

Vzorový príklad pre MIPSIM  
Princípy počítačového inžinierstva 2022  
Mária Matušisková

### Zadanie

Napíšte program, ktorý bude simulovať pohyb hráča v bludisku podľa obrázka. Ľavé horné políčko bludiska má súradnice (riadok, stĺpec) = (1,1) a pravé spodné políčko má súradnice (8,3). V bludisku sa nachádzajú nepriechodné steny, vyznačené hrubou čiarou. Hráč môže začínať hru na ktoromkoľvek políčku a môže vykonávať kroky o 1 políčko smerom na niektorú svetovú stranu. Hráč sa môže pokúsiť aj o krok smerom do steny, ale jeho súradnice sa v takomto prípade nezmenia. Z bludiska vedie jeden východ na jeho okraji.

Od adresy **0h** so 4-bajtovými rozstupmi (4h, 8h, ch, 10h, 14h, 18h, 1ch, 20h, atď.) bude pred spustením programu v pamäti údajov uložená postupnosť hodnôt reprezentujúcich pohyby hráča o 1 políčko nasledovne:

- **1h** – pohyb hore,
- **2h** – pohyb vpravo,
- **3h** – pohyb dole,
- **4h** – pohyb vľavo,
- **0h** – koniec.

Hodnota 0 reprezentuje ukončenie postupnosti. Zistíte, či počas vykonávania zadanej postupnosti hráč nájde alebo nenájde východ z bludiska. Ak áno, program by mal ihneď skončiť a na adresu **c0h** uložte hodnotu 1. Ak nie, teda program načíta ukončovací znak 0, uložte na adresu **c0h** hodnotu 0. Pod nájdením východu sa rozumie vykonanie kroku smerom k východu z bludiska. Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu hráča uchováajte na adresách **a0h** resp. **b0h**.

Po načítaní hodnoty **0h** alebo pri východe z bludiska sa program ukončí. Môžete predpokladať, že v postupnosti sa iné čísla ako **0h-4h** nebudú nachádzať.

### Úloha 24

	1	2	3
1			
2			
3			
4			
5			
6			
7			
8			

## Riešenie

### Pamäť programu

Adr.	Label	Inštrukcia	Komentár
0h		LW \$25,00a0(\$0)	načítame začiatočnú riadkovú súradnicu z PÚ z adresy a0h do registra R25
4h		LW \$26,00b0(\$0)	načítame začiatočnú stĺpcovú súradnicu z PÚ z adresy b0h do registra R26
8h	zac	LW \$22,0000(\$20)	do registra R22 načítame prvok postupnosti z PÚ z adresy, na ktorú ukazuje ukazovateľ v registri R20
ch		ADDI \$20,\$20,0004	zvážime ukazovateľ v registri R20 o 4, aby ukazoval na ďalší prvok postupnosti v poradí
10h		NOP	
14h		BEQ \$22,\$1,jedna	ak je načítaný prvok postupnosti v reg. R22 rovný 1 (konštantu 1 máme uloženú v reg. R1) skoč na podprogram pre vykonanie pohybu hore ktorý sa nachádza na labeli „jedna“
18h		BEQ \$22,\$2,dva	ak je načítaný prvok postupnosti v reg. R22 rovný 2 (konštantu 2 máme uloženú v reg. R2) skoč na podprogram pre vykonanie pohybu vpravo ktorý sa nachádza na labeli „dva“
1ch		BEQ \$22,\$3,tri	ak je načítaný prvok postupnosti v reg. R22 rovný 3 (konštantu 3 máme uloženú v reg. R3) skoč na podprogram pre vykonanie pohybu dole ktorý sa nachádza na labeli „tri“
20h		BEQ \$22,\$4,styri	ak je načítaný prvok postupnosti v reg. R22 rovný 4 (konštantu 4 máme uloženú v reg. R4) skoč na podprogram pre vykonanie pohybu vľavo ktorý sa nachádza na labeli „styri“ inak to musí byť 0 a pokračujeme ďalej:
24h		SW \$25,00a0(\$0)	uložíme výslednú riadkovú adresu z R25 do PÚ na adresu a0h
28h		SW \$26,00b0(\$0)	uložíme výslednú stĺpcovú adresu z R26 do PÚ na adresu b0h
2ch		BEQ \$0,\$0,kon	a skočíme niekam na koniec programu
...	...		
40h	jedna	BNEQ \$25,\$1,1podX4	je riadková súradnica rovná 1? ak nie, skočíme na 1podX4 (jedna - podmienka, že súradnica x sa rovná 4)
44h		BEQ \$0,\$0,1podY1	ak áno, skoč na 1podY1 (jedna – podmienka, že súradnica y sa rovná 1 )
48h	pohyb1	SUBI \$25,\$25,0001	pohyb hore – znížime riadkovú súradnicu o 1
4ch		BEQ \$0,\$0,zac	a skočíme naspäť na začiatok
...	...		
54h	1podX4	BNEQ \$25,\$4,pohyb1	je riadková súradnica rovná 4? ak nie, môžeme sa pohnúť hore
58h		BEQ \$0,\$0,1podY2	ak áno, skoč na 1podY2 (jedna – podmienka, že súradnica y sa rovná 2)
5ch	1podY1	BNEQ \$26,\$1,zac	je stĺpcová súradnica rovná 1? ak nie, nemôžeme sa už pohnúť vyššie, skočíme naspäť na začiatok
60h		BEQ \$0,\$0,uloz1	ak áno, skoč na ulož, našli sme východ z bludiska
...	...		
68h	uloz1	SW \$1,00c0(\$0)	ulož prvok z registra R1 hodnotu do pamäte
6ch		BEQ \$0,\$0,kon	a skočíme niekam na koniec programu
...	...		
74h	1podY2	BNEQ \$26,\$2,pohyb1	je stĺpcová súradnica rovná 2? ak nie, môžeme sa pohnúť hore
78h		BEQ \$0,\$0,zac	ak áno, sme na hornom okraji, nerobíme nič, skočíme naspäť na začiatok
...	...		

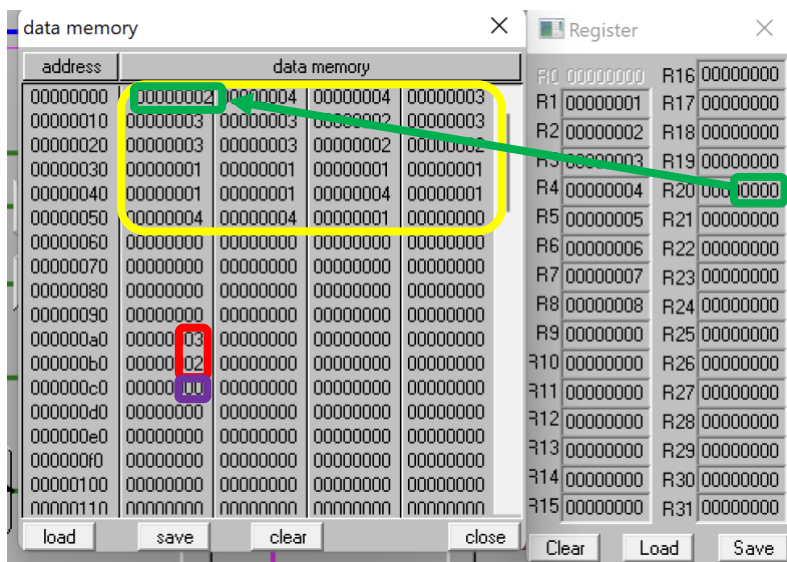
80h	dva	BNEQ \$26,\$3,2podY2	je stĺpcová súradnica rovná 3? ak nie, skoč na 2podY2 (dva – podmienka, že súradnica y sa rovná 2)
84h		BEQ \$0,\$0,zac	ak áno, sme na pravom okraji, nerobíme nič, skočíme naspäť na začiatok
88h	pohyb2	ADDI \$26,\$26,0001	pohyb vpravo – zvýšime stĺpcovú súradnicu o 1
8ch		BEQ \$0,\$0,zac	a skočíme naspäť na začiatok
...	...		
94h	2podY2	BNEQ \$26,\$2,2podY1	je stĺpcová súradnica rovná 2? ak nie, skoč na 2podY1 (dva – podmienka, že súradnica y sa rovná 1)
98h		BEQ \$0,\$0,2podX2	ak áno, skoč na 2podX2 (dva – podmienka, že súradnica x sa rovná 2)
9ch	2podY1	BNEQ \$26,\$1,pohyb2	je stĺpcová súradnica rovná 1? ak nie, môžeme sa pohnúť vpravo
a0h		BEQ \$0,\$0,2podX4	ak áno, skoč na podmienku 2podX4 (dva – podmienka, že súradnica x sa rovná 4)
a4h	2podX2	BNEQ \$25,\$2,2podX3	je riadková súradnica rovná 2? ak nie, skočíme na 2podX3 (dva - podmienka, že súradnica x sa rovná 3)
a8h		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
	...	...	
b0h	2podX3	BNEQ \$25,\$3,pohyb2	je riadková súradnica rovná 3? ak nie, môžeme sa pohnúť vpravo
b4h		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
b8h	2podX4	BNEQ \$25,\$4,pohyb2	je riadková súradnica rovná 4? ak nie, môžeme sa pohnúť vpravo
bch		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
...	...		
c8h	tri	BNEQ \$25,\$8,3podX3	je riadková súradnica rovná 8? ak nie, skoč na 3podX3 (tri – podmienka, že súradnica x sa rovná 3)
cch		BEQ \$0,\$0,zac	ak áno, sme na dolnom okraji, nerobíme nič, skočíme naspäť na začiatok
d0h	pohyb3	ADDI \$25,\$25,0001	pohyb dole – zvýšime riadkovú súradnicu o 1
d4h		BEQ \$0,\$0,zac	a skočíme naspäť na začiatok
...	...		
dch	3podX3	BNEQ \$25,\$3,pohyb3	je riadková súradnica rovná 3? ak nie, môžeme sa pohnúť dole
e0h		BEQ \$0,\$0,3podY2	ak áno, skoč na 3podY2 (tri – podmienka, že súradnica y sa rovná 2)
...	...		
e8h	3podY2	BNEQ \$26,\$2,pohyb3	je stĺpcová súradnica rovná 2? ak nie, môžeme sa pohnúť dole
ech		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
...	...		
100h	štyri	BNEQ \$26,\$1,4podY3	je stĺpcová súradnica rovná 1? ak nie, skoč na 4podY3 (štyri – podmienka, že súradnica y sa rovná 3)
104h		BEQ \$0,\$0,zac	ak áno, sme na ľavom okraji, nerobíme nič, skočíme naspäť na začiatok
108h	pohyb4	SUBI \$26,\$26,0001	pohyb vľavo – znížime stĺpcovú súradnicu o 1
10ch		BEQ \$0,\$0,zac	a skočíme naspäť na začiatok
...	...		
114h	4podY3	BNEQ \$26,\$3,4podY2	je stĺpcová súradnica rovná 3? ak nie, skoč na 4podY2 (štyri – podmienka, že súradnica y sa rovná 2)
118h		BEQ \$0,\$0,4podX2	ak áno, skoč na 4podX2 (štyri – podmienka, že súradnica x sa rovná 2)
...	...		
120h	4podY2	BNEQ \$26,\$2,pohyb4	je stĺpcová súradnica rovná 2? ak nie, môžeme sa pohnúť vľavo
124h		BEQ \$0,\$0,4podX4	ak áno, skoč na 4podX4 (štyri – podmienka, že súradnica x sa rovná 4)
128h	4podX2	BNEQ \$25,\$2,4podX3	je riadková súradnica rovná 2? ak nie, skoč na 4podX3 (štyri – podmienka, že súradnica x sa rovná 3)

12ch		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
...	...		
134h	4podX3	BNEQ \$25,\$3,pohyb4	je riadková súradnica rovná 3? ak nie, môžeme sa pohnúť vľavo
138h		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
...	...		
140h	4podX4	BNEQ \$25,\$4,pohyb4	je riadková súradnica rovná 4? ak nie, môžeme sa pohnúť vľavo
144h		BEQ \$0,\$0,zac	ak áno, skočíme naspäť na začiatok
...	...		
150h	kon	SW \$25,00a0(\$0)	uloženie konečnej súradnice x do pamäte
154h		SW \$26,00b0(\$0)	uloženie konečnej súradnice y do pamäte
158h		NOP	koniec programu

Simulujeme napríklad takúto postupnosť krokov od adresy 305: 2 4 4 3 3 3 2 3 3 3 2 2 1 1 1 1 1 4 1 4 4 1 0. Žltou farbou sú vyznačené tie kroky, ktoré sa nepodaria vykonať, lebo hráč narazí na okraj bludiska alebo steny v bludisku. Podľa zadania ho nemôže opustiť, okrem pozície (1,1), čo je ľavý horný okraj bludiska, takže ostane na súčasnej pozícii. Konečné súradnice hráča po vykonaní postupnosti krokov budú (riadok, stĺpec) = (1,1), teda východ z bludiska v smere nahor – podľa obrázka nižšie:

	1	2	3
1	H↑	←	←
2			↑
3	↓	←H	↑
4	↓		↑
5	↓		↑
6	→	↓	↑
7		↓	↑
8		→	↑

Obsah registrov a pamäti údajov pred spustením programu

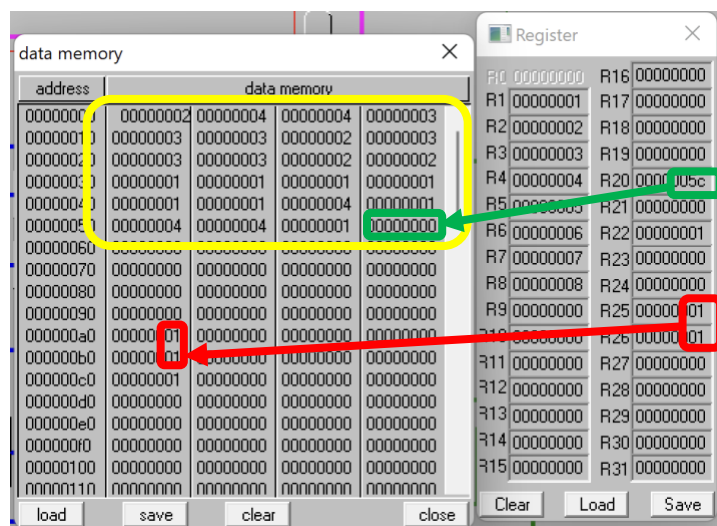


Register	Údaj	Komentár
R1	1h	konštanta 1 na porovnávanie
R2	2h	konštanta 2 na porovnávanie
R3	3h	konštanta 3 na porovnávanie
R4	4h	konštanta 4 na porovnávanie
R5	5h	konštanta 5 na porovnávanie
R6	6h	konštanta 6 na porovnávanie

R7	7h	konštanta 7 na porovnávanie
R8	8h	konštanta 8 na porovnávanie
R20	0h	ukazovateľ do postupnosti prvkov, na začiatku ukazuje na 1. prvok
R22	0h	sem sa bude načítavať prvok postupnosti z pamäte údajov
R25	0h	sem sa načíta začiatková riadková súradnica z PÚ z adresy <b>a0h</b> následne sa bude počas behu programu aktualizovať
R26	0h	sem sa načíta stĺpcová súradnica z PÚ z adresy <b>b0h</b> následne sa bude počas behu programu aktualizovať

Adresa	Údaj	Komentár
<b>0h – 40h</b>	1h, 1h, 1h, 2h, 2h, ..., 0h	postupnosť krokov
<b>a0h</b>	3h	začiatočná riadková súradnica
<b>b0h</b>	2h	začiatočná stĺpcová súradnica
<b>c0h</b>	0h	začiatočná výsledná hodnota

### Obsah registrov a pamäti údajov po spustení programu



Register	Údaj	Komentár
<b>R20</b>	5ch	ukazovateľ do postupnosti prvkov, na konci ukazuje za posledný prvok
<b>R25</b>	1h	konečná riadková súradnica, ktorá sa uloží do PÚ na adresu <b>a0h</b>
<b>R26</b>	1h	konečná stĺpcová súradnica, ktorá sa uloží do PÚ na adresu <b>b0h</b>

Adr.	Údaj	Komentár
<b>0h – 40h</b>	1h, 1h, 1h, 2h, 2h, ..., 0h	postupnosť krokov
<b>a0h</b>	1h	konečná riadková súradnica
<b>b0h</b>	1h	konečná stĺpcová súradnica
<b>c0h</b>	0h	konečná výsledná hodnota

### Zhodnotenie:

Bolo veľmi náročné poprepájať všetky podmienky, tak aby to bolo zrozumiteľné programu. Program skákal z inštrukcie na inštrukciu, najviac boli využité inštrukcie BNEQ a BEQ, ktoré porovnávali rovnosť prvkov a ak sa rovnali alebo nerovnali, podľa toho program skočil na dané adresy.

Využili sme Registre od R0 po R8, kde boli súradnice bludiska spolu so smermi pohybu. R20 bol využitý na ukazovanie na prvky, ktoré potom R22 využíval. R25 a R26 boli využívané na ukazovanie momentálnej pozície hráča.

Zhodnotenie, program by mal byť funkčný za predpokladu, ak dané smery budú zadané od 0 po 4.