



Kako drugo leto še lažje in hitreje rešiti naloge na tekmovanju

Matija Lokar,

FAKULTETA ZA MATEMATIKO IN FIZIKO
UNIVERZA V LJUBLJANI

Matija.Lokar@fmf.uni-lj.si

HOW DOES COMPUTER
PROGRAMMING WORK?

MAGIC.



全高漫

PROGRAMIRANJE JE
(med drugim tudi) VEŠČINA

In za vsako veččino



PRACTICE
MAKES PERFECT



Zelo pomembno

**Kdaj je naloga
rešena?**

Sestavi program, ki prebere celo število in izpiše njegovo obratno vrednost

```
n = int(input('Vnesi celo število: '))  
rec = 1/n  
print('Obratna vrednost je ', rec)
```

<

Shell

Python 3.5.1

>>> %Run a.py

```
Vnesi celo število: 12  
Obratna vrednost je 0.08333333333333333
```

>>> %Run a.py

```
Vnesi celo število: 5  
Obratna vrednost je 0.2
```

>>> %Run a.py

```
Vnesi celo število: 4  
Obratna vrednost je 0.25
```




Practice does not make perfect.
Only perfect practice makes perfect.

— *Vince Lombardi* —

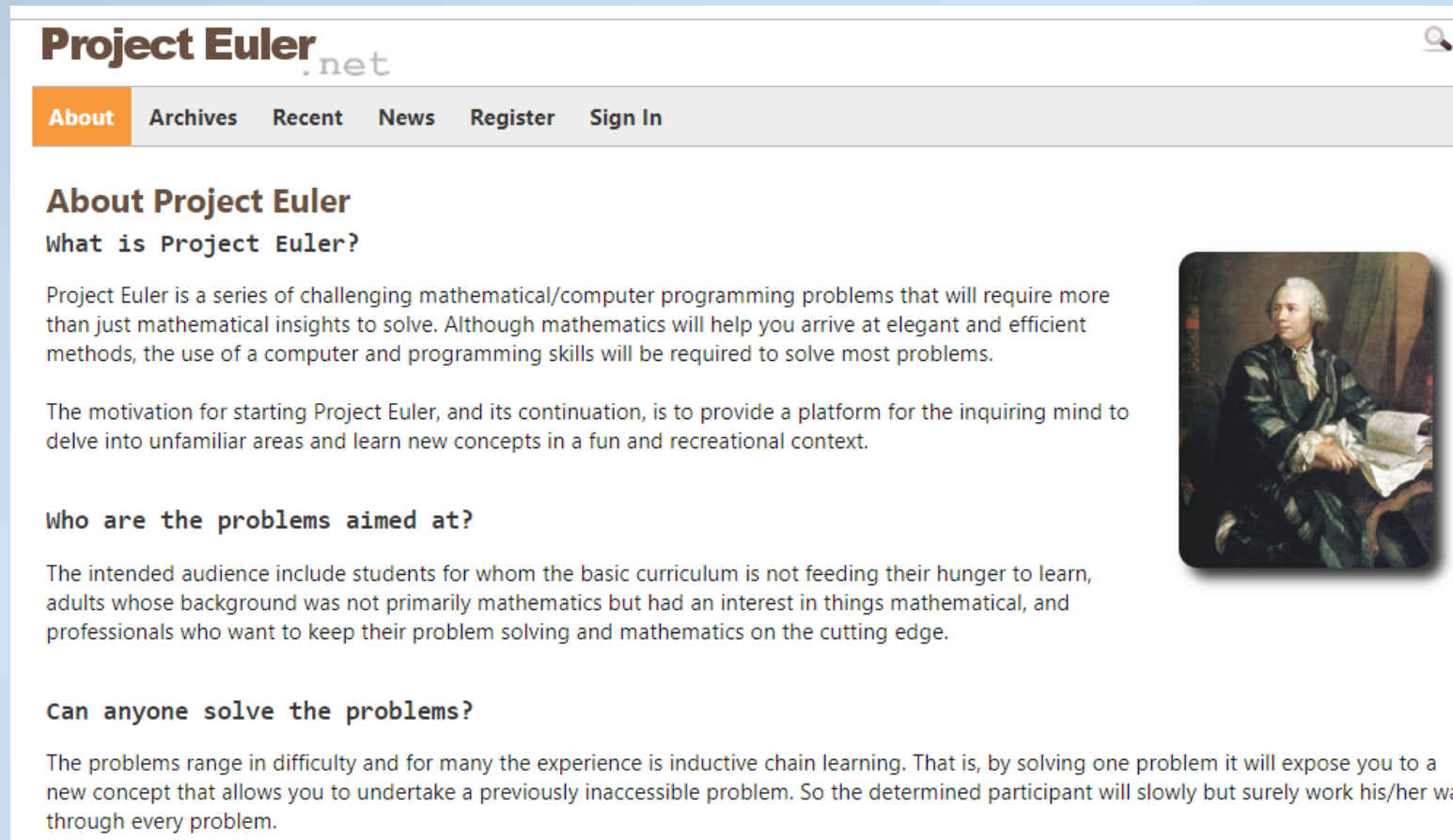
AZ QUOTES

Sistemi za avtomatsko preverjanje pravilnosti programov

Različni pristopi

Project Euler

<https://projecteuler.net/>



The screenshot shows the Project Euler website. At the top left is the logo "Project Euler .net". To the right of the logo is a search icon. Below the logo is a navigation bar with links: "About", "Archives", "Recent", "News", "Register", and "Sign In". The "About" link is highlighted in orange. Below the navigation bar is the section "About Project Euler". Under this section is the heading "What is Project Euler?". The text below reads: "Project Euler is a series of challenging mathematical/computer programming problems that will require more than just mathematical insights to solve. Although mathematics will help you arrive at elegant and efficient methods, the use of a computer and programming skills will be required to solve most problems." Below this is another paragraph: "The motivation for starting Project Euler, and its continuation, is to provide a platform for the inquiring mind to delve into unfamiliar areas and learn new concepts in a fun and recreational context." To the right of this text is a portrait of Leonhard Euler. Below the portrait is the heading "Who are the problems aimed at?". The text below reads: "The intended audience include students for whom the basic curriculum is not feeding their hunger to learn, adults whose background was not primarily mathematics but had an interest in things mathematical, and professionals who want to keep their problem solving and mathematics on the cutting edge." Below this is the heading "Can anyone solve the problems?". The text below reads: "The problems range in difficulty and for many the experience is inductive chain learning. That is, by solving one problem it will expose you to a new concept that allows you to undertake a previously inaccessible problem. So the determined participant will slowly but surely work his/her way through every problem."

Project Euler.net


[About](#) [Archives](#) [Recent](#) [News](#) [Register](#) [Sign In](#)

About Project Euler

What is Project Euler?

Project Euler is a series of challenging mathematical/computer programming problems that will require more than just mathematical insights to solve. Although mathematics will help you arrive at elegant and efficient methods, the use of a computer and programming skills will be required to solve most problems.

The motivation for starting Project Euler, and its continuation, is to provide a platform for the inquiring mind to delve into unfamiliar areas and learn new concepts in a fun and recreational context.



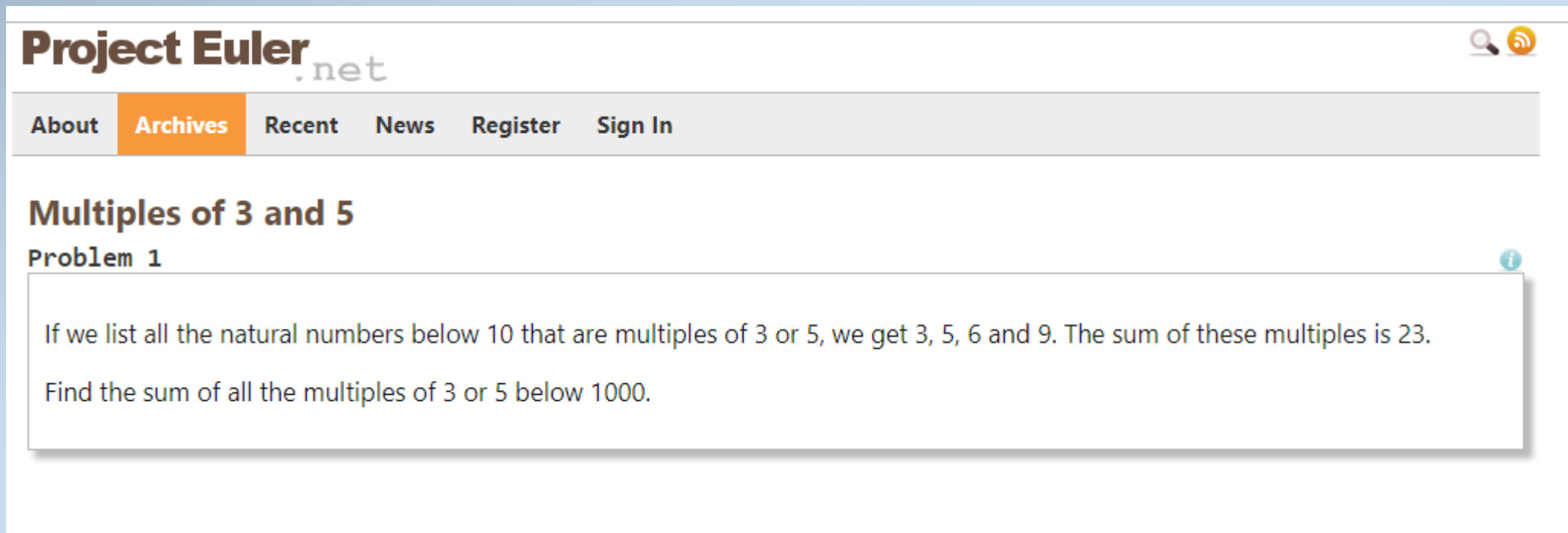
Who are the problems aimed at?



The intended audience include students for whom the basic curriculum is not feeding their hunger to learn, adults whose background was not primarily mathematics but had an interest in things mathematical, and professionals who want to keep their problem solving and mathematics on the cutting edge.

Can anyone solve the problems?

The problems range in difficulty and for many the experience is inductive chain learning. That is, by solving one problem it will expose you to a new concept that allows you to undertake a previously inaccessible problem. So the determined participant will slowly but surely work his/her way through every problem.

Naloge take, da je rešitev vsake neko število



Project Euler.net  

[About](#) [Archives](#) [Recent](#) [News](#) [Register](#) [Sign In](#)

Multiples of 3 and 5

Problem 1

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

Oddamo samo to število

Project Euler.net Logged in as **lokar_test**
Wed, 20 Mar 2019, 11:55

[About](#) [Archives](#) [Recent](#) [Progress](#) [Account](#) [News](#) [Friends](#) [Statistics](#) [Sign Out](#)

← →

Even Fibonacci numbers

Problem 2


Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...


By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Answer:

Confirmation Code:

Confi 

Click image for new code



Click image for new code

[About](#)[Archives](#)[Recent](#)[Progress](#)[Account](#)[News](#)[Friends](#)[Statistics](#)[Sign Out](#)

Sorry, but the answer you gave appears to be incorrect.

[Go back to Problem 2.](#)

















Congratulations, the answer you gave to problem 1 is correct.

You are the 834625th person to have solved this problem.

This problem had a difficulty rating of 5%. The highest difficulty rating you had previously solved was 0%. [i](#)
This is a new record. Well done!

Splača se pogledati opis uradnih rešitev (in objave v Forumu)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 Go to Problem:

 ID 	Description / Title	 Solved By 	 Difficulty 		
1	Multiples of 3 and 5	834623		   KAPb14 2 hours	217 posts
2	Even Fibonacci numbers	667194			
3	Largest prime factor	476599			

```
target=999
sum=0
for i=1 to target do
if (i mod 3=0) or (i mod 5)=0 then sum:=sum+i
output sum
```

`SumDivisibleBy(3)+SumDivisibleBy(5)-SumDivisibleBy(15)`

Let's look at the details of our function and take as example $n=3$.

We would have to add:

$$3+6+9+12+\dots+999=3*(1+2+3+4+\dots+333)$$

For $n=5$ we would get:

$$5+10+15+\dots+995=5*(1+2+\dots+199)$$

Now note that $199=995/5$ but also $999/5$ rounded down to the nearest integer.

In many programming languages there exists a separate operator for that: `div` or `\`.

If we now also note that $1+2+3+\dots+p=\frac{1}{2}*p*(p+1)$ our program becomes:

```
target=999
```

```
Function SumDivisibleBy(n)
  p=target div n
  return n*(p*(p+1)) div 2
EndFunction
```

```
Output SumDivisibleBy(3)+SumDivisibleBy(5)-SumDivisibleBy(15)
```

```
def pel_1(n):  
    '''Vsota števil od 1 do n-1 deljivih s 3 ali 5'''  
    vsota = 0  
    for i in range(1000):  
        if i % 3 == 0 or i % 5 == 0:  
            vsota += i  
    return vsota
```

```
def vsota(n, k):  
    '''Vsota števil od 1 do n-1 deljivih s k'''  
    koliko = n // k  
    return k * koliko * (koliko + 1) // 2  
  
def pel_2(n):  
    '''Vsota števil od 1 do n-1 deljivih s 3 ali 5'''  
    vs3 = vsota(n, 3)  
    vs5 = vsota(n, 5)  
    vs15 = vsota(n, 15)  
    return vs3 + vs5 - vs15
```

```
import timeit
čas1 = timeit.timeit('pe1_1(10000)', globals = globals(),
number = 500)
čas2 = timeit.timeit('pe1_2(10000)', globals = globals(),
number = 500)
print(čas1, čas2, str(int((čas1 - čas2)/čas2)) + 'x hitreje')
```

```
>>> %Run pe.py
0.061132902 0.00056788700000000169 106x hitreje
>>> %Run pe.py
0.060167494 0.00050418799999999888 118x hitreje
>>> %Run pe.py
0.059383569 0.00050809400000000006 115x hitreje
>>> %Run pe.py
0.059155813 0.00049937999999999937 117x hitreje
```

Kattis

- <https://open.kattis.com>
- Strogo predpisana oblika rezultata

Output

If all the houses are already connected to the internet, output one line containing the string `Connected`. Otherwise, output a list of house numbers in increasing order, one per line, representing the houses that are not yet connected to the internet.

Sample Input 1

```
6 4
1 2
2 3
3 4
5 6
```

Sample Output 1

```
5
6
```

- Smiselno za tekmovanja
- Malo manj za priprave

"matija"



REŠITEV



"ajita"

= ?

"ajitam"

Omejena povratna informacija

3918041	2019-03-12 14:59:41	An I for an Eye	✘ Wrong Answer	0.02 s	Python 3
3918034	2019-03-12 14:57:47	An I for an Eye	✘ Run Time Error	0.02 s	Python 3
3869317	2019-02-28 12:22:07	Where's My Internet??	✘ Run Time Error	0.03 s	Python 2
3869313	2019-02-28 12:21:22	Where's My Internet??	✘ Run Time Error	0.03 s	Python 2
3869286	2019-02-28 12:12:17	Where's My Internet??	✘ Time Limit Exceeded	> 1.00 s	Python 3
3860088	2019-02-26 12:00:40	Where's My Internet??	✔ Accepted	0.73 s	Python 3

Smiselno za tekmovanja
Malo manj za priprave



Ein Unternehmen der Zeppelin Park Group
ZEPELIN on Print
Plakate ankleben
verboten!



<https://www.projekt-tomo.si>

+01110

Vaš osebni učitelj programiranja.

Speaker Deck

Talk by Matija Pretnar

Kako uporabljamo
Projekt Tomo?



ToonClips.com

#4400

service@toonclips.com

<http://getdrawings.com/butler-drawing#butler-drawing-4.jpg>

B2-PKP

Elektro in računalniška šola Velenje

Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

Fakulteta za matematiko in fiziko UL

Fakulteta za naravoslovje in matematiko, UM

Fakulteta za računalništvo in informatiko UL

Gimnazija Bežigrad

Gimnazija in ekonomska srednja šola Trbovlje

Gimnazija Jesenice

Gimnazija Kranj

Gimnazija Lava

Gimnazija Litija

Gimnazija Murska Sobota

Gimnazija Nova Gorica

Gimnazija Ptuj

Gimnazija Šentvid

Gimnazija Šiška

Gimnazija Vič

I. gimnazija v Celju

Inštitut 4.0

OŠ Dobravlje

OŠ Dragomelj

OŠ Griže

OŠ Ivana Cankarja Ljutomer

OŠ Janka Glazerja Ruše

OŠ Ledina

OŠ Leskovec pri Krškem

OŠ Matije Čopa Kranj

OŠ Tomo Brejc

OŠ Zbora odposlancev Kočevje

Projekt NAPOJ

ProNAL

Prva gimnazija Maribor

Seminar

Škofijska klasična gimnazija

USTANOVE, UPORABNIKI SISTEMA TOMO

ProNAL

poletni 2018

OŠ Matije Čopa Kranj

OŠ Tomo Brejc

OŠ Zbora odposlancev Kočevje

Projekt NAPOJ

ProNAL

ProNAL peskovnik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Še en peskovnik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Starejši - učbenik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Starejši - zbirka nalog ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-dopolni ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-ostalo ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-Parsons ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

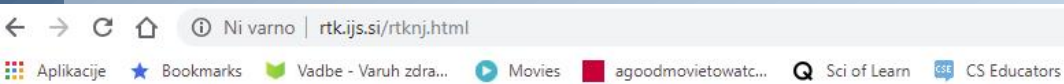
@{ProNAL}

Tekmovanje-popravi ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Vir nalog



- [Glavna stran](#)
- [Rezultati](#)
- [Razpis](#)
- [Pravilnik](#)
- [Prijave](#)
- [Off-line naloga \[novo\]](#)
- [Tekmovanje v razvoju spletnih aplikacij](#)
- [Izobraževalni videoposnetki](#)
- [Bober](#)

- [Tekmovanje 2018](#)
- [Tekmovanje 2017](#)
- [Tekmovanje 2016](#)
- [Tekmovanje 2015](#)
- [Tekmovanje 2014](#)
- [Tekmovanje 2013](#)
- [Tekmovanje 2012](#)
- [Tekmovanje 2011](#)
- [Tekmovanje 2010](#)
- [Tekmovanje 2009](#)
- [Tekmovanje 2008](#)
- [Tekmovanje 2007](#)
- [Tekmovanje 2006](#)
- [Prejšnja tekmovanja](#)
- [Zbirka nalog](#)

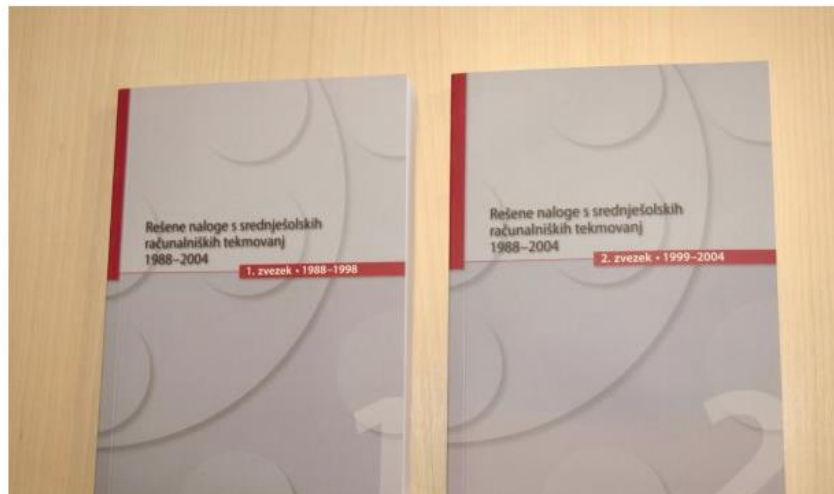
- [Mailing lista](#)
- [Povezave](#)
- [Pokrovitelji](#)
- [Kontakt](#)

Zbirka rešenih nalog s srednješolskih računalniških tekmovanj

V tej zbirki so naloge in rešitve z republiških srednješolskih tekmovanj v znanju računalništva (1988-2004) in tekmovanj v poznavanju Unixa (1999-2004).

- [PDF s hiperlinki](#) (primeren za gledanje na računalniku): [A4](#), [A5](#).
- [PDF brez hiperlinkov](#) (primeren za tiskanje): [A4](#), [A5](#).
- [Izvorna koda](#) rešitev v zbirki (vsebuje tudi testne primere iz let 1995, 1996 in 1997).
- Testni primeri za 3. skupino: [2004](#), [2003](#), [2002](#), [2001](#).

Zbirka je izšla tudi v knjižni obliki:



+OMMO Tekmovanje Matija Lokar (odjava) ?

- 0% Sofinasiranje projekta
- 0% rtk 1988
- 0% rtk 1996
- 0% rtk 1998
- 0% rtk 1999
- 0% rtk 2000
- 0% rtk 2001
- 0% rtk 2002
- 0% rtk 2004

Žiga Flajs
Katarina Koderman
Romi Koželj
Vid Kriznar
Karel Križnar
Projektotomo Lokar
Klementina Pirc
Mitja Rozman
Kristijan Šaver
Jelko Urbančič
ProNAL

Naloge s tekmovanja ACM - RTK (<http://rtk.ijs.si/>) V sistem TOMO so bile prenešene v sklopu projekta ProNAL (<http://naslokar.fmf.uni-lj.si/ProNAL/>)



A v različnih oblikah

[OŠ Matije Čopa Kranj](#)

[OŠ Tomo Brejc](#)

[OŠ Zbora odposlancev Kočevje](#)

[Projekt NAPOJ](#)

[ProNAL](#)

ProNAL peskovnik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Še en peskovnik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Starejši - učbenik ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Starejši - zbirka nalog ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-dopolni ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-ostalo ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-Parsons ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje-popravi ☆

Žiga Flajs, Katarina Koderman, Romi Kož...

@{ProNAL}

Tekmovanje

1996.1.2 Opiši postopek, ki preveri, če sta dve ogrlici enaki. Ogrlica je sestavljena iz N kroglic, ki so nanizane ena za drugo. Ogrlico predstavimo s tabelo znakov, kjer znaki predstavljajo barve kroglic.

```
type Ogrlica = array [1..N] of char;
```

Ker so ogrlice krožne, moramo tako obravnavati tudi tabelo. Primeri ogrlic:

```
'1234567890' je enaka '7890123456'  
'1234567890' ni enaka '1234567809'
```

Opišite postopek, ki na vhodu dobi dve ogrlici, na izhodu pa vrne vrednost `true`, če sta ogrlici enaki, in vrednost `false`, če nista.

1996.1.2 (napisi)



Ogrlice

1. podnaloga

Maja v prostem času rada sestavlja ogrlice, pri tem pa za vsako uporabi n kroglic, ki jih naniza eno za drugo. Vsako izdelano ogrlico opiše z nizom števil od 0 do 9, pri čemer vsako število označuje določeno barvo.

Primer zapisa ogrlice: `'12345'`

Naloga

Pomagaj Maji sestaviti funkcijo `primerjaj_ogrlici(ogrlica1, ogrlica2)`, s katero bo lahko preverila ali sta dve ogrlici enaki ali ne. Funkcija naj vrne `True`, če sta ogrlici enaki, in `False`, če se razlikujeta.

Ogrlici sta enaki, če sta enaka njuna niza. Ker so ogrlice krožne, lahko za prvo kroglico izberemo poljubno kroglico iz ogrlice. Enako torej velja za njen niz. Predpostavimo, da je število n za ogrlici, ki ju primerjamo enako.

Vhodni podatki

Dva niza, ki predstavljata ogrlici.

Izhodni podatki

Funkcije vrne vrednost `True` ali `False`.

Primer:

```
>>> primerjaj_ogrlici('1234567890', '7890123456')  
True  
>>> primerjaj_ogrlici('1234567890', '1234567809')  
False
```


Tekmovanje - dopolni

Naloga

Pomagaj Maji dopolniti funkcijo `primerjaj_ogrlici(ogrlica1, ogrlica2)`, s katero bo lahko preverila ali sta dve ogrlici enaki ali ne. Funkcija naj vrne `True`, če sta ogrlici enaki, in `False`, če se razlikujeta. Mesto dopolnjevanja je označeno z `###`.

Ogrlici sta enaki, če sta enaka njuna niza. Ker so ogrlice krožne, lahko za prvo kroglico izberemo poljubno kroglico iz ogrlice. Enako torej velja za njen niz. Predpostavimo, da je število n za ogrlici, ki ju primerjamo enako.

```
def primerjaj_ogrlici(ogrlica1, ogrlica2):
    '''Funkcija vrne True, če sta ogrlici enaki in False sicer.'''

    n = ###
    for i in range(n):
        ogrlica3 = ogrlica1[###] + ogrlica2[###]
        if ogrlica1 == ogrlica3:
            return True
    return False
```

Tekmovanje - popravi

Naloga

Pomagaj Maji popraviti funkcijo `primerjaj_ogrlici(ogrlica1, ogrlica2)`, s katero bo lahko preverila ali sta dve ogrlici enaki ali ne. Funkcija naj vrne `True`, če sta ogrlici enaki, in `False`, če se razlikujeta.

Ogrlici sta enaki, če sta enaka njuna niza. Ker so ogrlice krožne, lahko za prvo kroglico izberemo poljubno kroglico iz ogrlice. Enako torej velja za njen niz. Predpostavimo, da je število n za ogrlici, ki ju primerjamo enako.

```
def primerjaj_ogrlici(ogrlica1, ogrlica2):
    '''Funkcija vrne True, če sta ogrlici enaki in False sicer.'''

    n = len(ogrlica1)
    for i in range(n):
        ogrlica3 = ogrlica2[i:] + ogrlica2[:i]
        if ogrlica1 == ogrlica3:
            return False
    return True
```

Tekmovanje - Parsons

1. podnaloga

Maja v prostem času rada sestavlja ogrlice, pri tem pa za vsako uporabi n kroglic, ki jih naniza eno za drugo. Vsako izdelano ogrlico opiše z nizom števil od 0 do 9, pri čemer vsako število označuje določeno barvo.

Primer zapisa ogrlice: `'12345'`

Naloga

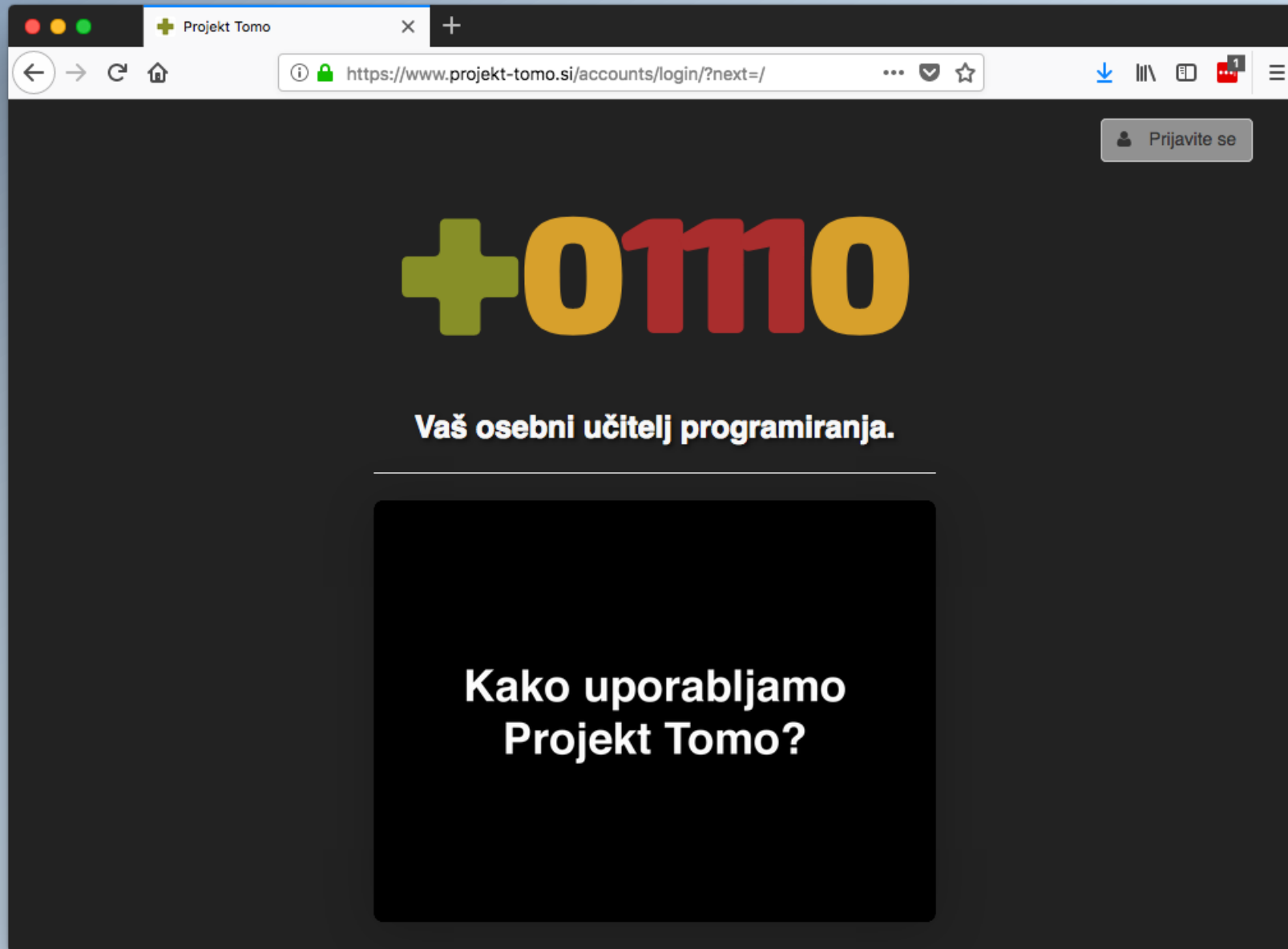
Maja je napisala funkcijo `primerjaj_ogrlici(ogrlica1, ogrlica2)`, s katero bo lahko preverila ali sta dve ogrlici enaki ali ne. Funkcija vrne `True`, če sta ogrlici enaki, in `False`, če se razlikujeta.

Ogrlici sta enaki, če sta enaka njuna niza. Ker so ogrlice krožne, lahko za prvo kroglico izberemo poljubno kroglico iz ogrlice. Enako torej velja za njen niz. Predpostavimo, da je število n za ogrlici, ki ju primerjamo, enako.

Ko pa je Maja na hitro zapustila računalnik, jih je mlajši bratec ponagajal. Vrstice v funkciji je pomešal. Na srečo ni iz njih zbrisal nobenega znaka.

Pomagaj Maji in vrstice ponovno uredi v pravilni vrstni red!

Kako rešujemo naloge na Tomu



Obiščemo www.projekt-tomo.si

The image shows a web browser window displaying the login page for 'Projekt Tomo'. The browser's address bar shows the URL <https://www.projekt-tomo.si/accounts/login/?next=/>. The main content area has a dark background with the 'Projekt Tomo' logo in large, colorful letters. Below the logo, the text reads 'Vaš osebni učitelj programiranja.' (Your personal programming teacher). A large black box contains the question 'Kako uporabljamo Projekt Tomo?' (How do we use Projekt Tomo?). On the right side, there is a white sidebar with the heading 'Prijavite se z zunanjim ponudnikom' (Log in with an external provider). It offers three login options: 'ArnesAAI' (with a building icon), 'Facebook', and 'Google'. Below these options, there is explanatory text: 'Prijavite se lahko z enim od zgoraj naštetih zunanjih ponudnikov. Slovenskim učencem in učiteljem priporočamo, da uporabijo enotno prijavo ArnesAAI.' (You can log in with one of the external providers listed above. We recommend Slovenian students and teachers to use the single sign-on ArnesAAI). At the bottom of the sidebar, it states: 'S prijavo se strinjate z našimi Pogoji poslovanja. Spletna stran za delovanje uporablja piškotke.' (By logging in, you agree to our Terms of Service. The website uses cookies for operation).

Izberemo način prijave



Programiranje 1

75% Popravi program

22% Izpisi

33% Okolje



Prvi koraki v Python

0% Seznami

0% Zanke

0% Pogojni stavek

Drugi koraki v Python



Matjaž Lohar

@{Fakulteta za matematiko in fiziko}

Informatika



Andrej Bratkovič, Viktorija Kravos

@{Gimnazija Vič}

Informatika



Uroš Bratkovič

@{Gimnazija Jurija Vege Idrija}

Na začetni strani **izberemo sklop**, ki ga bomo reševali.



Programiranje 1

75% Popravi program

22% Izpisi

33% Okolje



Prvi koraki v Python

0% Sezname

0% Pogojni stavek



Drugi koraki v Python



Mirko Luber

@{Fakulteta za matematiko in fiziko}

Informauka



Andrej Bradač, Valerija Krpanj

@{Gimnazija Vič}

Informauka



Marko Bradač

@{Gimnazija Jurija Vege Idrija}

Na začetni strani **izberemo sklop**, ki ga bomo reševali.



Pogojni stavek

Tukaj je nabor vaj iz pogojnih stavkov.

Datumi



Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje gregorijanski koledar.

1. podnaloga

Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto prestopno (po gregorijanskem koledarju). Zgled:

```
>>> je_prestopno(2004)
True
>>> je_prestopno(1900)
False
```

Deljivost



1. podnaloga

Datumi



Deljivost



Bratska igra



Avtomobilska zavarovanja



Enačba



Hladilnik



Indeks telesne mase



Indiana Jones



Na strani sklopa **izberemo nalogo**, ki jo bomo reševali.



Pogojni stavek

Tukaj je nabor vaj iz pogojnih stavkov.

Datumi

Koledar, ki ga trenutno uporabljamo koledar.

1. podnaloga

Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto prestopno (po gregorijanskem koledarju). Zgled:

```
>>> je_prestopno(2004)
True
>>> je_prestopno(1900)
False
```

Deljivost

1. podnaloga

Datoteka
z nalogo



Avtomobilska zavarovanja



Enačba



Hladilnik



Indeks telesne mase



Indiana Jones



Na strani sklopa **izberemo nalogo**, ki jo bomo reševali.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje
# gregorijanski koledar.
# =====@000272=
# 1. podnaloga
# Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto
# prestopno (po gregorijanskem koledarju). Zgled:
#
#     >>> je_prestopno(2004)
#     True
#     >>> je_prestopno(1900)
#     False
# =====
```

Ln: 16 Col: 0

V prenešeni datoteki so vpisana **navodila naloge**.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje
# gregorijanski koledar.
# =====@000272=
# 1. podnaloga
# Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto
# prestopno (po gregorijanskem koledarju). Zgled:
#
#     >>> je_prestopno(2004)
#     True
#     >>> je_prestopno(1900)
#     False
# =====

def je_prestopno(leto):
    return leto % 4 == 0
```

Ln: 18 Col: 24

Datoteko **dopolnimo** z rešitvijo in **poženemo**.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Ko
# gr
# ==
# 1.
# Se
# pr
#
#
#
# ==
def
```

```
Python 3.4.2 Shell
Python 3.4.2 (default, Oct 19 2014, 17:55:38)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.54)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
1. podnaloga nima veljavne rešitve.
- Izraz je_prestopno(1900) vrne True namesto False.
>>>
```

Ln: 9 Col: 4

Ko datoteko poženemo, se **rešitev preveri in shrani**.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje
# gregorijanski koledar.
# =====@000272=
# 1. podnaloga
# Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto
# prestopno (po gregorijanskem koledarju). Zgled:
#
#     >>> je_prestopno(2004)
#     True
#     >>> je_prestopno(1900)
#     False
# =====

def je_prestopno(leto):
    return leto % 4 == 0 and leto % 100 != 0
```

Ln: 18 Col: 44

Ln: 9 Col: 4

Rešitev ustrezno **popravimo** in znova **poženemo**.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Python 3.4.2 Shell
# Python 3.4.2 (default, Oct 19 2014, 17:55:38)
# [GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.54)] on darwin
# Type "copyright", "credits" or "license()" for more information.
# >>> ===== RESTART =====
# >>>
# Shranjujem rešitve na strežnik... Rešitve so shranjene.
# 1. podnaloga nima veljavne rešitve.
#   - Izraz je_prestopno(1900) vrne True namesto False.
# >>> ===== RESTART =====
# >>>
# Shranjujem rešitve na strežnik... Rešitve so shranjene.
# 1. podnaloga nima veljavne rešitve.
#   - Izraz je_prestopno(2000) vrne False namesto True.
# >>> |
```

Ln: 14 Col: 4

Rešitev ustrezno **popravimo** in znova **poženemo**.

```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje
# gregorijanski koledar.
# =====@000272=
# 1. podnaloga
# Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto
# prestopno (po gregorijanskem koledarju). Zgled:
#
#     >>> je_prestopno(2004)
#     True
#     >>> je_prestopno(1900)
#     False
# =====

def je_prestopno(leto):
    return leto % 4 == 0 and leto % 100 != 0 or leto % 400 == 0
```

Ln: 1 Col: 38

Ln: 14 Col: 4

Postopek **ponavljamo, dokler naloge ne rešimo.**


```
datumi.py - /Users/matija/Downloads/Projekt Tomo/datumi.py (3.4.2)
# =====
# Datumi
#
# Ko
# gr
# ==
# 1.
# Se
# pr
#
#
#
#
# ==
def
>>> ===== RESTART =====
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
1. podnaloga nima veljavne rešitve.
- Izraz je_prestopno(1900) vrne True namesto False.
>>> ===== RESTART =====
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
1. podnaloga nima veljavne rešitve.
- Izraz je_prestopno(2000) vrne False namesto True.
>>> ===== RESTART =====
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
1. podnaloga ima veljavno rešitev.
>>>
```

Ln: 11 Col: 0

Postopek **ponavljamo, dokler naloge ne rešimo.**

Ustvari niz, izpiši niz

Tu je sklop osnovnih nalog za spoznavanje funkcij v Pythonu

1. podnalog

Vaša rešitev

Submission time: 23. oktober 2017. 09:01

```
def kopija(znak, n):  
    '''Vrne n kopij znaka'''  
    return n * znak
```



Uradna rešitev

```
def kopija(znak, n):  
    '''Vrne niz, ki ga sestavlja n kopij znaka'''  
    niz = znak * n  
    return niz
```

2. podnalog

Vaša rešitev

Submission time: 23. oktober 2017. 14:26

```
def crkam(n, znak):  
    '''Izpiše črko H določene velikosti'''  
    znaki = 2*n + 1  
    znak2 = n + 2  
    crkaH = znaki * znak2
```



Uradna rešitev

```
def crkam(n, znak):  
    '''Izpiše črko H'''  
    # zgornji del  
    vrstica = znak + ' ' * n + znak + '\n'  
    print(n * vrstica, end = '') # na  
    # srednji del  
    print(' ' * n, end = '\n')
```

Po pravilno rešitvi lahko rešitev primerjamo z uradno.



Pogojni stavek

Tukaj je nabor vaj iz pogojnih stavkov.

Datumi



Koledar, ki ga trenutno uporabljamo v krščanskem (zahodnem) svetu, se imenuje gregorijanski koledar.

1. podnaloga

Sestavite funkcijo `je_prestopno(leto)`, ki preveri, ali je dano leto prestopno (po gregorijanskem koledarju). Zgled:

```
>>> je_prestopno(2004)
True
>>> je_prestopno(1900)
False
```

Deljivost



1. podnaloga

Datumi



Deljivost



Bratska igra



Avtomobilska zavarovanja



Enačba



Hladilnik



Indeks telesne mase



Indiana Jones



Naloga je tako rešena, mi pa **izberemo novo...**



DEMO

PRIHAJA BRATRANEC



<http://pisek.acm.si>





Labirint I.

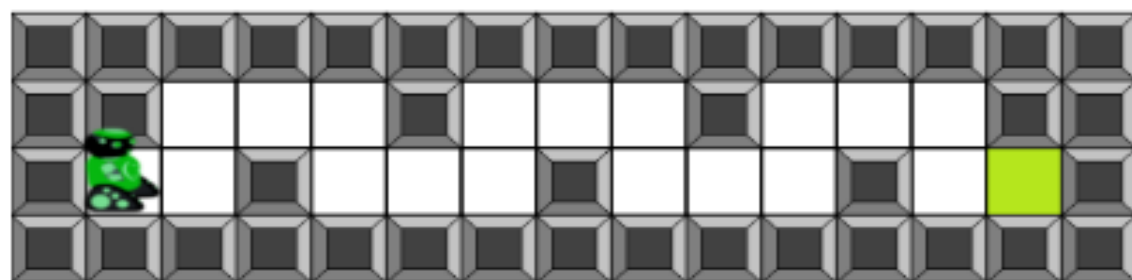
STATEMENT

Stopnja☆☆

Stopnja☆☆☆

Stopnja☆☆☆☆

Sprogramiraj robota, da bo prišel do zelenega območja na koncu poti.



|< |> > >> >>> >|

Oddaj program

12 kock izmed 16 imaš še na voljo.

premakni se desno

premakni se gor

premakni se dol

ponavlajaj 10 krat

izvedi

Program

premakni se desno

premakni se gor

ponavlajaj 10 krat

izvedi premakni se dol



Stopnja☆☆

Stopnja☆☆☆

Stopnja☆☆☆☆

Napiši navodilo, po katerem bo piščanček Pišek pobral zrno in ga odnesel v gnezdo.

Pod blok Program priprni navodila programa.

Navodilo se lahko ponovi večkrat z uporabo zanke ponavlja. To zanko uporabiš tako, da pod blok ponavlja tipkovnico vpišeš število ponovitev.

Oddaj program

34 kock izmed 40 imaš še na voljo.

premakni se desno

premakni se levo

premakni se gor

premakni se dol

poberi zrno

pospravi zrno

ponavlja 10 krat

izvedi

Program

ponavlja 10 krat

izvedi premakni se levo

ponavlja 10 krat

izvedi premakni se gor

poberi zrno

pospravi zrno

ponavlja 10 krat

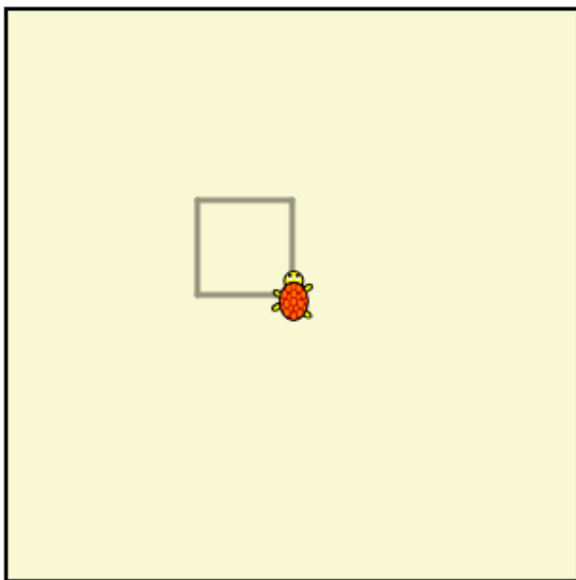
izvedi

Spoznaj Želvo

Tvoj program naj nariše sliko po vzoru sive črte.

S programom upravljaš premikanje želve. Z uporabo bloka *pojdi naprej za* želva nariše črto, katere dolžino določa blok *pojdi naprej za*. Želva se bo obrnila levo, če uporabiš blok *zavij levo*, in obrnila se bo desno, če boš uporabil blok *zavij desno*.

Ali lahko s svojim programom narišeš isti vzorec, ki je narisani s sivo črto?



Oddaj program

10 kock izmed 10 imaš še na voljo.

pojdi naprej za 1

Program

zavij levo ↶

zavij desno ↷

Začni znova

STATEMENT

Stopnja☆☆

Stopnja☆☆☆

Stopnja☆☆☆☆

Napiši program, ki prebere celo število in izpiše besedilo **DA**, če je število večje od 42, sicer pa naj ne izpiše nič.

Vhod: Izhod:

50 DA

Vhod: Izhod:

42

Test 1

Input:
50

Output:

2 kocki izmed 8 imaš še na voljo.

- Pisanje
- Branje
- Spremenljivke
- Besedilo
- Logika
- Matematika

```
Program
če
  preberi celo število > 42
izvedi
  izpiši " DA "
```



DEMO

