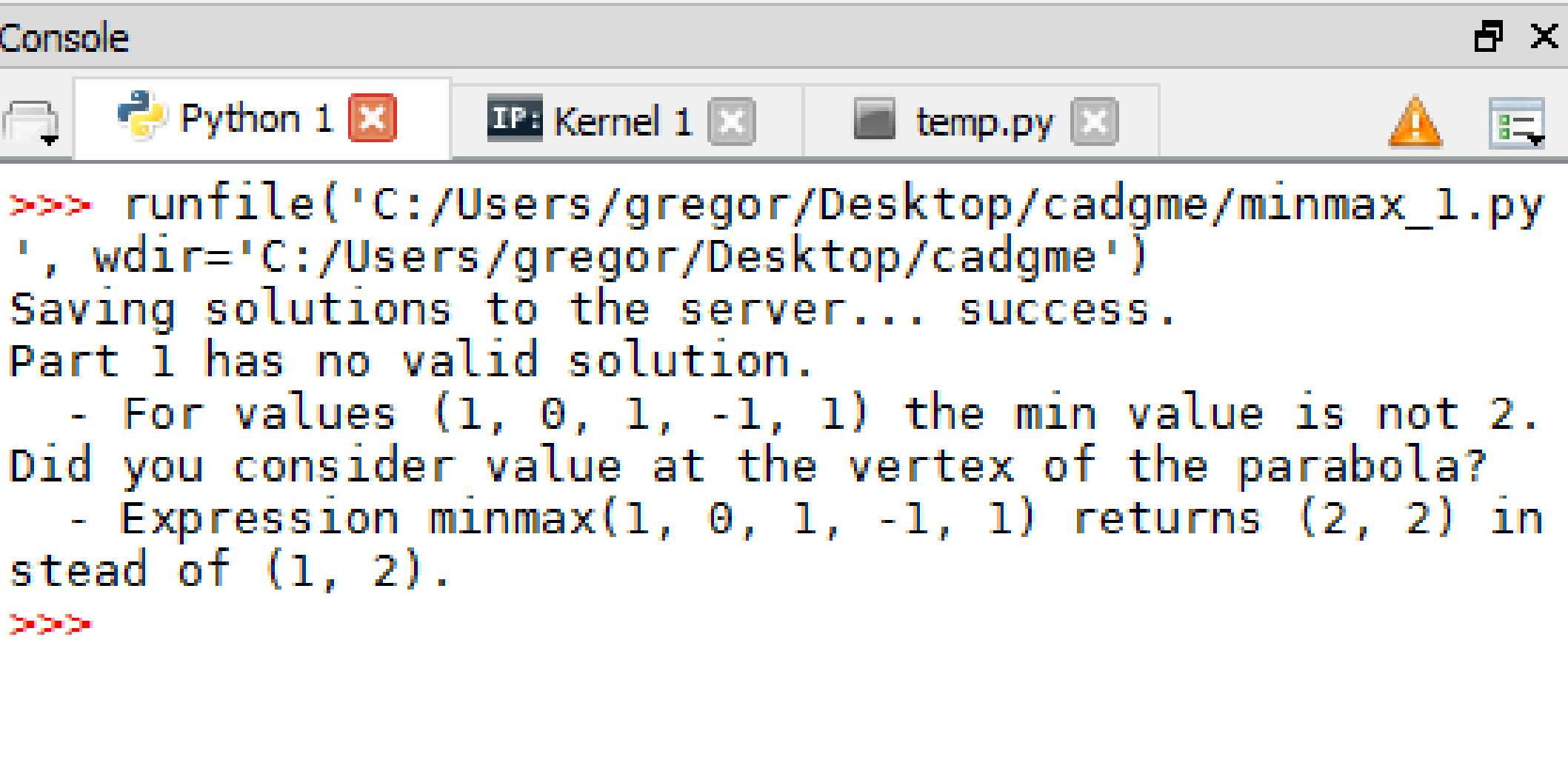
- ITM<18.5: `You are underweight.`
- 18.5≤ITM<25: `You have normal weight.`
- 25≤ITM<30: `You are overweight.`
- 30≤ITM: `You are obese.`

```
# ============================================================================
# Minmax 1
#
# Find the max and min values of a function $ax^2 + bx + c$ an a closed interval.
# ============================================================================@002828=
# 1. part
# Write a function minmax(a, b, c, d, e) that returns min value $m$ and max
# value $M$ of the function $ax^2 + bx + c$ an a closed interval $[d, e]$.
# Values must be returned as a tuple $(m, M)$.
#
# Example:
#
#     >>> minmax(1, 0, 1, -1, 1)
#     (1, 2)
#     >>> minmax(1, 0, 1, 1, 2)
#     (2, 5)
# ============================================================================

import numpy

def minmax(a, b, c, d, e):
    def f(x):
        return a*numpy.square(x) + b*x + c
    values = [f(d), f(e)]
    return (min(values), max(values))

```


Console
Python 1
IP: Kernel 1
temp.py
>>> runfile('C:/Users/gregor/Desktop/cadgme/minmax_1.py', wdir='C:/Users/gregor/Desktop/cadgme')
Saving solutions to the server... success.
Part 1 has no valid solution.
- For values (1, 0, 1, -1, 1) the min value is not 2.
Did you consider value at the vertex of the parabola?
- Expression minmax(1, 0, 1, -1, 1) returns (2, 2) instead of (1, 2).
>>>

```python
import numpy

def minmax(a, b, c, d, e):
    def f(x):
        return a*numpy.square(x) + b*x + c
    points = [d, e]
    vertex_x = -b / (2*a)
    if d <= vertex_x and vertex_x <= e:
        points.append(vertex_x)
    values = [f(point) for point in points]
    return (min(values), max(values))

```

Console

Python 1 | IP: Kernel 1 | temp.py

```
>>> runfile('C:/Users/gregor/Desktop/cadgme/minmax_1.py', wdir='C:/Users/gregor/Desktop/cadgme')
Saving solutions to the server... success.
Part 1 has a valid solution.
>>>
```

Solutions

Minmax 1

Find the max and min values of a function $ax^2 + bx + c$ an a closed interval.

1. part

Write a function minmax(a, b, c, d, e) that returns min value m and max value M of the function $ax^2 + bx + c$ an a closed interval $[d, e]$. Values must be returned as a tuple (m, M).

Example:

```
>>> minmax(1, 0, 1, -1, 1)
(1, 2)
>>> minmax(1, 0, 1, 1, 2)
(2, 5)
```

Minmax 1

Simple problems

1. part

Your solution

```
import numpy

def minmax(a, b, c, d, e):
    def f(x):
        return a*numpy.square(x) + b*x + c
    points = [d, e]
    vertex_x = -b / (2*a)
    if d <= vertex_x and vertex_x <= e:
        points.append(vertex_x)
    values = [f(point) for point in points]
    return (min(values), max(values))
```

Official solution

```
def minmax(a, b, c, d, e):
    # Store the x-coordinates of the candidates in the array.
    candidates_x = [d, e, -b/(2*a)]
    # Filter them out so that they lie on the interval [e, f]
    candidates_x = [c for c in candidates_x if c<=e and c>=d]
    # Compute function values at this points
    values = [a*x**2 + b*x + c for x in candidates_x]
    return (min(values), max(values))
```

Tomo: teachers perspective

- List of courses are available
- Editing options in classes taught by the teacher
- Easily tracks student progress
- Can add/edit problem sets / exercises

CADGME 2016

- 0% Octave
- 20% Workshop
- 0% Excercise 14

Računalništvo 1

- 14% Požrešna metoda
- 38% Iskanje z bisekcijo
- 27% Deli in vladaj

Drugi koraki v Python

Matija Lokar, UROŠ VAUPOTIČ

@{Fakulteta za matematiko in

Informatika

Andrej Brodnik, Valentin Kragelj, VALENTIN KRAGELJ

@{Gimnazija Vič}

Informatika

Klemen Bajec

@{Gimnazija Vič}

Računalništvo 1

8%	Sklad (1. del)
14%	Sklad (2. del)
46%	Vrsta
39%	Verižni seznam
31%	Dvojiška drevesa

Fakulteta za matematiko in fiziko

Praktična matematika, 3. letnik

[illegible]
Gregor Jerse
Gregor Jerše
Matija Lokar
[illegible]

[illegible]

Sklad (2. del)

RPN kalkulator

Ko z enostavnim kalkulatorjem želimo izračunati malo zapletenejši račun, vedno nastanejo težave z zapisom oklepajev. Izkaže pa se, da se lahko uporabi oklepajev v celoti izognemo z obrnjenim poljskim zapisom (reverse Polish notation oz. na krajše RPN).

V tem zapisu operacij ne pišemo med argumenti, temveč za njimi. Tako namesto `4 + 5` pišemo `4 5 +`. Če želimo izračunati `(2 + 4) * 3`, pa napišemo `2 4 + 3 *`. Ko napišemo `2 4 +`, je to isto, kot če bi napisali `6`, in ko temu dodamo še `3 *`, dobimo iskani rezultat `18`.

V splošnem števila dajemo na sklad, z operacijo pa s sklada vzamemo dve vrhnji števili, nanj pa vrnemo rezultat operacije.

Implementacijo razreda `Sklad` lahko dobite tukaj.

Vrsta

- Delo z vrsto
- Zavetišče
- Vrsta z dvema skladoma
- Sladoled
- Družabna vrsta

Verižni seznam

- Vozel
- Še o vozlih
- Izštevanka
- Urejanje verižnih seznamov

1. part

The formula for converting Fahrenheit d

Write a function $C(F)$ that implements t

Minmax 1

Find the max and min values of a fun

1. part

Write a function minmax(a, b, c, d, e) tha
interval $[d, e]$. Values must be returned a

Example:

```
octave> minmax(1, 0, 1, -1, 1)
(1, 2)
octave> minmax(1, 0, 1, 1, 2)
(2, 5)
```

+ Add new problem

Add new problem

Title

Title

Description

Description

Language

Python 3

Add Cancel

Simple problems

Simple problems

Minmax 1

File for solving
File for editing

Find the max and min values of a function $ax^2 + bx + c$ an a closed interval.

1. part

Write a function minmax(a, b, c, d, e) that returns min value m and max value M of the function $ax^2 + bx + c$ an a closed interval $[d, e]$. Values must be returned as a tuple (m, M).

Example:

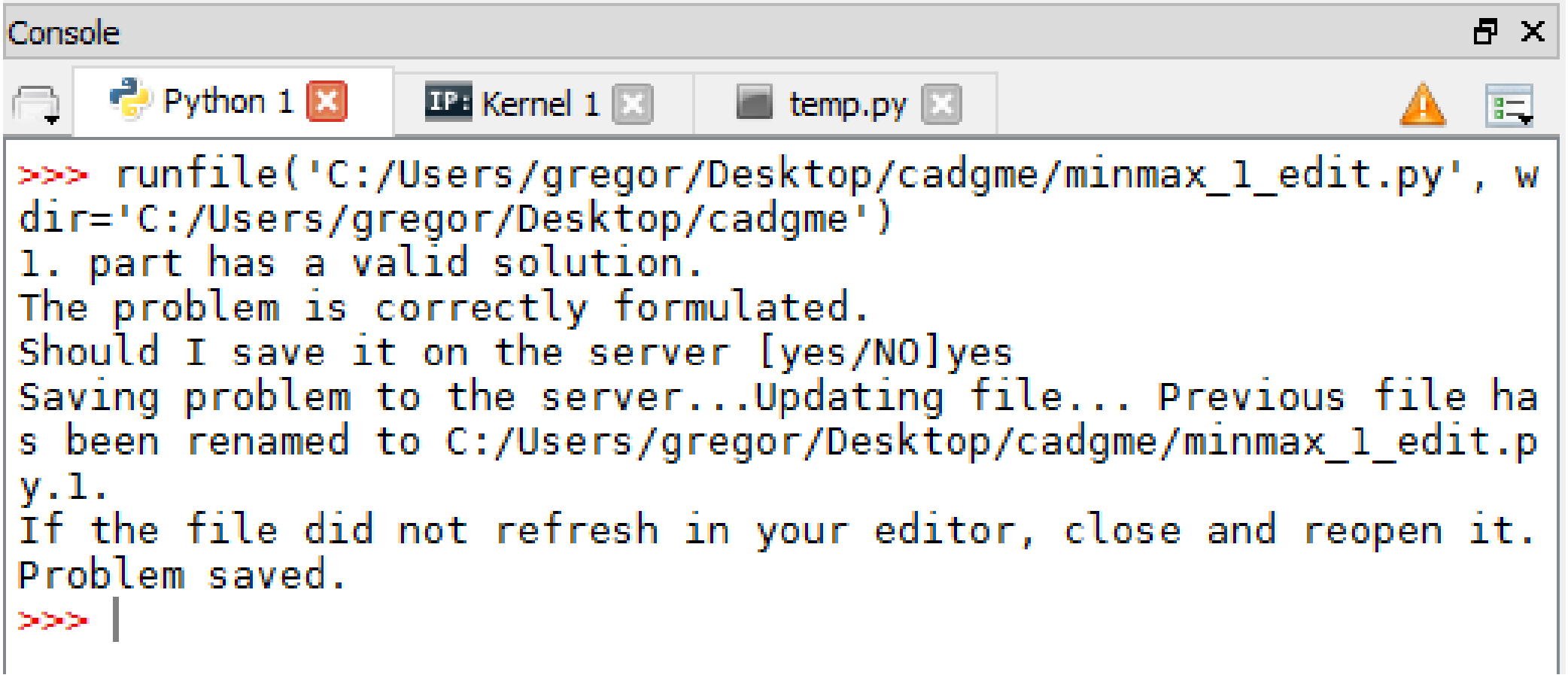
```
>>> minmax(1, 0, 1, -1, 1)
(1, 2)
>>> minmax(1, 0, 1, 1, 2)
(2, 5)
```

```
with open(__file__, encoding='utf-8') as f:
    source = f.read()
exec(source[source.find("# =L=I=B=""R=A=R=Y=@="):])
problem = extract_problem(__file__)
Check.initialize(problem['parts'])

# ==========================================================================
# Minmax 1
#
# Find the max and min values of a function $ax^2 + bx + c$ an a closed interval.
# ==========================================================================@002828=
# Write a function minmax(a, b, c, d, e) that returns min value $m$ and max
# value $M$ of the function $ax^2 + bx + c$ an a closed interval $[d, e]$.
# Values must be returned as a tuple $(m, M)$.
#
# Example:
#
#     >>> minmax(1, 0, 1, -1, 1)
#     (1, 2)
#     >>> minmax(1, 0, 1, 1, 2)
#     (2, 5)
# ==========================================================================
def minmax(a, b, c, d, e):
    # Store the x-coordinates of the candidates in the array.
    candidates_x = [d, e, -b/(2*a)]
    # Filter them out so that they lie on the interval [e, f]
    candidates_x = [c for c in candidates_x if c<=e and c>=d]
    # Compute function values at this points
    values = [a*x**2 + b*x + c for x in candidates_x]
    return (min(values), max(values))
```

```
def minmax(a, b, c, d, e):
    # Store the x-coordinates of the candidates in the array.
    candidates_x = [d, e, -b/(2*a)]
    # Filter them out so that they lie on the interval [e, f]
    candidates_x = [c for c in candidates_x if c<=e and c>=d]
    # Compute function values at this points
    values = [a*x**2 + b*x + c for x in candidates_x]
    return (min(values), max(values))


Check.part()
(m, M) = minmax(1, 0, 1, -1, 1)
if M == 1:
    Check.error(("For values {0} the max value is not 1. "
                 "Did you consider the endpoints of the interval?"),
                (1, 0,1,-1,1), )
if m != 1:
    Check.error(("For values {0} the min value is not {1}."
                 "Did you consider value at the vertex of the parabola?"),
                (1, 0,1,-1,1), m)
Check.equal('minmax(1, 0, 1, -1, 1)', (1, 2))
```


Console
Python 1
IP: Kernel 1
temp.py
>>> runfile('C:/Users/gregor/Desktop/cadgme/minmax_1_edit.py', w
dir='C:/Users/gregor/Desktop/cadgme')
1. part has a valid solution.
The problem is correctly formulated.
Should I save it on the server [yes/NO]yes
Saving problem to the server...Updating file... Previous file ha
s been renamed to C:/Users/gregor/Desktop/cadgme/minmax_1_edit.p
y.1.
If the file did not refresh in your editor, close and reopen it.
Problem saved.
>>>

minmax_1_edit.m

```
10  # NE BRIŠI prvih vrstic
11
12  # ================================================================================
13  # Minmax 1
14  #
15  # Find the max and min values of a function $ax^2 + bx + c$.
16  # ================================================================================@002831=
17  # Write a function minmax(a, b, c, d, e) that returns min value $m$ and max
18  # value $M$ of the function
19  # $ax^2 + bx + c$ an a closed interval $[d, e]$.
20  # Values must be returned as a vector $[m; M]$.
21  #
22  # Example:
23  #
24  #     octave> minmax(1, 0, 1, -1, 1)
25  #     (1, 2)
26  #     octave> minmax(1, 0, 1, 1, 2)
27  #     (2, 5)
28  # ================================================================================
29  function mM = minmax(a, b, c, d, e)
30      candidates_x = [d; e];
31      vertex_x = -b/(2*a);
32      if (vertex_x >= d && vertex_x <= e)
33          candidates_x = [candidates_x; vertex_x];
34      endif
35      # Compute function values at this points
36      values = [a*candidates_x.*candidates_x + b*candidates_x + c]
37      mM = [min(values); max(values)]
38  endfunction
39
40  check_part()
41  ret = minmax(1, 0, 1, -1, 1);
42  m = ret(1);
43  M = ret(2);
44  if (M == 1)
45      check_error('For values (1, 0, 1, -1, 1) the max value is not 1. Did you consider the endpoints
46  endif
```

minmax_1_edit.m

```
10  # NE BRIŠI prvih vrstic
11
12  # ================================================================================
13  # Minmax 1
14  #
15  # Find the max and min values of a function $ax^2 + bx + c$.
16  # ================================================================================@002831=
17  # Write a function minmax(a, b, c, d, e) that returns min value $m$ and max
18  # value $M$ of the function
19  # $ax^2 + bx + c$ an a closed interval $[d, e]$.
20  # Values must be returned as a vector $[m; M]$.
21  #
22  # Example:
23  >> minmax_1_edit
24
25  ans =  1
26  The problem in correctly formulated. Should I save it on the server? (yes/no)
27
28
29  Command Window   Editor   Documentation
30
31
32      if (vertex_x >= d && vertex_x <= e)
33          candidates_x = [candidates_x; vertex_x];
34      endif
35      # Compute function values at this points
36      values = [a*candidates_x.*candidates_x + b*candidates_x + c]
37      mM = [min(values); max(values)]
38  endfunction
39
40  check_part()
41  ret = minmax(1, 0, 1, -1, 1);
42  m = ret(1);
43  M = ret(2);
44  if (M == 1)
45      check_error('For values (1, 0, 1, -1, 1) the max value is not 1. Did you consider the endpoints
46  endif
```

Tomo: feedback

- Included inside test cases
- Test cases are programs written in the chosen language
- Test cases can be arbitrarily complex
- There is class Check (in Python3) that simplifies common tasks
- Students’ source code is available for inspection in the test program

```
20 from numpy.linalg import eigvals
21 from numpy import prod
22
23 def product_of_eigenvalues(A):
24     eigs = eigvals(A)
25     return prod(eigs)
26
27 Check.part()
28
29 import ast
30 tree = ast.parse(Check.current_part['solution'])
31 for node in ast.walk(tree):
32     if isinstance(node, ast.Call):
33         name = node.func.id
34         if name == 'det':
35             Check.error('det function should not be used')
36
37 Check.equal('float(product_of_eigenvalues([[1, 0], [0, 1]]))', 1)
38 Check.equal('float(product_of_eigenvalues([[-1, 3], [2, 1]]))', -7)
39 Check.secret(product_of_eigenvalues([[2, 1], [0, 1]]))
40
```

Projekt Tomo

- Open Source
- Code is available on GitHub
- https://github.com/matijapretnar/projekt-tomo