Patricio Bulić

Dvojiški zapis in logika

Računalnik je elektronska naprava, ki zna računati in tako-ali-drugače obdelovati razne podatke. Celo ko v zimskih večerih v postelji, pokriti s toplo odejo, na prenosniku gledamo film, računalnik zelo intenzivno računa. Osnovni elektronski gradniki, tiste najmanjše elektronske komponente, ki jim pravimo tranzistorji, delujejo tako, da je na njihovih priključkih v katerem koli trenutku možno zaznati **le dve vrednosti električne napetosti**. No kakšen mikroelektronik in tehnolog integriranih silicijevih vezij bi temu ostro nasprotovala in trdila, da je sicer res, da tranzistorji delajo v nelinearnem območju, a so napetosti še vedno zvezne. Ampak oni živijo v svetu mikrovoltov, mikrowatov na kvadratni nanometer in podobo, kar mi niti ne opazimo in nas ne zanima – no verjetno vas res ne, mene pa vseeno “en mičkn firbec matra”, a je to stvar neke druge zgodbe. Taka je pač narava teh, nam čudnih, elektronskih vezij in jo, če v današnjem času vsaj delno želimo razumeti delovanje računalnikov, moramo sprejeti. In kaj za nas, navadne uporabnike, pomeni dejstvo, da lahko na nekem priključku izredno majhnega elektronskega elementa izmerimo le dve napetosti? To pomeni, da nam ta elektronski element na tem priključku sporoča, da je v nekem trenutku nekaj “res” ali pa “ni res”, ali pa da nečesa “ni” ali pa “je”, mogoče celo, da je nekaj “belo” ali “črno”. Torej, vsak tak, še tako majhen priključek, lahko hrani neko informacijo in se ta lahko spreminja. Največji možni količini informacije, ki jo lahko spravimo v nekaj, kar ima le dve možnosti (npr. da je nekaj “črno” ali “belo”, ali da je nekaj “res” ali “ni res”) pravimo ***bit***. In če se sedaj spravimo na “malo višji nivo” in se oddaljimo od fizike ali elektronike, pravimo, da “**vsak osnovni gradnik računalnika lahko hrani ali obdeluje en bit informacije**”. In seveda nam sedaj ne bi smelo biti tuje, če vam povem, da je tudi slika, ki jo hranite kot ozadje na zaslonu le mešanica velikega števila bitov. In da se med gledanjem vam ljubega filma, v računalniku pretaka in obdeluje ogromno število teh bitov. In (punce pozor!) da sta Julia Roberts in Hugh Grant v filmu Nothing Hill le množica bitov, ki je v danem trenutku nekje v računalniku postavljena in določena tako, da se Julia in Hugh na zaslonu strastno poljubljata. In (pozor fantje!) da so Messi, Neymar, Iniesta in Chavi, medtem ko na prenosniku gledate njihov tik-tak, le ogromna množica bitov! In medtem ko se vi zabavate, računalnik računa z biti. Čeprav se tega ne zavedate.

Dvojiški zapis

In se vrnimo nazaj k našim bitom. Računalnikarji, medtem ko računalnike programiramo, opisujemo kdaj in kako se bodo vsi ti bitki spreminjali. Zato si radi za ti dve stanji, določamo nekakšne oznake. Lahko bi na primer zapisali kot "črno" in "belo" mogoče celo "jabolko" in "hruška". A ker smo računalnikarji potomci (v sicer izjemno kratki evoluciji) matematikov (malo tudi elektrotehnikov, ki so matematikom priskočili na pomoč s tehnično izvedbo abstraktnih postopkov), so ti določili, da bomo ti dve vrednosti zapisali kot 0 in 1. Saj nam števke pravzaprav olajšajo percepcijo računanja (čeprav bi z malo več truda in zmede lahko računali tudi z jabolki in hruškami). V računalništvu največkrat govorimo, da ima nek ***bit vrednost 0 ali 1***.

Ker je v računalniku v enem od njegovih osnovnih gradnikov mogoče hraniti in obdelovati le bite, potem mora biti vse, kar hranimo v računalniku in kar računalnik obdeluje zapisano z ničlami in enicami (ne bi pa bilo prav do fantov, ki se za nas trudijo in živijo v svetu mikrowattov, in nanometrov, da pozabimo, da sta 0 in 1 le abstrakcija dejanske napetosti na posameznem priključku). Pa si najprej za začetek poglejmo, kako zapišemo števila (kako zapisujemo oz. kodiramo informacijo boste lahko prebrali v drugem delu tega zvezka).

Spomnimo se še naše osnovne šole in poglejmo, kako smo se učili ugotavljati vrednosti desetiških števil. Predpostavimo število 7043. Vem, da ste že "na prvi pogled" ugotovili vrednost tega števila (prvič ker vidite in ker so naši možgani izjemen procesni stroj, ki je zmožen procesirati vizualne podatke). Ampak računalnik načeloma ne vidi (čeprav se na vsakem izpitu najde kakšen študent, ki me prepričuje v obratno). In ne zna neposredno zapisati in interpretirati desetiške informacije. Vrednost številu 7043 določimo takole:

7 tisočic + 0 stotic + 4 desetice + 3 enice

Če zapišemo na malo bolj formalen način, dobimo:

$$7*10^3 + 0*10^2 + 4*10^1 + 3*10^0$$

Govorimo, da je v *desetiškem* sistemu osnova *deset* (10) in vsaka števka ima lahko eno izmed *desetih* vrednosti. Pozicija oz. mesto števke v zapisu določa njeno utež. Tako ima števka na mestu 0 (skrajno desna števka) utež ena (10^0) in števka na mestu 3 (tisočica) utež 1000 (10^3). Ali bi lahko hranili desetiška števila v računalniku? **Ne neposredno**! Zakaj? Ker smo rekli, da ima posamezen priključek osnovnih gradnikov le dve možni vrednosti in ne deset! Zato moramo števila v računalniku ***zapisovati dvojiško***. V *dvojiškem sistemu* bo osnova *dve* (2), vsaka števka pa bi imela le *dve* možni vrednosti (0 in 1). In bo pozicija števke prav tako določala njeno utež (a razen enice, ne bomo mogli govoriti o deseticah, stoticah, ...). Pa si poglejmo kar na primeru in skušajmo ugotoviti vrednost dvojiškega števila 1101110000011 (pa poskusite sedaj "videti" vrednost, ha!):

$$1*2^{12} + 1*2^{11} + 0^{*10} + 1*2^9 + 1*2^8 + 1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$

Huh, dolga, a ne? Pa dajmo to sešteti. Pred tem si poglejmo kakšne so uteži v dvojiškem sistemu:

$2^{12} = 4096$	$2^{8} = 256$	$2^{4} = 16$	$2^{1} = 0$
$2^{11} = 2048$	$2^{7} = 128$	$2^{3} = 8$	
$2^{10} = 1024$	$2^{6} = 64$	$2^{2} = 4$	
$2^{9} = 512$	$2^{5} = 32$	$2^{1} = 2$	

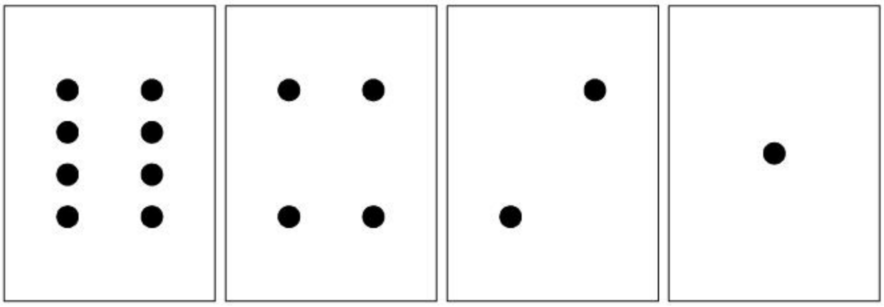
Seštejemo ustrezne uteži 4096+2048+512+256+128+2+1 = 7043! No super, nismo dokazali, a mi lahko verjamete, da je v dvojiškem sistemu možno zapisati poljubno desetiško celo število. Le malo več števk (=bitov) potrebujemo!

Opazimo lahko, da je vsaka naslednja utež dvakratnik prejšnje. Zato lahko preprosto otrokom rečemo, da imajo za zapis števil na voljo sicer dva bita, a njihova pozicija določa vrednosti 1, 2, 4, 8, 16, 32, 128, ...

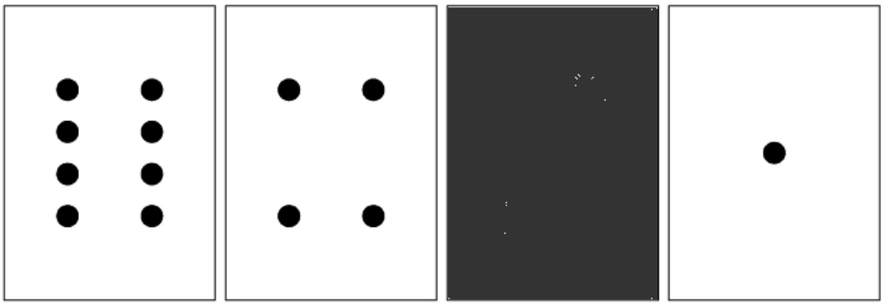
Kaj pa pretvorba iz desetiškega v dvojiško? Vidra pravi, da imajo otroci svoj način razmišljanja. Otrok ve, da ima na razpolago števila 1, 2, 4, 8, 16, 32,... Med njimi poišče največje število, ki je še manjše ali enako številu, ki ga pretvarja. Če bi npr. želeli pretvoriti število 134 v dvojiško, bi ugotovili, da je največja dvojiška utež, ki je manjša od 134 enaka 128. Zato si zapiše 1 in 128 odšteje od 134 ter dobi 6. Sedaj poskuša naprej po vrsti: najprej preveri, ali je 64 manjše od 6. Ker ni, dopiše tisti prejšnji enici eno ničlo in dobi 10. Nato poskuša z 32. Ker tudi ta ni manjši od 6 dopiše še eno ničlo in dobi 100. Nato poskusi s 16. Ker tudi ta ni manjši od 6, dopiše še eno ničlo in dobi 1000. Nato poskusi z 8. Ker tudi ta ni manjši od 6, dopiše še eno ničlo in dobi 10000. Nato poskusi s 4. Ker je ta manjši od 6 dopiše eno enko in dobi 100001. Potem 4 odšteje od 6, dobi 2 ter poskuša naprej z utežjo 2. Ker je ta enaka 2 pripiše še eno enico ter dobi 1000011. Nato odšteje 2-2 in dobi 0. Zato na konec doda še zadnjo ničlo. Sedaj je postopek končan. Število 134, zapisano dvojiško, je torej 10000110. Tudi ostala števila (ulomljena) in neštevila (črke, barve, ...) se da zapisati z biti.

Dvojiški zapis - Bevri

Ozadje naloge je dvojiški zapis števil. Če vrednosti kovancev zapišemo od največje proti najmanjši, bomo dobili zaporedje vrednosti, kot ga imajo karte z Vidre:

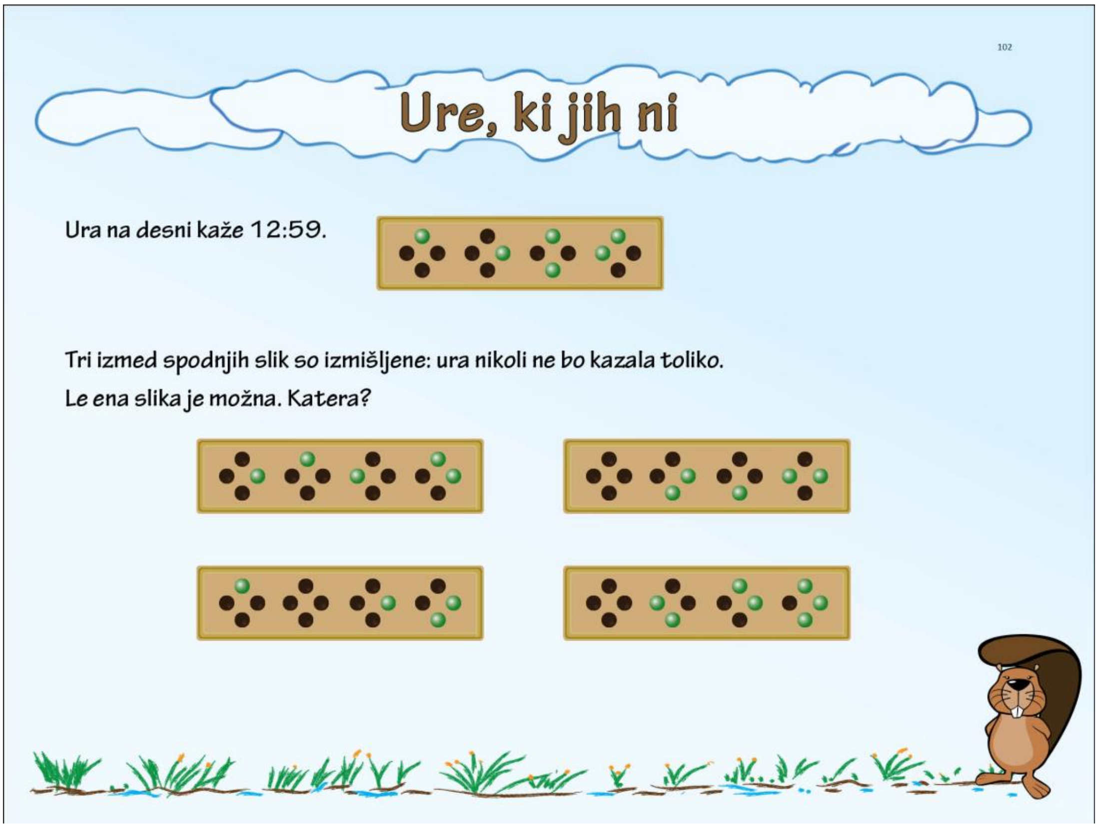


S temi kartami moramo zapisati vrednost 13. Ker karte prikazujejo vrednosti posameznega bita v 4-bitnem dvojiškem zapisu, lahko rečemo, da iščemo takšno postavitev bitov, ki nam bo dala vrednost 13. Vemo, da so vrednosti v dvojiškem zapisu določene enoznačno, torej obstaja le ena možnost zapisovanja vrednosti 13 s štirimi biti v dvojiškem zapisu. Vprašanje naloge lahko prevedemo na naslednje vprašanje: kakšno je najmanjše število bitov v 4-bitnem dvojiškem zapisu potrebnih za zapis vrednosti 13? Odgovor na to vprašanje dobimo s preprosto pretvorbo desetiške vrednosti 13 v dvojiški zapis. Spomnimo se, kaj pravi Vidra: otrok ve, da ima na razpolago števila 8, 4, 2 in 1. Med njimi bo poiskal največje število (kovanec), ki je enako ali manjše od števila, ki ga pretvarja. V našem primeru bo izbral število 8. Ker mu do 13 manjka še 5, bo izmed števil (kovancev) poiskal največje, ki je enako ali manjše od števila 5. V našem primeru bo to 4. Do 5 mu ostane le še 1, zato bo vzel kovanec z vrednostjo 1. Torej, za 13 bevrov bo potreboval kovanec za 8 bevrov, kovanec za 4 bevre, nič kovancev za 2 bevra ter en kovanec za 1 bevro. Če to izbiro ponazorimo z Vidrinimi kartami imamo

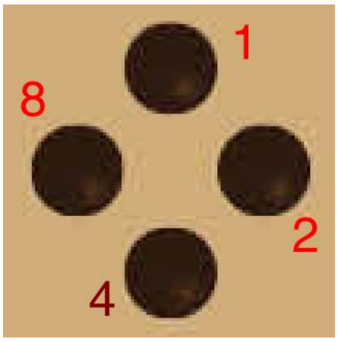


V dvojiškem zapisu je to 1101. Odgovor na vprašanje iz naloge se glasi: 3 kovance.

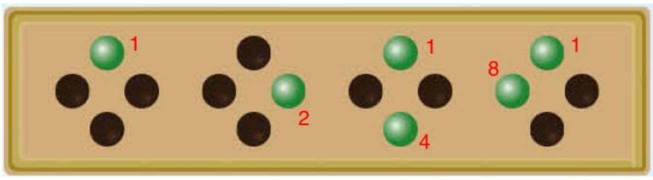
Binarni zapis – ure



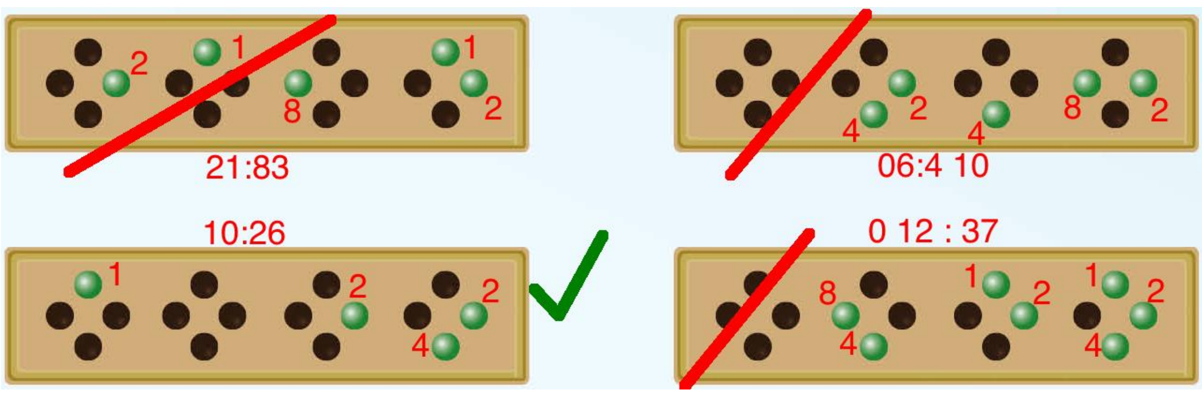
Naloga spet skriva binarni zapis števil. Vsako števko (0...9), ki tvori zapis v uri, je možno zapisati z binarnim zapisom s štirimi biti. In ravno to skriva zgornja naloga, pri čemer je vsak posamezen bit označen s črnim krogcem. Pa si poglejmo:



Z zgornjim krogcem kodiramo bit s težo 1, z desnim krogcem bit s težo 2, s spodnjim krogcem bit s težo 4 ter z levim krogcem bit s težo 8. Zeleno obarvan krogec označuje bit z vrednostjo 1 in črno obarvan krogec bit z vrednostjo 0. Tako je ura 12:59 res zapisana s spodnjo kombinacijo krogcev,

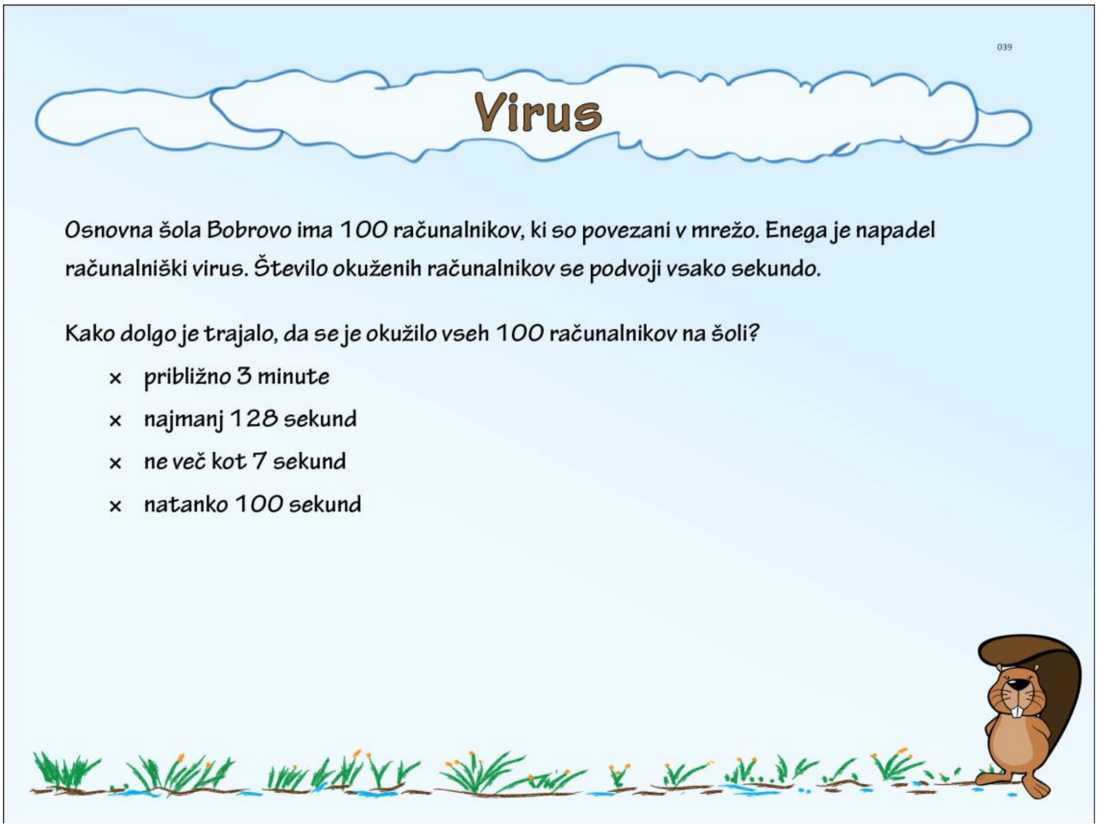


saj je v skrajno levem četverčku krogcev z zeleno obarvan krogec na mestu bita s težo 1, tj. ti štirje krogci zapisujejo bitno kombinacijo 0001. V naslednjem četverčku krogcev je zeleno obarvan krogec s težo 2 in je bitna kombinacija, ki jo zapisujejo ti štirje krogci 0010. V naslednjem četverčku sta zeleno obarvana krogca s težo 1 in 4 in je pripadajoča bitna kombinacija 0101. Zadnji četverček pa zapisuje bitno kombinacijo 1001, saj sta zeleno obarvana krogca s težo 8 in 1. Poglejmo si, kakšne vrednosti ponujajo štirje ponujeni zapisi iz naloge:



Možna je slika spodaj levo, na kateri je zapisana ura 10:26.

Virusi in eksponentna rast



Čeprav naloga omenja nekakšen hud virus, je ozadje naloge eksponentna rast. A je naloga po svoje zanimiva tudi s stališča binarnega kodiranja, saj si lahko z njim pomagamo do rešitve. Ker se število okužb vsako sekundo podvoji, nas to dejstvo močno spomni na to, da se teža vsakega bita v binarnem zapisu podvoji, če gremo iz desne proti levi. Spomnimo se, da so teže teh bitov: 1, 2, 4, 8, 16, 32, 64, 128,... Nalogo bi torej lahko prevedli na naslednje vprašanje: kako daleč se od bita s težo 1 nahaja bit s težo vsaj 100?

1 -> 2 -> 4 -> 8 -> 16 -> 32 -> 64 -> 128

Ko se okuži prvi, se bosta po eni sekundi okužila 2, po dveh sekundah 4, po treh sekundah 8, po štirih sekundah bo okuženih že 16, po petih sekundah bo okuženih 32, po šestih sekundah 64 računalnikov in po 7 sekundah pa že 128.

Nalogo bi v luči binarnega kodiranja lahko rešili tudi takole: Na kateri poziciji se nahaja bit s težo vsaj 100? Spomnimo se, da se bit z najmanjšo težo (skrajno desno) nahaja na ničtem mestu v binarnem zapisu, saj je njegova teža $1 = 2^0$, teža bita na prvem mestu je $2 = 2^1$, teža bita na drugem mestu je $4 = 2^2$, teža bita na tretjem mestu je $8 = 2^3$, teža bita na četrtem mestu je $16 = 2^4$, teža bita na petem mestu je $32 = 2^5$, teža bita na šestem mestu je $64 = 2^6$ in teža bita na sedmem mestu je $128 = 2^7$. Potrebnih je torej ne več kot 7 sekund, da okužimo 100 računalnikov.

Alja lovi ravnotežje

Naloga se glasi takole:

Poglej, s čim se Alja ukvarja že celo jutro! Poredna majhna bobrovka je šla na podstrešje iskat skrite zaklade in našla staro tehtnico in škatlo s sedmimi utežmi, ki tehtajo 1, 2, 4, 8, 16, 32 in 64 kilograma. Zdaj poskuša postaviti uteži na obe strani tehtnice tako, da bo uravnotežena. Obupala je že nad tem, da bi uporabila vse uteži, in bi bila vesela že, če bi ji uspelo tehtnico uravnotežiti vsaj z nekaj utežmi.

Kateri namig bi ji lahko pomagal?

a) Uteži za 4 kg in 16 kg morata biti na nasprotnih straneh tehtnice.

b) Uteži za 4 kg in 16 kg morata biti na isti strani tehtnice.

c) Morala bi uporabiti bodisi utež za 4 kg bodisi utež za 16 kg; ti dve uteži ne moreta biti uporabljani hkrati.

d) Tehtnice ni mogoče uravnotežiti s temi utežmi.

Opazimo, da bi se dalo posamezne uteži zapisati z dvojiškimi števili, ki vsebujejo le eno enico. Vsako binarno število pa lahko zapišemo **le na en način**. Npr. dvojiško število z vrednostjo 64 lahko zapišemo le kot 100 0000. Kaj to pomeni? Če želimo na eni strani imeti utež, ki tehta 64 kilogramov, bi morali na drugi strani nabrati uteži za 64 kilogramov. Če to obrnemo na dvojiška števila, bi z ustreznim seštevkom preostalih števil morali dobiti vrednost 64. A če seštejemo vsa preostala števila, dobimo 63, oz. dvojiško 111111. Povedano drugače: ker imamo na voljo le po en bit (1, 2, .. 64) na različnih binarnih mestih, in za zapis enega števila porabimo enega ali več teh bitov, s preostalimi ne moremo zapisati istega števila, saj bi to pomenilo, da se eno število, da zapisati na več različnih načinov.

Otrok bi morebiti razmišljal takole: če na eno stran dam utež s težo 64 kg, mi na drugi strani ostane za 63 kg uteži. Torej vseh uteži zagotovo ne morem uporabiti. Potem bi lahko utež za 64 kg dal stran in poskusil z utežjo za 32 kg, a bi ugotovil, da mu na drugi strani ostane za 31 kg uteži. Tako bi s postopkom eliminacije prišel do pravilnega odgovora.

Pravilen odgovor je torej odgovor D.

Alja lovi ravnotežje 2

Podobna naloga se glasi takole:

Poglej, s čim se Alja ukvarja že celo jutro! Poredna majhna bobrovka je šla na podstrešje iskat skrite zaklade in našla staro tehtnico in škatlo s sedmimi utežmi, ki 1/8, 1/4, 1/2, 1, 2, 4 in 8 kilogramov. Zdaj poskuša postaviti uteži na obe strani tehtnice tako, da bo uravnotežena. Obupala je že nad tem, da bi uporabila vse uteži, in bi bila vesela že, če bi ji uspelo tehtnico uravnotežiti vsaj z nekaj utežmi.

Kateri namig bi ji lahko pomagal?

a) Uteži za 1/2 in 2 kg morata biti na nasprotnih straneh tehtnice.

b) Uteži za 1/2 in 2 kg morata biti na isti strani tehtnice.

c) Morala bi izpustiti bodisi utež za 1/2 kg bodisi utež za 2 kg; ti dve uteži ne moreta biti uporabljani hkrati. Enako velja za ostale pare.

d) Tehtnice ni mogoče uravnotežiti s temi utežmi.

Rešitev je enaka, kot pri zgornji nalogi. Če nam je težko razmišljati z ulomki, si pomagamo tako, da vse vrednosti uteži pomnožimo z 8.

Bobrov merski sistem

Bobrčki ne merijo dolžin v metrih. Namesto tega, uporabljajo osem merilnih palic, ki jih preprosto imenujejo A, B, C, D, E, F, G in H. Palica A je najdaljša, dolžina palice B je ena polovica dolžine palice A, dolžina palice C je ena polovica dolžine palice B in tako naprej.

Ko ljudje merimo nekaj, izražamo dolžino v metrih, na primer: 7 m in 10 cm. Bobri, ki so manj natančni, merijo dolžino s kombinacijo palic, pri čemer se lahko vsaka palica pojavlja največ enkrat. Dolžina je tako lahko, na primer A + C + D, kar preprosto zapišejo ACD. Ali recimo B + C + G, kar preprosto zapišejo kot BCG.

Bobrčki Marija, Ana, Janez in David so izmeril velikosti svojih čolničkov:

Marija: BG

Ana: BF

Janez: CEFGH

David: CDEH

Kakšen je vrstni red čolničkov, če jih razvrstimo od najdaljšega do najkrajšega?

A) David, Ana, Marija, Janez

B) Janez, David, Marija, Ana

C) Ana, Marija, David, Janez

D) Janez, David, Marija, Ana

Ozadje naloge je spet dvojiški zapis. Spomnimo se, da je utež vsakega bita pol manjša od uteži njegovega levega soseda in enkrat večja od uteži njegovega desnega soseda. Zaporedje palic ABCDEFGH lahko predstavimo dvojiško z osmimi biti. Pri tem nam npr. bit z vrednostjo 0 pomeni, da pripadajoče palice ni, bit z vrednostjo 1 pa da pripadajoča palica je.

Tako bi na primer dolžino ACD dvojiško z 8 biti zapisali kot 10110000, saj je A = 1 (torej je prisotna), B = 0, C = 1, D = 1, E = 0, F = 0, G = 0 in H = 0.

Dolžino BCG pa dvojiško zapišemo kot 01100010.

Za rešitev naloge moramo torej dvojiško zapisati vse štiri izmerjene dolžine in jih primerjati med seboj. Lotimo se najprej dvojiškega zapisa:

Marija: BG = 01000010

Ana: BF = 01000100

Janez: CEFGH = 00101111

David: CDEH = 00111001

Kako bi sedaj te vrednosti primerjali in uredili po velikosti? Poglejmo si tri možne načine.

Prvi je, da za vsako od zapisanih dvojiških števil ugotovimo desetiško vrednost. Tako je dolžina Marijinega čolna 66 (64 + 2), dolžina Aninega čolna je 68 (64 + 4), dolžina Janezovega čolna je 47 (32 + 8 + 4 + 2 + 1) ter dolžina Davidovega čolna 57 (32+16+8+1).

Izmed dveh dvojiških števil je večje tisto, ki ima najbolj levi bit pri katerem se števili razlikujeta enak 1. Poglejmo si na primeru števil 01000010 (66) in 01000100 (68). Najbolj levi bit pri katerem se števili razlikujeta je bit na mestu 2, tj. bit s težo 4: 01000**0**10 in 01000**1**00. Zato je večje drugo število. Primerjajmo na enak način še števili 00101111 in 00111001. Razlikujeta se pri bitu s težo 16 tj. 001**0**1111 < 001**1**1001. Razvrstimo sedaj na ta način binarna števila, ki kažejo velikosti čolnov:

01000100 > 01000010 > 00111001 > 00101111

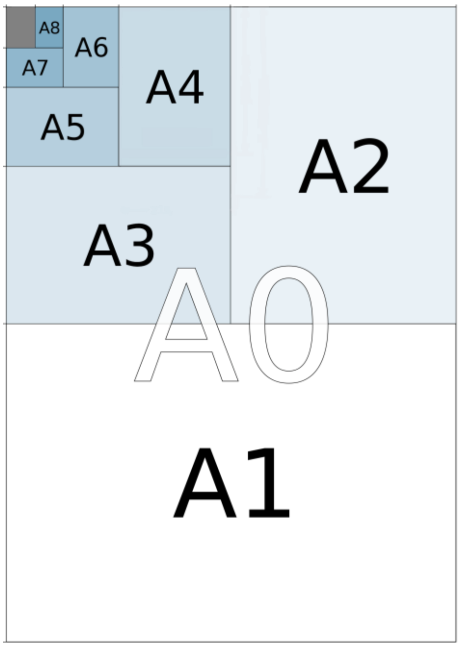
Pravkar opisan način razvrščanja pa uporabljamo pri abecednem razvrščanju! Gre za t.i. leksikografsko razvrščanje. Pri razvrščanju otrok po abecedi uporabljajo učitelji isti postopek: iščejo prvo črko v priimku (ali imenu) po kateri se dva priimka razlikujeta in na osnovi te črke določijo vrstni red. Ker naši bobrčki uporabljajo črke za merjenje dolžine in dolžina posamezne črke ustreza tudi njenemu mestu v abecedi, bi lahko izmerjene dolžine čolnov razvrstili tudi z leksikografskim razvrščanjem:

BF > BG > CDEH > CEFGH.

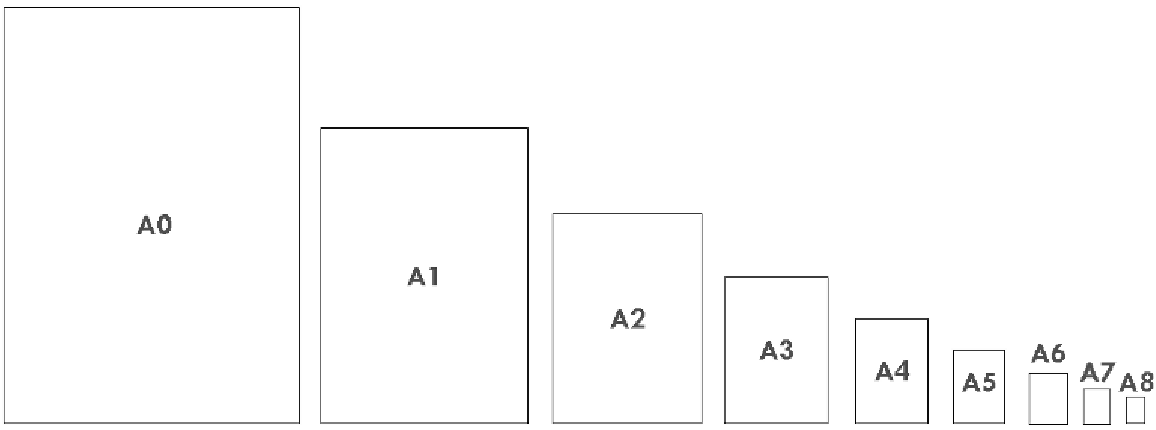
Pravilen vrstni red je Ana, Marija, David in Janez (odgovor C).

Posetnice

Standardne velikosti papirjev dobimo, kot kaže slika, iz osnovnega lista velikosti A0 (1189 x 841 mm). Če list papirja A0 razpolovimo, dobimo dva lista velikosti A1 in če sedaj razpolovimo list velikosti A1, dobimo dva lista velikost A2 ter tako naprej.



Na zalogi imamo 9 listov različnih velikosti, kot kaže spodnja slika, A0, A1, A2, A3, A4, A5, A6, A7 in A8.



Narediti želimo 19 posetnic velikosti A8 tako, da bomo porabili celotne liste papirja iz naše zaloge brez odpadkov. Katere liste naj porabimo?

A) A4, A7 in A8

B) A3 in A7s

C) A5, A6 in A8

D) A4 in A6

Ozadje naloge je spet dvojiški sistem. Če namreč vsaki velikosti papirja dodelimo ustrezno dvojiško utež (ustrezen bit v dvojiškem zapisu), potem se vprašanje iz naloge prevede na vprašanje: kateri biti morajo biti prižgani (enaki 1) v zapisu števila z vrednostjo 19? Torej velikosti A8 dodelimo težo 1 (2^0), A7 težo 2 (2^1), A6 težo 4 (2^2), A5 težo 8 (2^3), A4 težo 16 (2^4), A3 težo 32 (2^5), A2 težo 64 (2^6), A1 težo 128 (2^7) in A0 težo 256 (2^8). Z drugimi besedami, A0 list vsebuje natanko 256 listov velikosti A8, A1 list natanko 128 listov velikosti A8, A2 list natanko 64 listov A8, A3 list natanko 32 listov A8, A4 list natanko 16 listov A8, A5 list natanko 8 listov A8, A6 čist natanko 4 liste A8, A7 list natanko 2 lista A8 in list A8 natanko 1 list velikosti A8.

Število 19 v dvojiškem zapisu z devetimi biti:

$000010011 = 0*2^8 + 0*2^7 + 0*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$

Zapisano z listi je to : 0*A0 + 0*A1 + 0*A2 + 0*A3 + 1*A4 + 0*A5 + 0*A6 + 1*A7 + 1*A8.

Pravilen odgovor je torej odgovor A).

Raznašalka pizz

Naloga spet skriva binarni zapis števil. Spomnimo se, da je vsako število v dvojiškem zapisu zapisano enoznačno, tj. unikatno. Istega števila namreč ne moremo zapisati na dva ali več načinov. Ker so bobrčki naročili 11 pizz, je to dvojiško zapisano kot 1011, tj.

$$1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 8 + 0 + 2 + 1$$

Vidimo, da so pozabili pospraviti tablo pred drugo hišo iz leve (tablo s številko 4). Če bi namreč vztrajali, da 4 ostane, ne bi nikakor mogli zapisati števila 11.

Booleova logika

Z biti moramo znati tudi računati. Osnovne operacije računanja z biti podaja Booleova algebra. Tri osnove operacijo, s katerimi je možno tvoriti poljubne funkcije znotraj Booleove algebre (ter tako implementirati vse zahtevne računske postopke v računalniku), so logična vsota (ALI), logični produkt (IN) in negacija (NE). Te tri operacije računajo izključno nad biti in jih običajno definiramo s pravilnostnimi tabelami.

Pravilnostna tabela, ki opisuje logično vsoto (ALI) je:

x	**y**	**x ALI y**
0	0	0
0	1	1
1	0	1
1	1	1

Z besedami jo opišemo takole: *Če sta vhoda x ALI y enaka 1, potem je rezultat enak 1.* Z drugimi besedami, *vsaj eden od vhodov mora biti* 1, da bo rezultat 1.

Pravilnostna tabela, ki opisuje logični produkt (IN) je:

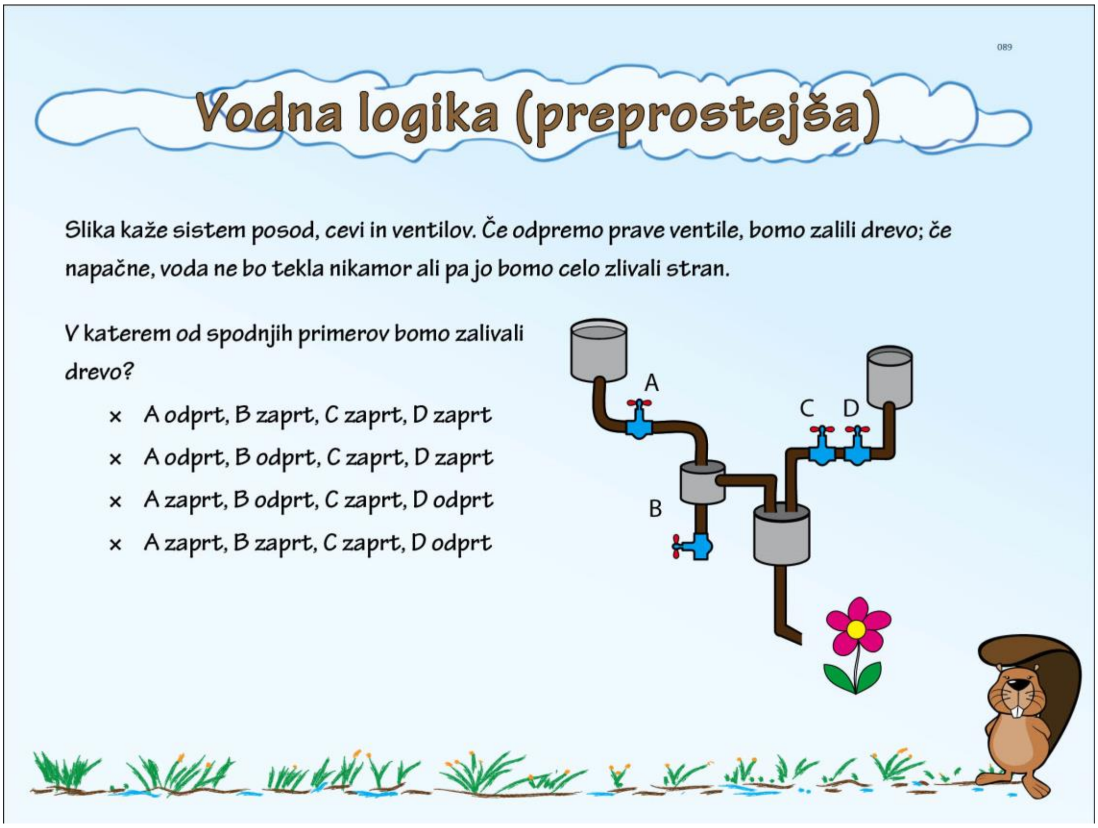
x	**y**	**x IN y**
0	0	0
0	1	0
1	0	0
1	1	1

Z besedami jo opišemo takole: *Če sta vhoda x ALI y enaka 0, potem je rezultat enak 0.* Z drugimi besedami, *oba vhoda hkrati morata biti 1, da bo rezultat 1.*

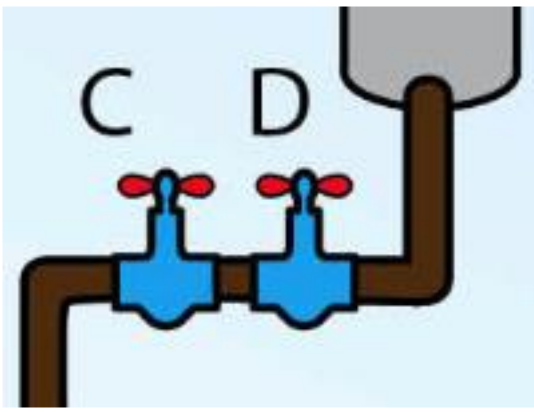
Pravilnostna tabela, ki opisuje logično negacijo (NE) je:

x	**NE x**
0	1
1	0

Vodna logika



Naloga v sebi skriva Booleovo logiko oz. funkcije IN, ALI in NE. Poglejmo si najprej, kako bi prišla voda do cveta po desni cevi, tj. skozi ventila C in D (spodnja slika).



Takoj vidimo, da **morata biti odprta oba** hkrati, če želimo, da voda teče. Lahko rečemo, da ***voda mora teči skozi ventila A IN B, če želimo, da priteče ven***. Pretok vode skozi ventil A in B lahko zapišemo s spodnjo pravilnostno tabelo:

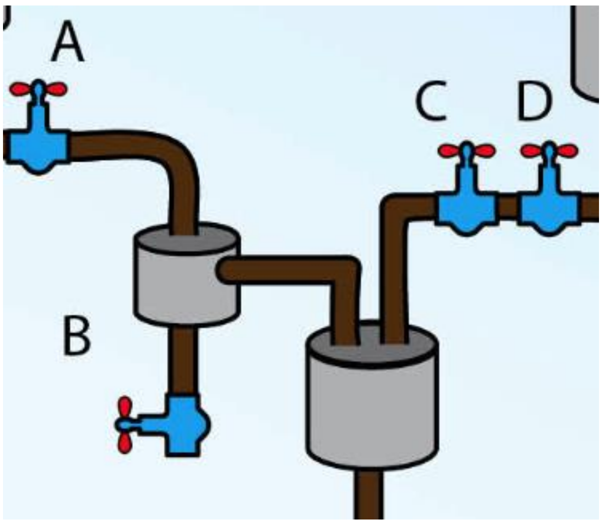
Ventil A	Ventil B	Izhod
Ne teče	Ne teče	Ne teče
Ne teče	Teče	Ne teče
Teče	Ne teče	Ne teče
Teče	Teče	Teče

Če bi pojav, da voda teče, "kodirali" z 1, in pojav ko voda ne teče, "kodirali " z 0, bi lahko v Booleovi algebri to opisali s pravilnostno tabelo:

Ventil A	Ventil B	Izhod
0	0	0
0	1	0
1	0	0
1	1	1

Ta tabela pa opisuje logično operacijo IN.

Poglejmo sedaj, kako voda priteče po levi cevi. Po levi cevi voda priteče le, če je odprt ventil A. Kaj pa skozi osrednjo posodo do cveta (spodnja slika)?



Če odmislimo ventil B, ki mora biti vedno zaprt, bo ***voda skozi osrednjo posodo tekla, ko bo odprt ventil A ALI, ko bosta odprta oba ventila C in D.*** Pretok vode skozi osrednjo posodo lahko zapišemo s spodnjo pravilnostno tabelo:

Ventil A	Ventila C in D	Izhod
Ne teče	Ne teče	Ne teče
Ne teče	Teče	Teče
Teče	Ne teče	Teče
Teče	Teče	Teče

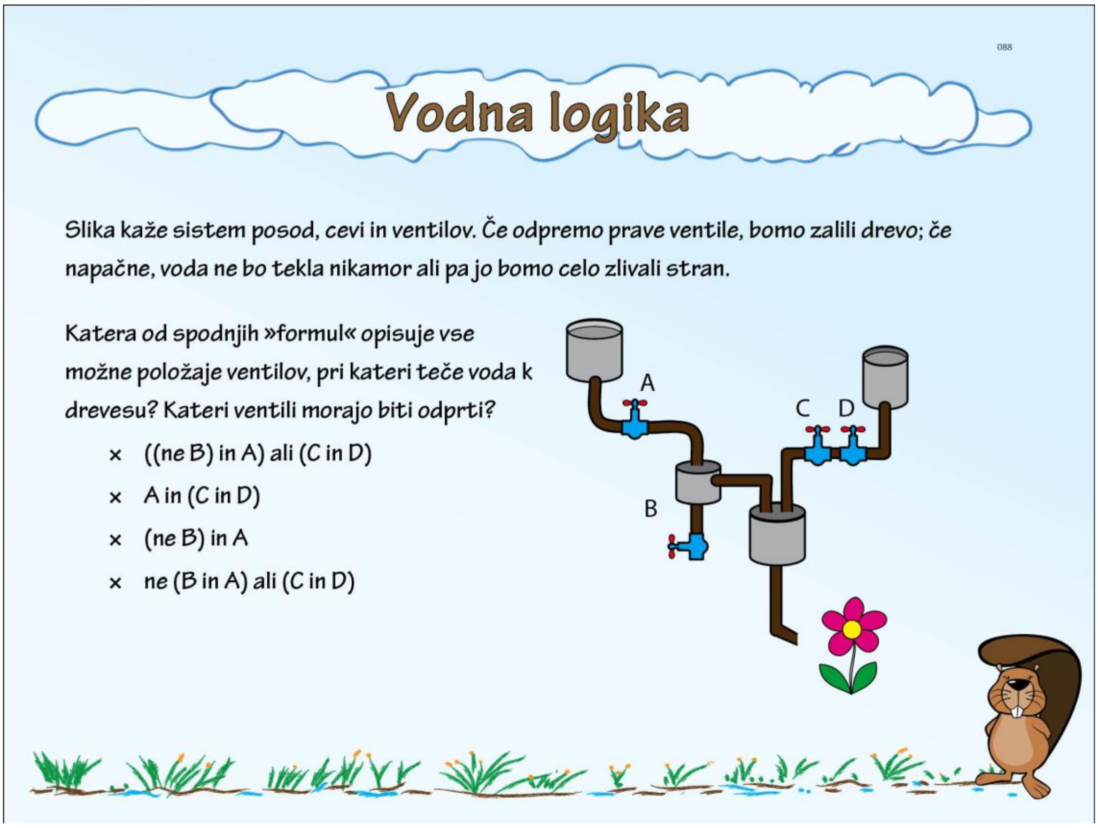
Če bi pojav, da voda teče, “kodirali” z 1, in da voda ne teče “kodirali ” z 0, bi lahko v Booleovi algebri to opisali s pravilnostno tabelo:

Ventil A	Ventila C in D	Izhod
0	0	0
0	1	1
1	0	1
1	1	1

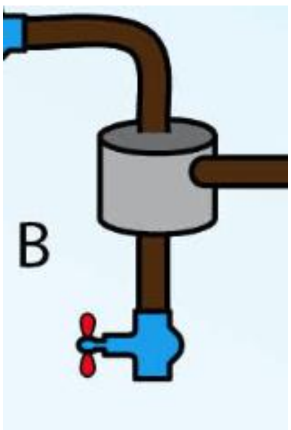
Ta tabela pa opisuje logično operacijo ALI.

Naloga je sila preprosta in jo otroci rešijo brez težav in brez poznavanja Booleove logike. Ker je v vseh ponujenih odgovorih ventil C vedno zaprt, voda ne teže skozi desno cev. Edina možnost da zalijemo cvet je, da je B zaprt in A odprt (prvi odgovor).

Vodna logika 2



Naloga v sebi spet skriva Booleovo logiko oz. operacije IN, ALI in NE. IN in ALI smo predstavili v prejšnji nalogi. Kje se skriva NE? Poglejmo si pomen postavitve posode z odtokom pred ventilom B:



Voda bo iz posode odtekala po desni cevi, če bo ventil zaprt, sicer bo šla skozi ventil.

Zapišimo to s pravilnostno tabelo:

Ventil B	**Izhod (desna cev)**
Ne teče	Teče
Teče	Ne teče

Če bi pojav, da voda teče, “kodirali” z 1, in da voda ne teče “kodirali ” z 0, bi lahko v Booleovi algebri to opisali s pravilnostno tabelo:

Ventil B	**Izhod (desna cev)**
0	1
1	0

Ta tabela opisuje logično operacijo NE.

Če sedaj z logičnimi operacijami zapišemo “formulo” za zalivanje, je ta:

((NE B) IN A) ALI (C IN D)