

## »55. srečanje mladih raziskovalcev Slovenije«



### UJETI ŠTEFAN

INTERDISCIPLINARNO (MATEMATIKA IN RAČUNALNIŠTVO)

RAZISKOVALNA NALOGA

**Avtorice:** Lena Jagrič, Tineja Frangež, Gaja Bumbak

**Mentorica:** Jožica Špec, prof.

**Maribor, maj 2021**

## »55. srečanje mladih raziskovalcev Slovenije«



### UJETI ŠTEFAN

INTERDISCIPLINARNO (MATEMATIKA IN RAČUNALNIŠTVO)

RAZISKOVALNA NALOGA

**Avtorice:** Lena Jagrič, Tineja Frangež, Gaja Bumbak

**Mentorica:** Jožica Špec, prof.

**Maribor, maj 2021**

## KAZALO VSEBINE

POVZETEK .....	1
1 UVOD .....	2
2 GLAVNI DEL .....	4
2.1 Ujeti Štefan.....	4
2.2 Metoda poskušanja.....	5
3 SPORAZUMEVANJE S POMOČJO ROBOTA.....	6
4 UGOTOVITVE.....	17
5 ZAKLJUČEK.....	19
6 DRUŽBENA ODGOVORNOST.....	21
7 LITERATURA.....	22

### Kazalo slik

Slika 1: Robot v filmu Marsovec (Vir: <a href="https://www.youtube.com/watch?v=k-GH3mbvUro">https://www.youtube.com/watch?v=k-GH3mbvUro</a> , 2020.)..	10
Slika 2: Prikaz celotnega programa (Vir: Lastna programska koda.).....	11
Slika 3: Podroben prikaz prvega dela programa (Vir: Lastna programska koda.) .....	13
Slika 4: Podroben prikaz drugega dela programa (Vir: Lastna programska koda.) .....	14
Slika 5: Podroben prikaz tretjega dela programa (Vir: Lastna programska koda.) .....	15
Slika 6: Prikaz vstavljenega dela v program (Vir: Lastna programska koda.) .....	16
Slika 7: Prikaz sestavljenega robota (Vir: Lasten model robota.) .....	16

### Kazalo tabel

Tabela 1: Štefanovo sporočilo .....	4
Tabela 2: Štefanova koda .....	5
Tabela 3: Tabela črk in znakov .....	8

## **ZAHVALA**

Zahvaljujemo se svoji učiteljici, da nam je dala možnost reševanja naloge ujeti Štefan. Zahvale grede tudi našim prijateljem in staršem, saj so nas spodbujali ter močno podpirali pri našem delu. Hvala vsem, da ste verjeli v nas.

## POVZETEK

Jin-jang, dobro-zlo, prav-narobe, črno-belo, okužen-zdrav, 0-1 ... – nasprotja srečujemo vsak dan, v številnih življenjskih situacijah ter znotraj najrazličnejših ved, od logike, fizike, prava, medicine, arhitekture, matematike. Znani rek sicer pravi, da življenje ni samo črno-belo, a je razmišljanje na tak način pogosto edina prava pot do rešitve zelo zapletenega problema. V matematiki je uporaba sistema z zgolj dvema številoma (0-1) imenovana dvojiški sistem; v 17. stoletju ga je utemeljil Leibniz.

V svoji nalogi poskušamo najti rešitve za situacije s pomočjo prav takšnega načina razmišljanja. V poznani matematični uganki poskuša ujeti Štefan iz zaklenjene stavbe skozi okno s pomočjo prižiganja in ugašanja luči poklicati na pomoč. V filmu Marsovec poskuša astronaut komunicirati z ekipo na Zemlji preko omejenega števila signalov, natančneje 16. Prav zadnje nas posebej zanima. Uporabili bomo literaturo in raziskovali z eksperimentiranjem v sistemu LEGO MINDSTORMS. Preveriti želimo, kako bi bilo mogoče z dvojiškim sistemom rešiti astronauta z Marsa, ali pa gre za filmsko fikcijo. Cilj naloge je dokazati uporabnost matematike in računalništva v vsakdanjem življenju.

**Ključne besede:** nasprotja, dvojiški sistem, uporabnost matematike, uporabnost računalništva.

**Key words:** opposites, binary system, applicability of mathematics, applicability of computing.

## 1 UVOD

Kitajska zgodovina pripisuje zapis sistema polnih in črtkanih črtic kralju in filozofu Fuxiju, ki naj bi živel pred več kot 4000 leti. Pomen tega pa se je skozi stoletja zgubil. Uganko so razvozlati šele v 18. stoletju, ko so uvideli podobnost s številskim sistemom, sestavljenim iz le dveh znakov. Evropski mislec Leibniz je v svojem spisu poročal, da je poslal francoskemu menihu jezuitskega redu v Peking predstavitev svojega računskega sistema, ki sta ga sestavljali le dve števili: 0 in 1 (Leibniz 1703, str. 226–227).

Leta 1703 je opisal prednosti uporabe binarnega sistema za računanje, recimo, da se posamezniku ne bi bilo treba na pamet učiti niti seštevanja niti množenja. Če bi bila matematika tako enostavna, bi to danes razveselilo vsakega šolarja. Leibniz je navedel tudi dva primera praktične uporabe druge vrste številkega sistema, sestavljenega le iz enic in ničel, in sicer bi tehtničar lahko z manjšim številom uteži opravil tehtanje in tudi pri kovanju denarja, saj bi z manj kovanci izrazili ustrezne vrednosti (Leibniz 1703, str. 224–225).

Na pacifiškem otoku Mangareva so domorodci že mnogo let, preden je Leibniz opisal binarni sistem, uporabljali pri računanju sistem, ki je bil vsaj deloma podoben temu, pri čemer to ni edini tak primer, kar znanstveniki ugotavljajo tudi s pomočjo analiziranja besed in jezika na posameznih oddaljenih območjih (Bender in Beller 2014, str. 1322). Na primeru iz otoka Mangareva se vidi, da ker napisanih znakov za števila niso imeli, je bilo za menjavo dobrin in trgovanje potrebno dobro računanje na pamet, zato so razvili čim enostavnejši sistem računanja (Bender in Beller 2014, str. 1323).

Enostavnost razmišljanja v binarnem sistemu je njegova velika prednost. Tako enostavno razmišljanje je lahko ključ do rešitev mnogih zelo zapletenih vprašanj, kot bomo prikazali v nadaljevanju naloge. Najprej pa pojasnimo, kaj sploh je binarni številski sistem.

Dvojiški (binarni) številski sistem je sistem z osnovo 2. Edini števili, uporabljeni v tem sistemu, sta 0 in 1.

### PRIMER 1:

Število: 33

Najprej deliš število vedno z 2. Število deliš tolikokrat, dokler ne prideš do 0.

$$33 : 2 = 16, \text{ ost. } 1$$

$$16 : 2 = 8, \text{ ost. } 0$$

$$8 : 2 = 4, \text{ ost. } 0$$

$$4 : 2 = 2, \text{ ost. } 0$$

$$2 : 2 = 1, \text{ ost. } 0$$

$$1 : 2 = 0, \text{ ost. } 1$$

Številke sedaj preberemo od spodaj navzgor, tako dobimo število 100001.

DESETIŠKO: 33

BINARNO: 100001

PRIMER 2:

Število: 25

$$25 : 2 = 12, \text{ ost. } 1$$

$$12 : 2 = 6, \text{ ost. } 0$$

$$6 : 2 = 3, \text{ ost. } 0$$

$$3 : 2 = 1, \text{ ost. } 1$$

$$1 : 2 = 0, \text{ ost. } 1$$

DESETIŠKO: 25

BINARNO: 11001

Ker je dvojiški številski sistem enostavno realizirati z elektronskimi vezji, je uporabljen v praktično vseh računalnikih. V računalniku je uporabljen zato, ker vezja lahko ponazorijo signal samo z 1 (električni tok je) ali 0 (ni električnega toka). Predstavlja teoretično osnovo digitalnega računalnika.

V dvojiškem sistemu je mogoče zapisati poljubno desetiško celo število. V primerjavi z desetiškimi potrebujemo le malce več števk.

Opazimo lahko, da je vsaka naslednja utež dvakratnik prejšnje. Zato lahko preprosto ugotovimo, da imamo za zapis števil na voljo sicer samo dve števki, pri čemer njun položaj določa, ali so vrednosti 1, 2, 4, 8, 16, 32, 64, 128 ... vsebovane v številu.

Dvojiški sistem je izumil John Von Neumann. Rodil se je leta 1903, umrl pa je leta 1950. Nемеc Konrad Zuse je prvi zasnoval računalnik, ki je za delovanje uporabljal binarni sistem (Kožuh 2002).

Cilj naloge je pokazati, da si nekatere igre lahko razložimo (jih rešimo) s pomočjo matematičnih metod, ki nas spremljajo povsod v življenju. Metodologija, ki smo jo v nalogi uporabile, je metoda poskušanja in proučevanje pisnih virov ter eksperimentiranje v sistemu LEGO MINDSTORM.

## 2 GLAVNI DEL

### 2.1 Ujeti Štefan

Čas epidemije je bil čas, ko smo morali iskati različne načine zabave in druženja z družino. Tako smo v matematični reviji našle igro ujeti Štefan, ki je bila naš navdih za raziskovanje naprej. Vzбудila je našo radovednost in tako smo poleg želje po rešitvi ujetega Štefana želele tudi podrobneje raziskati dvojiški sistem.

Uganka je bila sledeča: Štefan je bil ujet v trgovini. Bilo je tik pred božičem in hotel je kupiti darila za starše in ni opazil, kdaj so vrata trgovine zaklenili. S pomočjo prižiganja in ugašanja luči na drevesu je želel pritegniti pozornost računalničarja. Naša naloga je razbrati Štefanovo sporočilo, ki temelji na dvojiškem sistemu (Harey 2017, str. 52–53).

Tabela 1: Štefanovo sporočilo

16	8	4	2	1
0	1	1	1	1
0	0	0	0	1
0	0	0	0	0
1	0	0	0	1
1	0	0	0	0
0	1	1	1	0
1	0	0	0	0



0	0	1	0	0
0	0	0	0	0
1	0	1	1	0
0	0	1	1	0
1	0	1	0	1
0	0	0	0	0
1	0	0	1	1
0	0	1	1	0
0	1	1	1	0

**Tabela 2: Štefanova koda**

1	2	3	4	5
A	B	C	Č	D
6	7	8	9	10
E	F	G	H	I
11	12	13	14	15
J	K	L	M	N
16	17	18	19	20
O	P	R	S	Š
21	22	23	24	25
T	U	V	Z	Ž

## 2.2 Metoda poskušanja

Ko smo se v šoli odločile za to nalogo, smo takoj poskušale ugotoviti, kaj pomeni razpredelnica na sliki in kako dvojiški sistem sploh deluje, a nam ni uspelo, nismo bile niti blizu; sprva smo pomislile, da moramo številke gledati navpično namesto vodoravno, da seštejemo vse enke in da so ničle med njimi presledki. Odločile smo se, da bomo, ko pridemo domov, pobrskale po spletu in si naslednji dan izmenjale vse informacije. Poskušale smo ugotoviti delovanje dvojiškega sistema, brez pomoči spleta, a nam ni uspelo, saj smo mislile, da se razpredelnica rešuje navpično od zgoraj navzdol. Začele smo iskati po spletu in nekaj časa nismo našle

ničesar, kar bi nam lahko bilo v pomoč, saj so bili povsod samo pretvorniki iz dvojiškega sistema v kakšen drug številski sistem, dokler nismo našle videoposnetka z razlago, kako se dvojiški sistem uporablja. Tako smo se ponovno lotile reševanja. Sedaj nam je bilo jasno, da moramo v razpredelnici številke gledati vodoravno. Po več poskusih reševanja smo spregledale enko in mislile, da je presledek ali pa smo napačno seštele in dobile napačno črko, a na koncu smo le ugotovile, kaj je Štefan hotel sporočiti računalničarju nasproti.

V tabeli 1 vidimo, kdaj je bila jelka prižgana (1) in kdaj ne (0). V prvi vrstici tabele so napisane potence števila dva. V tabeli 2 pa so prikazane vrednosti črk. Lotimo se prve vrstice:

Dvojiška števila se ne začnejo na 0. Dobimo število 1111, ki je, ko ga pretvorimo v naravno število, 15. Lahko pa uporabimo potence na vrhu tabele 1 (uporabili bi jih med pretvarjanjem, ampak zaradi daljšega zapisa je to pretvarjanje težje) in samo seštejemo  $8 + 4 + 2 + 1$  in tudi tako dobimo 15. Število 15 pripada črki N in s to črko se začne njegovo sporočilo. Ta postopek seštevanja potenc uporabljamo vse do konca. Kjer je zadnje število 1110, tako kot pri prvem primeru, seštejemo številke, pod katerimi je število 1 oziroma je jelka prižgana. Seštejemo  $8 + 4 + 2$  in dobimo število 14, kar v njegovi kodi predstavlja črko M.

OPOMBA: Tam, kjer v vrstici ni nobene 1, je presledek med besedami.

Na koncu pridemo do rešitve, in sicer: NA POMOČ, UJET SEM.

### **3 SPORAZUMEVANJE S POMOČJO ROBOTA**

Kot smo predstavile, je Štefan (Harej 2017, str. 52–53) v svoji zagati sporočila v zunanji svet pošiljal s pomočjo luči, ki jo je sam prižgal in ugašal. Vendar pa je tako svojega naslovnika dosegel le, ko je le-ta prižiganje in ugašanje luči videl skozi okno. Vsakokrat je luč tudi sam fizično prižgal oziroma jo je ugasnil. Uporabnost takšne rešitve je precej omejena, zato v nadaljevanju predstavljamo možni način sporazumevanja, ki bi bil širše uporaben.

Zgodba in istoimenski film Marsovec (Weir 2014, str. 1–512) prikazuje misijo Ares 3 iz vesoljskega programa Ares. Cilj tega programa je bil, da naj bi človeštvo osvojilo Mars, ko bi prvič poslali ljudi na drug planet in s tem razširili horizont človeštva (Weir 2014, str. 1). Misiji

Ares 1 in Ares 2 sta bili uspešni misiji, saj so astronauti dosegli cilje, uspešno prispeli na Mars in se vrnili na Zemljo, pri misiji Ares 3 pa se je zapletlo. Astronauti so prispeli na Mars in pri opravljanju nalog jih je zunaj postaje dohitela nevihta. Vsi razen enega astronauta so se vrnili na postajo, medtem ko enemu izmed njih to ni uspelo. Glede na vse okoliščine so domnevali, da je izgubil življenje in se napotili nazaj proti Zemlji. Vendar pa temu ni bilo tako, nesrečni astronaut je le omedlel in se zbudil ter po preučitvi svoje situacije z grozo ugotovil, da je ostal sam, ujet na planetu Mars (Weir 2014, str. 4–6). Njegovi občutki so takšni (Weir 2014, str. 7):

»Takšna je torej situacija. Nasedel sem na planetu Mars. Ne morem se sporazumevati z raketo Hermes niti z Zemljo. Vsi mislijo, da sem umrl. Nahajam se na postaji, ki je bila razvita tako, da deluje 31 dni.«

Brezupnost položaja je več kot očitna. Astronavt je zato s preostalo tehnično opremo, ki v neurju ni bila poškodovana, poskušal vzpostaviti nek sistem, da bi se vendarle lahko sporazumeval s postajo na Zemlji. Uporabil je šestnajstiški sistem, v katerem so sprogramirani tudi računalniki. Iz Zemlje so na sondo pošiljali signale v šestnajstiškem sistemu, kako naj preprogramira roverja, da se bodo lahko lažje sporazumevali. Palica na kameri sonde je kazala na table, na katerih so bili znaki šestnajstiškega sistema. Astronavt si je napisal znake, na katere je pokazala palica, in jih pretvoril v navodila. Tako je rover priključil na sondo in je sporazumevanje v šestnajstiškem sistemu potekalo že kar avtomatsko, saj mu ni bilo več treba ročno pretvarjati znakov v navodila. Astronavt je na tablo napisal odgovor in preko kamere na sondi so na Zemlji prejeli fotografijo (Weir 2014, str. 1–512).

Zastavlja se vprašanje, ali bi bilo moč v tako brezupnem položaju, kot je bil astronautov, sporazumevanje izvesti tudi v manj kompleksnem tehnološkem sistemu, in to s pomočjo dvojiškega sistema. Preizkus bomo opravili s pomočjo makete, le da bi sistem deloval v dvojiškem, ne pa v šestnajstiškem sistemu. Za izdelavo makete uporabimo LEGO MINDSTORM kocke. Prav tako tudi programiranje poteka v LEGO MINDSTORM programu.

Robot je sestavljen iz osnovne enote, motorja, kabla, ki povezuje motor z osnovno enoto, kock, ki predstavljajo ogrodje, okrasne cevi in ploščic, ki predstavljajo sončne celice.

Princip delovanja makete sledi cilju, da lahko na osnovni enoti izbereš črko ali znak ( \_ . ?), robot pa jo prikaže v dvojiškem sistemu. Črke so urejene po abecedi, na koncu pa so še znaki, vendar pa niso vključeni šumniki, to so č, š, ž. Vsaki črki oziroma znaku je pripisana številka, ki predstavlja njeno kodo. Robot številko pretvori v dvojiški sistem in rezultat poda uporabniku. To stori na podoben način kot v filmski sceni iz Marsovca. In sicer ima naš robot tri položaje, ki jih prikaže z obračanjem ter s paličico. Pokaže lahko na tablo s številko 0, na tablo s številko 1 ali pa se postavi v osnovni položaj (med tablama 0 in 1). Zadnji položaj služi uporabniku za sledenje prikazovanja posameznih zaporednih položajev. Na koncu prikaže celoten zapis v dvojiškem sistemu na ekranu ter številko, ki ga je pretvarjal.

Vzemimo sedaj primer črke A, ki ima določeno število dva. Robot najprej pretvori število dva v dvojiški sistem, in to je 10. Najprej pomakne paličico na tablo s številom 1, nato se postavi v vmesni položaj, na koncu pa še pokaže na tablo s številom 0. Na ekranu pokaže, katero število je pretvarjal, v našem primeru pokaže število 2, ter sproti, ko se paličica premika, prikaže zapis v dvojiškem sistemu. Ko pokaže s paličico na tablo s številom 1, tudi na ekranu zapiše to število.

**Tabela 3: Tabela črk in znakov**

ČRKE	ŠIFRA ČRKE	ŠTEVILO V DVOJIŠKEM SISTEMU	ŠTEVILO V DVOJIŠKEM SISTEMU S STALNO DOLŽINO PETIH ZNAKOV*
A	2	10	00010
B	3	11	00011
C	4	100	00100
D	5	101	00101
E	6	110	00110
F	7	111	00111
G	8	1000	01000
H	9	1001	01001
I	10	1010	01010

J	11	1011	01011
K	12	1100	01100
L	13	1101	01101
M	14	1110	01110
N	15	1111	01111
O	16	10000	10000
P	17	10001	10001
R	18	10010	10010
S	19	10011	10011
T	20	10100	10100
U	21	10101	10101
V	22	10110	10110
Z	23	10111	10111
_	24	11000	11000
?	25	11001	11001
.	26	11100	11100

*\*Opomba: Razlaga četrtega stolpca je nad sliko 6*

Zgoraj je prikazana tabela črk in znakov, s katerimi operira robot. Vsaka črka ima določeno število. V zadnjem stolpcu pa so števila iz prejšnjega stolpca napisana v dvojiškem sistemu. Lahko opazimo, da se vsa liha števila končajo na 1, vsa soda pa na 0. Opazimo lahko tudi, da ima vsak naslednik sodega števila enake številke razen zadnje, ki je 1. PRIMER: 10010 – sodo število, 10011 – liho število (naslednik).



**Slika 1: Robot v filmu Marsovec (Vir: <https://www.youtube.com/watch?v=k-GH3mbvUro>, 2020.)**

Programiranje je bilo izvedeno s pomočjo aplikacije LEGO® MINDSTORMS® EV3 Software (LEGO Group, DK-7190 Billund, Denmark. Copyright © 2013). Programska koda za upravljanje robota je razdeljena v tri sklope (glej sliko 2):

- prvi sklop, ki je namenjen inicializaciji robota in zagonu vmesnika za vnos oziroma izbor črk;
- drugi sklop, ki je namenjen pretvorbi izbrane črke v binarni zapis;
- tretji sklop, ki je namenjen vodenju robotske roke na osnovi binarnega zapisa ter koda za nadzor prekinitve programa.



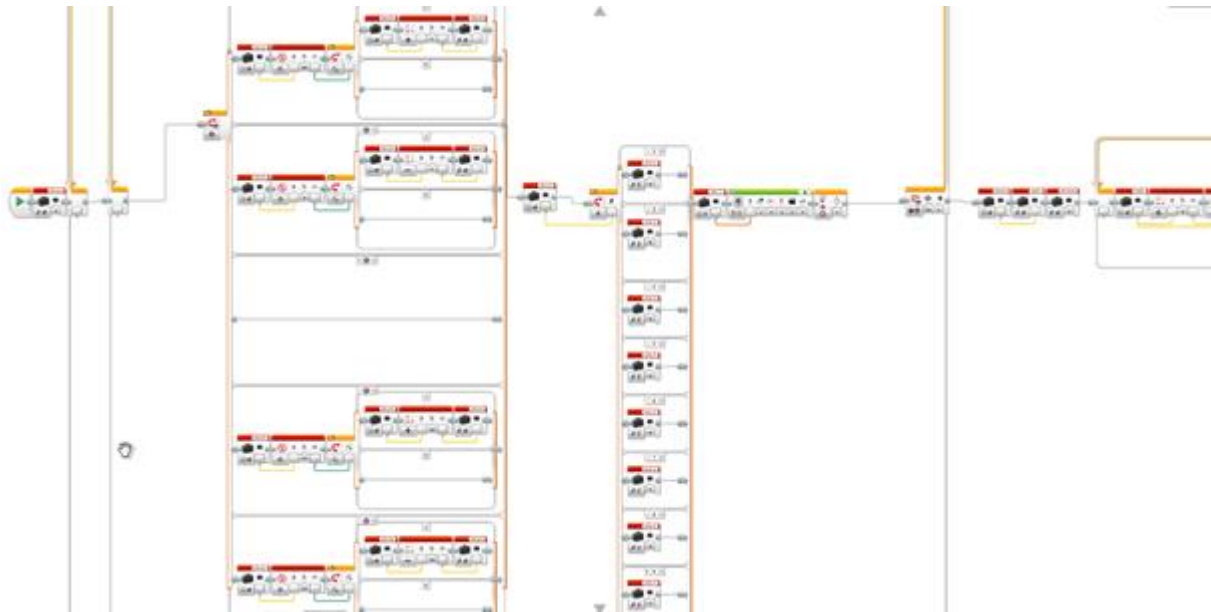
**Slika 2: Prikaz celotnega programa (Vir: Lastna programska koda.)**

Z vklopom kontrolne enote na robotu se naloži shranjena programska koda in se požene prvi sklop kode. Psevdo koda prvega sklopa je sledeča (glej sliko 3):

1. Zagon programa.
2. Oblikovanje nove spremenljivke »stevec«, ki ji določimo vrednost 2.
3. Zagon neskončne zanke, znotraj katere se izvajajo vsi trije sklopi kode.
4. Zagon zanke prvega sklopa, ki se izvaja, dokler uporabnik ne pritisne »sredinskega« gumba na kontrolni enoti.
5. Preverjanje stanja gumbov na kontrolni enoti:
  - a. Če je pritisnjen gumb »desni«, preberi vrednost »stevec«, preveri, ali je »stevec« manjši od 25:
    - i. Če je »stevec« manjši od 25, preberi »stevec«, povečaj »stevec« za 1, shrani novo vrednost v »stevec«.
    - ii. Ne spremeni vrednosti »stevec«.
  - b. Če je pritisnjen gumb »levi«, preberi vrednost »stevec«, preveri, ali je »stevec« večji od 3:

- i. Če je »stevec« večji od 3, preberi »stevec«, zmanjšaj »stevec« za 1, shrani novo vrednost v »stevec«.
    - ii. Ne spremeni vrednosti »stevec«.
  - c. Če je pritisnjen gumb »gor«, preberi vrednost »stevec«, preveri, ali je »stevec« manjši od 14:
    - i. Če je »stevec« manjši od 14, preberi »stevec«, povečaj »stevec« za 10, shrani novo vrednost v »stevec«.
    - ii. Ne spremeni vrednosti »stevec«.
  - d. Če je pritisnjen gumb »dol«, preberi vrednost »stevec«, preveri ali je »stevec« večji od 12:
    - i. Če je »stevec« večji od 12, preberi »stevec«, zmanjšaj »stevec« za 10, shrani novo vrednost v »stevec«.
    - ii. Ne spremeni vrednosti »stevec«.
  - e. Če ni pritisnjen gumb, ne stori ničesar.
- 6. Preberi novo vrednost »stevec«.
  - a. Če je vrednost »stevec« enaka 2, oblikuj novo spremenljivko »crka2« in ji dodeli vrednost »A«.
  - b. ...
  - c. Če je vrednost »stevec« enaka 23, oblikuj novo spremenljivko »crka2« in ji dodeli vrednost »Z«.
  - d. Če je vrednost »stevec« enaka 24, oblikuj novo spremenljivko »crka2« in ji dodeli vrednost »\_«.
  - e. Če je vrednost »stevec« enaka 25, oblikuj novo spremenljivko »crka2« in ji dodeli vrednost »?«.
  - f. Če je vrednost »stevec« enaka 23, oblikuj novo spremenljivko »crka2« in ji dodeli vrednost ».«.
- 7. Preberi vrednost »crka2«.
- 8. Izpiši vrednost »crka2« na ekranu kontrolne enote.
- 9. Počakaj 1 sekundo.
- 10. Preveri, ali je izpolnjen pogoj za končanje zanke prvega sklopa:
  - a. Če je pritisnjen »sredinski« gumb, zapiskaj, končaj zanko.
  - b. Če ni pritisnjen »sredinski« gumb, se vrni nazaj na začetek zanke.

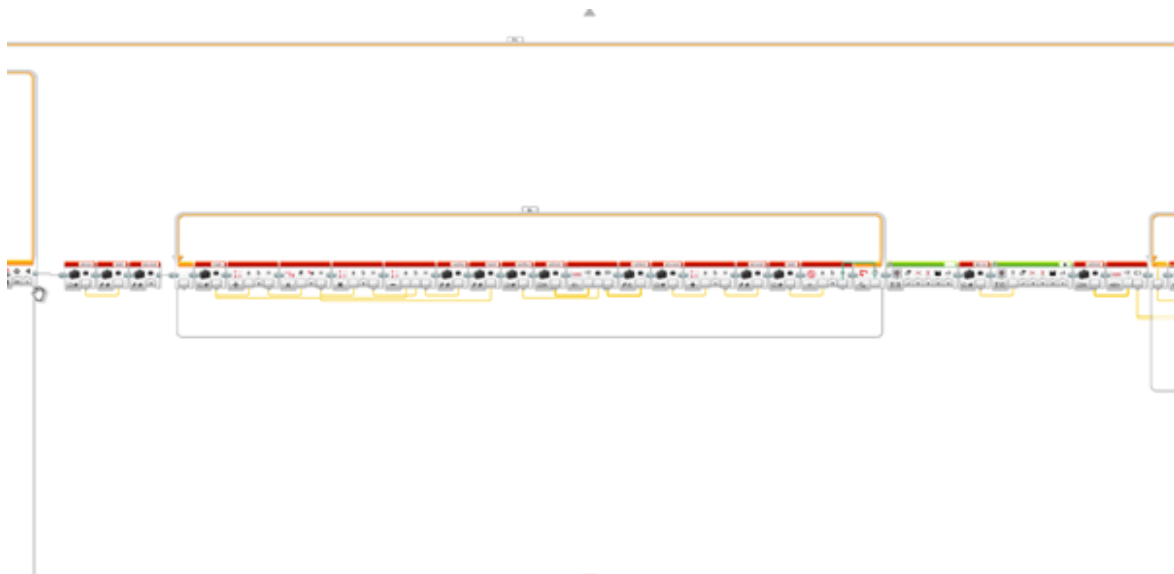




**Slika 3: Podroben prikaz prvega dela programa (Vir: Lastna programska koda.)**

Po izhodu iz zanke prvega sklopa se požene drugi sklop programa (glej sliko 4). Psevdo koda drugega sklopa je sledeča:

11. Preberi vrednost »stevec« in zapiši to vrednost v novo spremenljivko »test1«.
12. Oblikuj novo spremenljivko »stevec2« in ji dodeli vrednost 0.
13. Poženi zanko:
  - a. Deli vrednost »test1« z 2.
  - b. Rezultatu zbrisi decimalna mesta.
  - c. Izračunaj spremenljivko »razlika« med »test1« in  $(\text{»test1«} / 2 \text{ brez decimal}) * 2$ .
  - d. Shrani rezultat brez decimalnih mest v »test1«.
  - e. Oblikuj vektor »vektor1« in na novo prazno pozicijo zapiši vrednost spremenljivke »razlika«.
  - f. »stevec2« povečaj za 1.
  - g. Če je »test1« enak 0, zaključi zanko, sicer se vrni na začetek zanke.
14. Zbrisi ekran kontrolne enote.
15. Zapiši vrednost »stevec«.



Slika 4: Podroben prikaz drugega dela programa (Vir: Lastna programska koda.)

V tretjem sklopu kode se izračunan dvojiški zapis pretvori v ukaze za gibanje robotske roke (glej sliko 5). Psevdo koda tega sklopa je:

16. Preberi dolžino vektorja »vektor1«.

17. Poženi zanko in jo ponavljal tolikokrat, kot je dolžina vektorja »vektor1«.

a. Preberi vrednost »vektor1« na poziciji dolžina-števec zanke-1.

b. V vrstici števec zanke+1 zapiši vrednost iz »vektor1«.

c. Če:

i. Je vrednost »vektor1« enaka 1, zavrti motor robotske roke v levo za četrt obrata, počakaj 1 sekundo, zavrti motor robotske roke v desno za četrt obrata.

ii. Je vrednost »vektor1« enaka 0, zavrti motor robotske roke v desno za četrt obrata, počakaj 1 sekundo, zavrti motor robotske roke v levo za četrt obrata.

d. Povečaj števec zanke za ena.

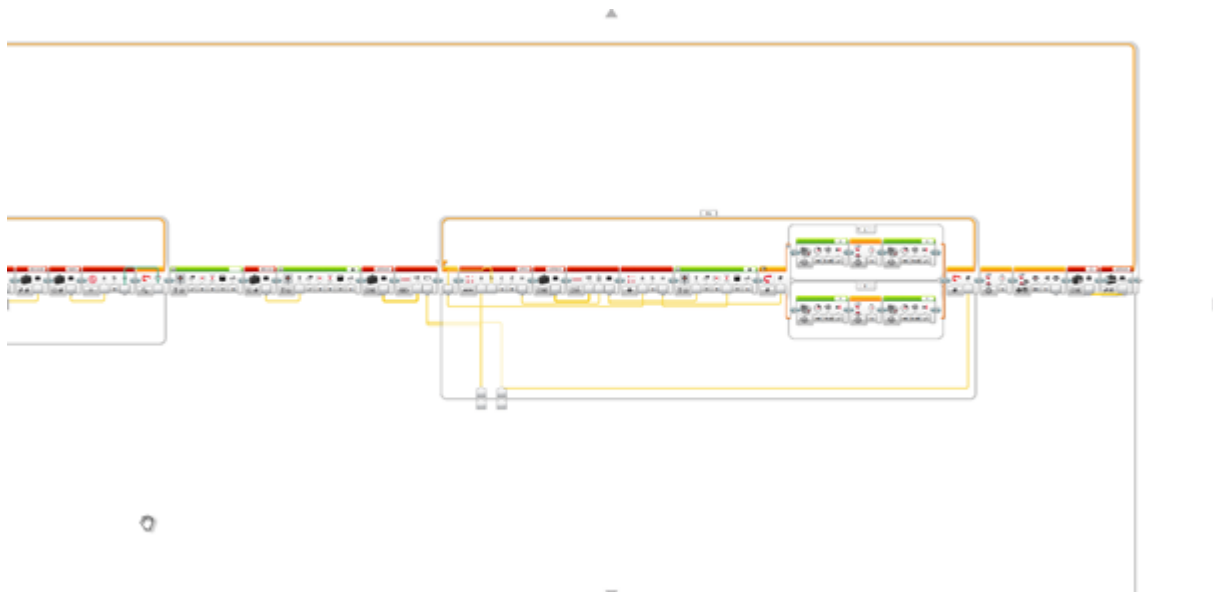
e. Če je števec zanke enak dolžini vektorja »vektor1«, končaj zanko, sicer ponovi.

18. Počakaj eno sekundo.

19. Počakaj na pritisk »sredinskega« gumba na kontrolni enoti.

20. Zbriši vektor »vektor1«.

21. Vrni se na začetek zanke programa.



**Slika 5: Podroben prikaz tretjega dela programa (Vir: Lastna programska koda.)**

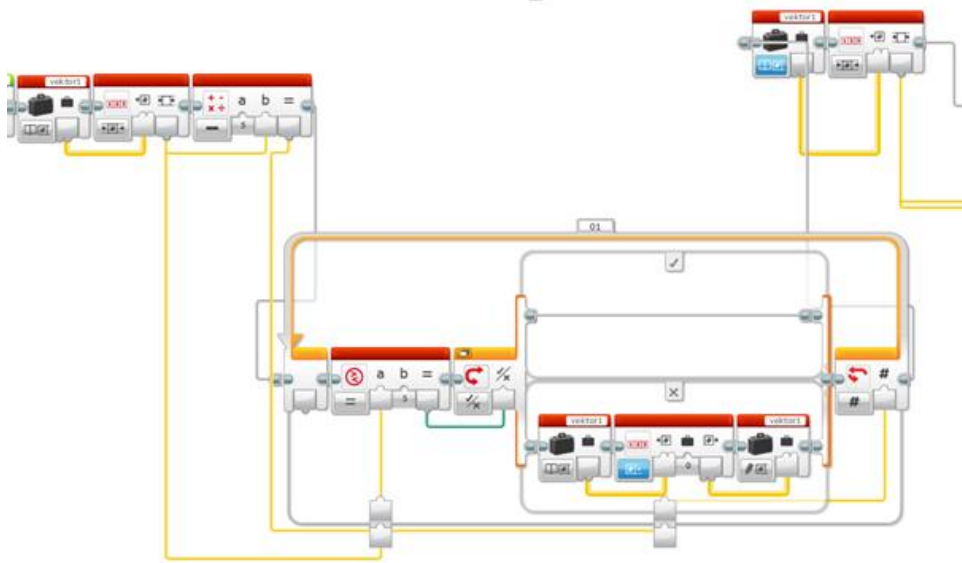
Ker dolžina binarnega zapisa varira (med 2 in 5), bi lahko za uporabnika bilo težko razločiti znake, če jih bi robot prejel več hkrati v zaporedju. Zato smo kodo dopolnili z dodatnim elementom, ki pri binarnem zapisu vedno zagotavlja dolžino pet. S tem uporabnik točno ve, kdaj se prične in konča posamezni znak. Dodatna koda je vrinjenja pred zadnji sklop, ki je namenjen upravljanju robotske roke. Psevdo koda tega vrinjenega sklopa je:

#####

1. Preberi dolžino vektorja "vektor1"
2. Izračunaj razliko: 5-dolžina
3. Ponavljaj zanko, dokler števec ni 0
  - a. Preveri, če je bila razlika večja od nič
    - i. Če "da", dodaj na koncu vektorja "vektor1" vrednost 0
    - ii. Če "ne", ne stori nič

#####

Dodane ničle v binarnem zapisu, njegove vsebine ne spremenijo, zagotavljajo pa, da ima vsak znak v tabeli 3 dolžino binarnega zapisa enako 5. Na sliki 6 je prikazana dodatna vrinejna koda.



Slika 6: Prikaz vstavljenega dela v program (Vir: Lastna programska koda.)

Na sliki 6 je prikazan sestavljen robot. Oblika robota sledi obliki v filmu. Pri dejanski uporabi robota v situaciji, kot je prikazana v filmu, bi bilo prvi sklop programske kode potrebno nadomestiti s kodo, ki bi omogočala radijski sprejem vrednosti spremenljivke »stevec«.



Slika 7: Prikaz sestavljenega robota (Vir: Lasten model robota.)

## 4 UGOTOVITVE

Števili 1 in 0 poskušamo razumeti s pomočjo narave in mnogih življenjskih situacij, kjer prav tako uporabimo le dve oznaki ali celo dve nasprotji, na primer da-ne, obstaja-ne obstaja, črno-belo, dan-noč, jin-jang, dobro-zlo. Ideja, da bi tudi matematične probleme reševali na tak način, in sicer s pomočjo uporabe le dveh števil, to sta 0 in 1, se zazdi zelo mikavna.

V raziskovalni nalogi smo se spoznale z dvojiškim številskim sistemom, ki je sestavljen iz 0 in 1. Na prvi pogled je preprostejši kot desetiški sistem, ki smo ga dobro poznale že pred začetkom svojega raziskovalnega dela. Pomislile smo: »Kako enostavna bi lahko bila matematika!«

Spoznale smo, kako se uporablja in bere ter kako spremeniš desetiško število v binarno število. Pri dvojiškem sistemu moraš poznati taktiko oziroma pomen. Dvojiški sistem je zelo priročen, saj smo ugotovile, kako lahko s signalom luči/svetilke osebi poveš, da si v nevarnosti oz. da potrebuješ pomoč. Ta sistem so najprej uporabljali v računalnikih oz. za računalnike kot računalniški jezik.

Pri spoznavanju pojavljanja in uporabe dvojiškega ter njemu podobnih sistemov smo ugotovili, da so se ti v zgodovini pojavili že zelo zgodaj, v različnih civilizacijah in na različnih delih sveta. Ugotovili smo, da so nekoč takšne sisteme uporabljali za več namenov, med drugim tudi trgovci in drugi pri računanju z velikimi števili. Tako smo ugotovili, da ima tak sistem veliko uporabno vrednost, ki je sprva sploh nismo poznali in to kljub svoji navidezni preprostosti.

Iz otroštva poznamo igro, v kateri si en igralec izbere poljubno npr. žival, drugi igralec pa to poskuša ugotoviti s postavljanjem vprašanj, na katera sme prvi igralec odgovarjati le z da-ne. Tako vidimo, da je za tudi precej zapleteno komuniciranje mogoče uporabiti (vsaj na strani enega igralca) zelo malo signalov, npr. le dva. Ko se spoznamo z matematično uganko "ujeti Štefan", ugotovimo, da lahko s pomočjo prižiganja in ugašanja luči, ki nadomeščata 0 in 1 iz dvojiškega sistema, posredujemo katero koli sporočilo. Z uporabo dveh signalov dobimo namreč dvojiški zapis katerega koli števila, ki pa ga ob uporabi kode prevedemo v črko. Uporabnost dvojiškega sistema odkrijemo v njegovi enostavnosti, saj potrebujemo zgolj dva signala.

Ob gledanju filma Marsovec smo videli zelo zapleteno situacijo, v kateri je poskušal astronaut komunicirati z ekipo na Zemlji preko sporočanja 16 signalov. Preveriti smo želeli, kako bi bilo

mogoče z dvojiškim sistemom rešiti astronavta z Marsa, ali pa gre za filmsko fikcijo. Ugotovili smo, da bi z uporabo robota lahko pošiljanje takih sporočil potekalo hitreje in bi tako astronaut res lahko poslal zelo kompleksna sporočila tudi zgolj z uporabo dvojiškega sistema.

Tako smo ugotovili, da ima dvojiški številski sistem veliko uporabno vrednost. Hkrati pa tudi, da je mogoče ob uporabi računalništva matematiko še enostavneje uporabiti v vsakdanjih življenjskih situacijah.

## 5 ZAKLJUČEK

V vsaki, na videz preprosti, igri se skriva veliko več strategij reševanja, kot si mislimo. Igra ujeti Štefan sodi med miselne igre, ki jo lahko za raziskovanje velikokrat uporabimo tudi pri pouku. Številski sistemi se skrivajo v ozadju hitrega razvoja tehnologije, ki nam na eni strani lajša življenje, hkrati pa v nas vzbuja strah, tako kot trenutna Covid situacija, s katero smo se malo poigrale:

Na Zemlji je brezskrbno in navadno življenje prekinil smrtonosni virus, ki se je razširil po vsem svetu in takrat se je življenje obrnilo za 180 stopinj. Zavladal je kaos. Izumili smo cepiva, zapirali razne ustanove, nosili maske in rokavice, si prali in razkuževali roke, a nič od tega ni pomagalo. Nismo imeli več upanja, zato smo v vesolje z radijskim signalom poslali: SOS. Nismo dobili odgovora. Pomisli smo, da je to zaradi tega, ker bitja iz vesolja ne govorijo našega jezika in potem smo se spomnili, da je matematika povsod enaka in nas bodo mogoče razumeli, če jim pošljemo sporočilo v dvojiškem sistemu. Zato so poslali signal v številkah: 10011 10000 10011 (SOS).

Ko so bitja iz vesolja prejela naš signal, so se vprašala, če smo dovolj marljivi, iznajdljivi, bistri in delavni, da nam pomagajo, zato so nam poslali računsko operacijo v dvojiškem sistemu, zraven pa so nam poslali 7 rezultatov, od katerih je bil pravilen le eden. Vsak rezultat je pomenil nekaj in mi smo morali ugotoviti, kaj nam Marsovci sporočajo z računsko operacijo, ki so nam jo poslali:

$$((110 \cdot 1010 + 101 \cdot (10 + 11) + 1101 \cdot 101) : (10 \cdot (101 + 10 + 11) + 101) + 1010 = ?$$

Rezultati, ki so nam jih poslali, so bili: 17, 22, 19, 16, 20, 15, 13. Vsak rezultat nekaj pomeni in ko izračunamo računsko operacijo in dobimo končni rezultat, ugotovimo, kaj nam sporočajo.

17: Nočemo vam pomagati, saj bi nam to vzelo preveč časa.

22: Ne zaupamo vam, ker ste uničili svoj planet; strah nas je, da boste še našega.

16: Pomagali vam bomo, ko vidimo, da ste rešili našo uganko.

20: Pomagali vam bomo samo v primeru, če nam prinesete nekaj dragocenega v zameno.

15: Mogoče, če se nam bo ljubilo.

13: Ne, mislimo, da nas hočete prevarati.

Ker smo na Zemlji res želeli in potrebovali njihovo pomoč, smo se odločili, da bomo poskusili rešiti njihovo uganko. Sprva smo imeli težave, a potem smo uspeli rešiti njihov račun in dobili rezultat 16, kar pomeni, da nam bodo pomagali!

Bili smo zelo veseli in hvaležni, da smo se odločili, da jih bomo v zameno za pomoč povabili na Zemljo, ko bo spet vse normalno.

Z enakim signalom smo tudi mi njim poslali računsko operacijo, da se malo pozabavajo:

$$1000 \cdot 111 - 10 \cdot (1001 - 101) : 100 - 1101 = ?$$

Poslali smo jim tudi 3 rezultate: 41, 46 in 39.

41: Ker smo tako hvaležni za vašo pomoč, smo se odločili, da lahko pridete na Zemljo, ko bo znova vse normalno.

46: Hvala vam za vašo pomoč, se vidimo.

39: Komaj čakamo, da vas spoznamo in se rešimo virusa. Hvala.

Prejeli so naše sporočilo in kmalu prišli po nas. V njihovi atmosferi je virus izumrl in Marsovci so prišli z nami na Zemljo. Začeli smo bolj in bolj komunicirati med seboj in vedno smo bili pripravljeni pomagati drug drugemu.

V nadgradnji raziskovalne naloge bi lahko poiskali najrazličnejše številske sisteme in raziskali njihovo delovanje, mogoče tudi kakšen »kodirni« sistem, ki ne bi temeljil na številih.



## 6 DRUŽBENA ODGOVORNOST

Vsaka igra ima svoj pomen. Pomeni igre se med sabo razlikujejo na več načinov. Nekatere pomene pri igri zaznamo takoj, drugih pa ne. To je odvisno od navodil, ki jih je avtor napisal. Avtor mora pomen igre poznati, saj bo le tako lahko napisal navodila. Če jih ni napisal jasno, bomo igro težje razumeli. Po navadi moramo igro večkrat igrati, da razumemo njen pomen in način igranja. Poznamo igre za različna starostna obdobja, zato je pomembno, kako izbiramo igre. Zato mlajši otroci niso sposobni odigrati igre, ki je primerna za starejše otroke. Pri igranju iger se krepijo človeški odnosi in razvoj možganov predvsem pri otrocih. Tudi naša igra ujeti Štefan je odlična za samostojno raziskovanje, krepitev logičnega mišljenja in sklepanja. Z njenim raziskovanjem razvijamo veliko matematičnih in drugih kompetenc: poznavanje, razumevanje, uporaba matematičnih pojmov in povezav med njimi ter izvajanje in uporaba postopkov, razumevanje in uporaba matematičnega jezika, uporaba in razumevanje delovanja informacijsko-komunikacijske tehnologije, navsezadnje pri vsaki taki družabni igri razvijamo svoje osebne kvalitete, kot so socialnost, samospoštovanje, medsebojne vrednote in obvladovanje čustev.

## 7 LITERATURA

- Weir, Andy (2014). *The Martian*. London: Del Rey Ebury Publishing.
- Bender, Andrea in Sieghard Beller (2014). Mangarevan invention of binary steps for easier calculation. *PNAS* January 28, 2014 111 (4) 1322–1327; <https://doi.org/10.1073/pnas.1309160110>.
- Leibniz (1703): Explanation of Binary Arithmetic WHICH USES ONLY THE CHARACTERS 0 AND 1, WITH SOME REMARKS ON ITS USEFULNESS, AND ON THE LIGHT IT THROWS ON THE ANCIENT CHINESE FIGURES OF FUXI (1703). Objavljeno v: *Die mathematische schriften von Gottfried Wilhelm Leibniz*, vol. VII C. I. Gerhardt (ed), pp 223–227, prevod iz francoščine, originalno besedilo objavljeno v *Memoires de l'Academie Royale des Sciences* ([www.leibniz-translations.com/binary.htm](http://www.leibniz-translations.com/binary.htm)).
- The Martian Hexadecimal Scene, <https://www.youtube.com/watch?v=k-GH3mbvUro>, 2020.
- Harej, Suzana (2017). Zakaj piškoti Leibniz? *Matematika v šoli* št.1, letnik 23, stran 52–53.
- Wikipedia (sl. [wikipedia.org](http://wikipedia.org)) pridobljeno: 20. 12. 2020; [https://sl.wikipedia.org/wiki/Dvoji%C5%A1ki\\_%C5%A1tevilski\\_sistem](https://sl.wikipedia.org/wiki/Dvoji%C5%A1ki_%C5%A1tevilski_sistem).
- LUSY ([lusy.fri.uni-lj.si](http://lusy.fri.uni-lj.si)) pridobljeno: 22. 12. 2020; <https://lusy.fri.uni-lj.si/ucbenik/book/1302/index4.html>.

- Kožuh, Rok (2002). Elektro-mehanski računalniki. SERŠ Maribor. E-gradiva. Pridobljeno: 9. 5. 2021; <http://www.s-sers.mb.edus.si/gradiva/w3/sistemi/elektromeh.html>.