**Naslov naloge[[1]](#footnote-1)[[2]](#footnote-2),[[3]](#footnote-3): Čebelica nabira nektar**

1. **Tip naloge** (izberi ustreznega)**:**

☐ glede na način izvedbe:

☐ želva – risanje

X premikanje junaka

X manipulacija s predmeti (uporaba senzorjev)

☐ vhod-izhod - uporaba ukazov za branje, izpis

☐ glede na okolje:

☐ mreža (izberi ustrezno):

X premikanje junaka/figure z ukazi za absolutno[[4]](#footnote-4) naslavljanje

☐ premikanje junaka/figure z ukazi za relativno[[5]](#footnote-5) naslavljanje

☐ barvanje polj

☐ doseganje ciljnega polja[[6]](#footnote-6)

☐ pospravljanje predmetov

x zbiranje predmetov [[7]](#footnote-7)

☐ drugo: vpišite tukaj

☐ opis okolja: tukaj opišite okolje

1. **Kategorija** (označi vse primerne, a največ tri – glej tudi težavnost):

x 4.- 6. razred OŠ – začetniki ☐ 4.- 6. razred OŠ – napredni  
☐ 7.- 9. razred OŠ – začetniki ☐ 7.- 9. razred OŠ – napredni  
☐ SŠ – začetniki ☐ SŠ - napredni

1. **Vrsta naloge:**

☐ izdelava od začetka[[8]](#footnote-8)X dopolnjevanje/popravljanje[[9]](#footnote-9)  
☐ Parsonsov tip[[10]](#footnote-10)

1. **Programerski koncepti:**

☐ Zaporedje ukazov x Osnovna raba zank[[11]](#footnote-11)  
☐ Pogojni stavek x Osnovna raba senzorjev   
☐ Napredna raba zank[[12]](#footnote-12) ☐ Spremenljivke  
☐ Branje/izpis podatkov ☐ Manipulacija z nizi  
☐ Tabele ☐ Osnovna raba funkcij[[13]](#footnote-13)  
☐ Napredna raba funkcij/podprogramov[[14]](#footnote-14) ☐ Rekurzija

1. **Težavnosti naloge[[15]](#footnote-15)**

* **za kategorijo** 4.- 6. razred OŠ – začetnikiXlahka ☐ srednja ☐ težka
* **za kategorijo** 4.- 6. razred OŠ – napredni☐lahka ☐ srednja ☐ težka
* **za kategorijo** tukaj izberite kategorijo☐lahka ☐ srednja ☐ težka

1. **Zasnovana na** (če obstaja, URL naloge, na kateri temelji):

<https://studio.code.org/s/course2/stage/8/puzzle/1>

1. **Predlagatelj** (ime, priimek, e-pošta):

Matija.Lokar@fmf.uni-lj.si

**Vsebina**

1. **Besedilo naloge[[16]](#footnote-16)**

*Čebelica nabira nektar. Hoče obiskati vse tri rožice. Program je že napisan, a ima napake, oziroma mu manjkajo delčki. Popravi ga!*

1. **Priložena grafika [[17]](#footnote-17)**

☐ želva (ali kak drug lik)

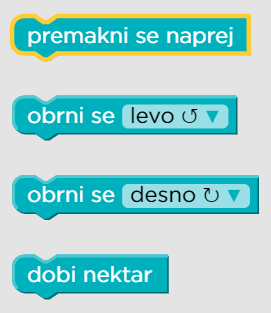
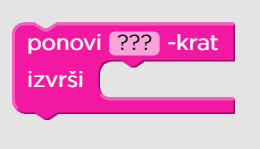
* želva (tu je lahko avto, svinčnik ....)
* ozadje

X mreža

* glej sliko za idejo. Like/figure je potrebno narisati na novo  
    
  

☐ vhod-izhod (grafika ni potrebna)

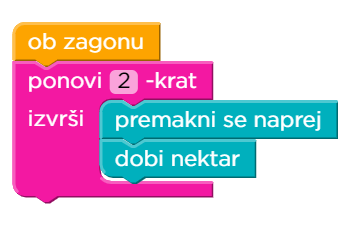
1. **Delčki (ukazi), ki so na voljo[[18]](#footnote-18)**

1. **Maksimalno dovoljeno število delčkov[[19]](#footnote-19):**

10 (dovolj jih je 6,a ...).

1. **Vnaprej podana koda[[20]](#footnote-20)**:



1. **Testni primeri[[21]](#footnote-21)**

☐ želva

* skica vzorca
* začetna točka želve
* input: eno celo število (neobvezno)

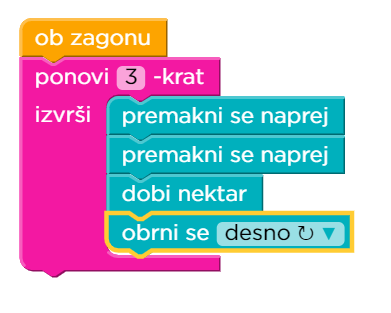
X mreža



☐ vhod-izhod

* izpis na vhodu (neobvezno)
* izpis na izhodu

1. **Rešitev**:



1. Zaželeno je, da je Obrazec čim natančneje izpolnjen. S tem bo olajšana tehnična izvedba naloge. Kljub temu pa smo veseli **poljubno dodelanih predlogov**. Če določenih podatkov ni (jih ne veš), pač označi! Obrazec je okvir in se ga ni potrebno (predvsem oblikovno) držati kot pijanec plota ... Predvsem je pomembna vsebina. Zaželeno pa je, da se držite navedenega vrstnega reda, ker bo tako lažje. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. Poimenuj nalogo, kot bo potem na Pišku [↑](#footnote-ref-3)
4. Gor/dol …, sever/jug/… (glede na mrežo) [↑](#footnote-ref-4)
5. Glede na smer lika (junaka) [↑](#footnote-ref-5)
6. Gre za eno polje v mreži, ki ga označimo kot cilj. [↑](#footnote-ref-6)
7. Imamo dva tipa predmetov; transportable (frnikole), collectible (zrna). Prve predmete lahko premikamo (običajno jih želimo pospraviti na označena polja (frnikole v luknjah)). Druge predmete pa junak samodejno pobere, če naleti na polje, na katerem je tak predmet (cilj je običajno, da ZBERE vse (zrna)). [↑](#footnote-ref-7)
8. V prostoru za skripto še ne obstaja koda [↑](#footnote-ref-8)
9. Del kode je že napisan v prostoru za skripto. [↑](#footnote-ref-9)
10. V prostoru za skripto so razmetani delčki [↑](#footnote-ref-10)
11. Ponovi N-krat [↑](#footnote-ref-11)
12. Dvojna zanka (gnezdenje zank), neskončna zanka ali zanka s sestavljenimi pogoji (iskanje izjeme, preverjanje različnih pogojev) [↑](#footnote-ref-12)
13. Brez parametrov. [↑](#footnote-ref-13)
14. Funkcije s parametri ali gnezdenje. [↑](#footnote-ref-14)
15. Uporabi kategorije, kot so izbrane pod točko 2 [↑](#footnote-ref-15)
16. Besedilo naj bo prilagojeno izbrani kategoriji. V primeru izbranih več kategorij priporočamo premislek – morda je bolje, da imamo dva (ali tri) predloge, vsakega za svojo kategorijo z nekoliko drugačno vsebino in likom. S tem se bolj prilagodimo miselnemu svetu reševalcev.   
    Zaželene so tudi slike v besedilu; prosimo tudi za oblikovanje - lokacija slike, odstavki ipd. [↑](#footnote-ref-16)
17. Izpolni za izbrani tip naloge. Zaželeno, da so priložene datoteke (pozor na avtorstvo – če nisi avtor grafike, napiši ustrezen Copyright (verjetno bomo potem sliko naredili na novo – zagotovo takrat, če CR ni naveden). Lahko je tudi samo opis grafike. [↑](#footnote-ref-17)
18. Navedeni morajo biti vsi delčki, ki so potrebni za rešitev, ter (morda) še kateri. Če so delčki razporejeni v kategorije, navedite te kategorije. Če naj bo določena kategorija polna (z vsemi ukazi kot so v kategoriji na <https://lusy.fri.uni-lj.si/ucbenik/prog/editor.html>), to označite [↑](#footnote-ref-18)
19. Če ni omejitve, napiši MAX ali ∞ [↑](#footnote-ref-19)
20. Glede na Vrsta naloge )točka 3 prvega lista) (za izdelava od začetka kode seveda ni) [↑](#footnote-ref-20)
21. Obvezno vsaj en testni primer, zaželeni so trije (če je smiselno), izpolni za izbrani tip naloge [↑](#footnote-ref-21)