

## Zadanie 4 (Úloha 43)

### Princípy počítačového inžinierstva 2023

Ondrej Krajčovič

#### Zadanie

Napište program, ktorý bude simulovať pohyb hráča na šachovnici podľa obrázka. Ľavé horné políčko šachovnice má súradnice (riadok, stĺpec) = (1,1) a pravé spodné políčko má súradnice (5,5). Na šachovnici sa nachádza určitý počet pokladov a pascí. Hráč môže začínať hru na niektorom z voľných políčok.

Od adresy 305 bude v pamäti údajov pred spustením programu uložená postupnosť čísel (každé číslo na samostatnej adrese) reprezentujúca trasu, ktorú hráč počas hry vykoná, a to nasledovným spôsobom. Pohyb smerom:

- severovýchod = 1,
- juhovýchod = 2,
- juhozápad = 3,
- severozápad = 4.

Hodnota 0 reprezentuje ukončenie postupnosti. Pre postupnosť budú vyhradené adresy 305-399, neukladajte tam žiadne iné údaje. Hráč má povolené prechádzať aj cez okraje šachovnice – ak sa napríklad nachádza na súradniciach (3,5) a vykoná pohyb smerom na severovýchod, jeho nové súradnice budú (2,1). Zistite, či počas vykonávania zadanej postupnosti hráč

- a) našiel poklad. V tomto prípade by mal program ihneď skončiť a na adresu 302 uložte hodnotu 1.
- b) stúpil na pascu. V tomto prípade by mal program ihneď skončiť a na adresu 302 uložte hodnotu 2.
- c) nenastalo a) ani b), teda program načítal ukončovací znak 0. V tomto prípade uložte na adresu 302 hodnotu 0.

Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu hráča uchovávať na adresách 300 a 301.

	1	2	3	4	5
1		T			
2			P		
3				P	
4					T
5					

P – poklad, T – pasca

## Vlastné Inštrukcie:

PRIDJ	Program:	UBER	Program:	EXIT	Program:
*	IR adr. > MBR	**	IR adr. > MBR	***	IR adr. > MBR
	MBR adr. => MAR		MBR adr. => MAR		MBR adr. => MAR
	READ MEM		READ MEM		ACC => MBR
	INCREMENT ACC		DECREMENT ACC		WRITE MEM
	INCREMENT ACC		DECREMENT ACC		HALT
	INCREMENT ACC		DECREMENT ACC		
	INCREMENT ACC		DECREMENT ACC		
	INCREMENT ACC		DECREMENT ACC		
	IR adr. => MBR		IR adr. => MBR		
	MBR adr. => MAR		MBR adr. => MAR		
	ACC => MBR		ACC => MBR		
	WRITE MEM		WRITE MEM		

PRIDJ – Inštrukcia Pridaj \*

Preskočí na súradnici argumentu na druhú stranu šachovnice kladne  
(zvýši argument typu pointer o 5)

UBER – Inštrukcia Uber \*\*

Preskočí na súradnici argumentu na druhú stranu šachovnice záporne  
(zníži argument typu pointer o 5)

EXIT – Inštrukcia exit/return \*\*\*

Slúži ako funkcia return, uloží načítanú hodnotu na pozíciu 302 a ukončí program

## Riešenie

Adresa	Inštrukcia	Komentár
0	LOAD [mem 302]	načítaj prvok postupnosti, ukazovateľ je uložený na adrese 302
1	CMP 1	je to 1?
2	JC 20	ak áno, skoč na podprogram pre pohyb hráča hore na adrese 20
3	CMP 2	ak nie, je to 2?
4	JC 40	ak áno, skoč na podprogram pre pohyb hráča vpravo na adrese 40
5	CMP 3	ak nie, je to 3?
6	JC 60	ak áno, skoč na podprogram pre pohyb hráča dole na adrese 60
7	CMP 4	ak nie, je to 4?
8	JC 80	ak áno, skoč na podprogram pre pohyb hráča vľavo na adrese 80
9	STORE [302]	ak nie, musí to teda byť 0, takže uložíme 0 na miesto 302
10		ukončíme program
...		
12	INC [302]	posun na ďalší prvok postupnosti, zväčši ukazovateľ o 1
13	JMP 100	a skoč na porovnávanie, či je na súradniciach poklad/pasca
...		
20	LOAD [300]	načítaj riadkovú súradnicu
21	CMP 1	je to 1? sme na hornom okraji šachovnice?
22	JNC 25	ak nie, posuň sa na riadok 25 (preskoč zväčšenie o 5)
23	PRIDJ [300]	ak áno, posuň sa na dolný okraj (zväčši riadkovú súradnicu o 5)
24	DEC [300]	Posuň sa hore o 1
25	LOAD [301]	načítaj stĺpcovú súradnicu
26	CMP 5	je to 5? sme na pravom okraji šachovnice?
27	JNC 31	ak nie, posuň sa na riadok 31 (preskoč zmenšenie o 5)
28	UBER [301]	ak áno, posuň sa na ľavý okraj (zmenši stĺpcovú súradnicu o 5)
29	INC [301]	Posuň sa doprava o 1
30	JMP 12	Preskoč na ďalší prvok
...		
40	LOAD [300]	načítaj riadkovú súradnicu
41	CMP 5	je to 5? sme na dolnom okraji šachovnice?
42	JNC 45	ak nie, posuň sa na riadok 45 (preskoč zmenšenie o 5)
43	UBER [300]	ak áno, posuň sa na horný okraj (zmenši riadkovú súradnicu o 5)
44	INC [300]	Posuň sa dole o 1
45	LOAD [301]	načítaj stĺpcovú súradnicu
46	CMP 5	je to 5? sme na pravom okraji šachovnice?
47	JNC 51	ak nie, posuň sa na riadok 51 (preskoč zmenšenie o 5)
48	UBER [301]	ak áno, posuň sa na ľavý okraj (zmenši stĺpcovú súradnicu o 5)
49	INC [301]	Posuň sa doprava o 1
50	JMP 12	Preskoč na ďalší prvok
...		

60	LOAD [300]	načítaj riadkovú súradnicu
61	CMP 5	je to 5? sme na dolnom okraji šachovnice?
62	JNC 65	ak nie, posuň sa na riadok 65 (preskoč zmenšenie o 5)
63	UBER [300]	ak áno, posuň sa na horný okraj (zmenši riadkovú súradnicu o 5)
64	INC [300]	Posuň sa dole o 1
65	LOAD [301]	načítaj stĺpcovú súradnicu
66	CMP 1	je to 1? sme na ľavom okraji šachovnice?
67	JNC 71	ak nie, posuň sa na riadok 71 (preskoč zmenšenie o 5)
68	PRIDJ [301]	ak áno, posuň sa na pravý okraj (zväčši stĺpcovú súradnicu o 5)
69	DEC [301]	Posuň sa doľava o 1
70	JMP 12	Preskoč na ďalší prvok
...		
80	LOAD [300]	načítaj riadkovú súradnicu
81	CMP 1	je to 1? sme na hornom okraji šachovnice?
82	JNC 84	ak nie, posuň sa na riadok 85 (preskoč zväčšenie o 5)
83	PRIDJ [300]	ak áno, posuň sa na dolný okraj (zväčši riadkovú súradnicu o 5)
84	DEC [300]	Posuň sa hore o 1
85	LOAD [301]	načítaj stĺpcovú súradnicu
86	CMP 1	je to 1? sme na ľavom okraji šachovnice?
87	JNC 89	ak nie, posuň sa na riadok 91 (preskoč zmenšenie o 5)
88	PRIDJ [301]	ak áno, posuň sa na pravý okraj (zväčši stĺpcovú súradnicu o 5)
89	DEC [301]	Posuň sa doľava o 1
90	JMP 12	Preskoč na ďalší prvok
...		

	1	2	3	4	5
1		T			
2			P		
3				P	
4					T
5					

100	LOAD [300]	načítaj riadkovú súradnicu
101	CMP 1	Je to 1? sme na riadku kde je pasca?
102	JC 115	Ak áno, skoč na podprogram pre porovnávanie stĺpcu pasce
103	CMP 2	Je to 2? sme na riadku kde je poklad?
104	JC 120	Ak áno, skoč na podprogram pre porovnávanie stĺpcu pokladu
105	CMP 3	Je to 3? sme na riadku kde je poklad?
106	JC 125	Ak áno, skoč na podprogram pre porovnávanie stĺpcu pokladu
107	CMP 4	Je to 4? sme na riadku kde je pasca?
108	JC 130	Ak áno, skoč na podprogram pre porovnávanie stĺpcu pasce
109	JMP 0	Skoč na začiatok programu
...		
115	LOAD [301]	načítaj stĺpcovú súradnicu
116	CMP 2	Je to 2? (stúpil na pascu)
117	JC 140	Ak áno (stúpil na pascu), skoč na podprogram vrátenia 2
118	JMP 0	Ak nie, skoč na začiatok programu
...		
120	LOAD [301]	načítaj stĺpcovú súradnicu
121	CMP 3	Je to 2? (našiel poklad)
122	JC 150	Ak áno (našiel poklad), skoč na podprogram vrátenia 1
123	JMP 0	Ak nie, skoč na začiatok programu
...		
125	LOAD [301]	načítaj stĺpcovú súradnicu
126	CMP 4	Je to 2? (našiel poklad)
127	JC 150	Ak áno (našiel poklad), skoč na podprogram vrátenia 1
128	JMP 0	Ak nie, skoč na začiatok programu
...		
130	LOAD [301]	načítaj stĺpcovú súradnicu
131	CMP 5	Je to 2? (stúpil na pascu)
132	JC 140	Ak áno (stúpil na pascu), skoč na podprogram vrátenia 2
133	JMP 0	Ak nie, skoč na začiatok programu
...		
141	LOAD [304]	Načítaj hodnotu na vrátenie: 2 (stúpil na pascu)
142	EXIT 302	Ulož hodnotu (2) z pozície 304 na pozíciu 302 (return)
...		
150	LOAD [303]	Načítaj hodnotu na vrátenie: 1 (našiel poklad)
151	EXIT 302	Ulož hodnotu (1) z pozície 303 na pozíciu 302 (return)

Simulujeme napríklad takúto postupnosť krokov: 2 3 4 4 1 2 1 1 1 0. Konečné súradnice hráča po vykonaní postupnosti krokov budú (riadok, stĺpec) = (1,2) – podľa obrázka nižšie:

	1	2	3	4	5
1		T			
2			P		
3		H		P	
4					T
5					

#### Pamäť údajov pred spustením programu

Adresa	Údaj	Komentár
300	3	začiatočná riadková súradnica hráča
301	2	začiatočná stĺpcová súradnica hráča
302	305	ukazovateľ na momentálne spracúvaný prvok postupnosti (neskôr Miesto na return value)
303	1	Návratová hodnota 1
304	2	Návratová hodnota 2
..	...	...
305	2	postupnosť krokov
306	3	
307	4	
308	4	
309	4	
310	1	
311	2	
312	1	
313	1	
314	1	
321	0	koniec postupnosti

#### Pamäť údajov po skončení programu

Adresa	Údaj	Komentár
300	1	konečná riadková súradnica hráča
301	2	konečná stĺpcová súradnica hráča
302	2	2 (hráč našiel pascu)

Ostatné hodnoty v pamäti údajov ostanú bez zmeny.

## Zhodnotenie

---

Ako prvé som vytvoril štruktúru výberu akcie, ktorá vyhodnocuje aktuálnu pozíciu spolu s aktuálnym vstupom a priraduje mu prislúchajúci podprogram. Potom som vytvoril logickú kompozíciu pohybu, ktorá pomocou mikroinštrukcií zmení pozíciu. Po pohybe program prejde sériou podmienok, ktoré zisťujú, či daný program pokračuje, alebo vráti výstup. Po podrobnej analýze programu som vytvoril vlastné podfunkcie, ktoré som následne implementoval. Výsledkom bolo zníženie počtu riadkov programu, ako aj jeho sprehľadnenie