

4. 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 nachádzajú statické šachové figúrky, ktoré ohrozujú hráča, ak tento sa nachádza na určitých políčkach šachovnice (viď. obrázok). Ďalej sa tu môžu nachádzať prekážky, ktoré bránia figúrkam vo výhľade. Hráč môže začínať hru na niektorom z voľných políčk mimo ohrozenia figúrkou a môže vykonávať kroky o 1 políčko smerom na niektorú svetovú stranu. Hráč má povolené prechádzať aj cez okraje šachovnice – ak sa napríklad nachádza na súradniciach (2,5) a vykoná krok vpravo, jeho nové súradnice budú (2,1).

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:

- hore = 1,
- vpravo = 2,
- dole = 3,
- vľavo = 4.

Hodnota 0 reprezentuje ukončenie postupnosti. Môžete predpokladať, že hráč sa nikdy nepokúsi vstúpiť na políčko s prekážkou, ani na políčko so šachovou figúrkou. Zistite, či počas vykonávania zadanej postupnosti dôjde alebo nedôjde k ohrozeniu hráča šachovou figúrkou. Ak áno, program by mal ihneď skončiť a na adresu 302 uložte hodnotu 1. Ak nie, teda program načíta ukončovací znak 0, uložte na adresu 302 hodnotu 0. Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu hráča uchovávajte na adresách 300 resp. 301.

Riešenie 4. zadania

Úloha 11

	1	2	3	4	5
1				•	
2		■	•	V	•
3				•	
4				•	
5				•	

V – veža, • - ohrozenie, ■ - prekážka

Rexim- Pamäť programu:

Pamäť:		
0	LOAD [mem(303)]	10 10010
1	CMP 1	00 01001
2	JZ 20	00 01011
3	CMP 2	00 01001
4	JZ 40	00 01011
5	CMP 3	00 01001
6	JZ 65	00 01011
7	CMP 4	00 01001
8	JZ 85	00 01011
9	CMP 0	00 01001
10	HALT	00 00000
11	-----	
12	INC [303]	01 01111
13	JMP 0	00 01010
14	-----	
15	-----	
16	-----	
17	-----	
18	-----	
19	-----	
20	LOAD [301]	01 10010
21	CMP 1	00 01001
22	JZ 31	00 01011
23	LOAD [301]	01 10010
24	CMP 3	00 01001
25	JNZ 36	00 01100
26	LOAD [300]	01 10010
27	CMP 1	00 01001
28	JZ 36	00 01011
29	OHR [302]	01 10100
30	-----	
31	LOAD [301]	01 10010
32	ADD 4	00 00001
33	STORE [301]	01 10001
34	JMP 12	00 01010
35	-----	
36	DEC [301]	01 10000
37	JMP 12	00 01010
38	-----	
39	-----	
40	LOAD [300]	01 10010
41	CMP 5	00 01001
42	JZ 55	00 01011
43	LOAD [300]	01 10010
44	CMP 3	00 01001
45	JNZ 47	00 01100
46	OHR [302]	01 10100
47	LOAD [300]	01 10010
48	CMP 1	00 01001
49	JNZ 60	00 01100
50	LOAD [301]	01 10010
51	CMP 2	00 01001
52	JNZ 60	00 01100
53	OHR [302]	01 10100
54	-----	
55	LOAD [300]	01 10010
56	SUB 4	00 00010
57	STORE [300]	01 10001
58	JMP 12	00 01010
59	-----	
60	INC [300]	01 01111
61	JMP 12	00 01010
62	-----	
63	-----	

Riadky 0 až 8 fungujú ako taký príkaz „switch“. Podľa načítaného čísla sa rozhoduje, kde sa skočí.

Príkaz HALT na riadku číslo 10

Po jednotlivých skokoch z riadkov 1 – 8 na telo programu sa vracia sem a skáče na začiatok po zvýšení [303] o 1. To znamená, že na pozícii nula sa načíta pamäť o jedna väčšia.

Ak bolo na vstupe číslo 1, program skočil na riadok 20. **Pohyb hore = Y**

Ak je Y rovné 1, tak skočí na riadok 31, pripočíta sa 4 a pozícia je rovná 5. (žlté políčka)

Ak je Y rovné 3, tak sa pozrie na hodnotu X, ak je rovná 1 (červené políčko), tak skáče na riadok 36, odpočíta sa 1. Inak sa jedná o modré políčka a program končí inštrukciou OHR.

	1	2	3	4	5
1				•	
2		■	•	V	•
3	■			•	
4				•	
5				•	

Ak bolo na vstupe číslo rovné 2, program skočil na riadok 40. **Pohyb vpravo = X**

Ak je X rovné 5, tak skočí na riadok 55, odpočíta sa 4 a pozícia je rovná 1. (žlté políčka)

Ak je X rovné 3, tak sa spustí inštrukcia OHR. (modré políčka)

Ak je X rovné 1 a zároveň je Y rovné 2 (červené políčko), tak sa spustí inštrukcia OHR

Ak nič z tohto neplatí, pripočíta sa 1.

	1	2	3	4	5
1				•	
2	■	■	•	V	•
3				•	
4				•	
5				•	

```

62 -----
63 -----
64 -----
65 LOAD [301]      01 10010
66 CMP 5           00 01001
67 JZ 76           00 01011
68 LOAD [301]      01 10010
69 CMP 1           00 01001
70 JNZ 81          00 01100
71 LOAD [300]      01 10010
72 CMP 1           00 01001
73 JZ 81           00 01011
74 OHR [302]       01 10100
75 -----
76 LOAD [301]      01 10010
77 SUB 4           00 00010
78 STORE [301]     01 10001
79 JMP 12          00 01010
80 -----
81 INC [301]       01 01111
82 JMP 12          00 01010
83 -----
84 -----
85 LOAD [300]      01 10010
86 CMP 5           00 01001
87 JNZ 89          00 01100
88 OHR [302]       01 10100
89 LOAD [300]      01 10010
90 CMP 1           00 01001
91 JNZ 105         00 01100
92 LOAD [301]      01 10010
93 CMP 2           00 01001
94 JNZ 100         00 01100
95 OHR [302]       01 10100
96 -----
97 -----
98 -----
99 -----
100 LOAD [300]     01 10010
101 ADD 4          00 00001
102 STORE [300]    01 10001
103 JMP 12         00 01010
104 -----
105 DEC [300]      01 10000
106 JMP 12         00 01010
107 -----
108 -----
109 -----
    
```

Ak bolo na vstupe číslo rovné 3, program skočil na riadok 65. **Pohyb dole = Y**

Ak je Y rovné 5, tak skočí na riadok 76, odpočíta sa 4 a pozícia je rovná 1. (žlté políčka)

Ak je Y rovné 1, tak sa pozrie na hodnotu X, ak je rovná 1 (červené políčko), tak skáče na riadok 81, pridá sa 1. Inak sa jedná o modré políčka a program končí inštrukciou OHR.

Ak nič z tohto nrapliti, pripočíta sa 1.

	1	2	3	4	5
1				•	
2			•	V	•
3				•	
4				•	
5				•	

Ak bolo na vstupe číslo rovné 4, program skočí na riadok 85. **Pohyb vľavo = X**

Ak je X rovné 5, program skončí inštrukciou OHR. (modré políčka)

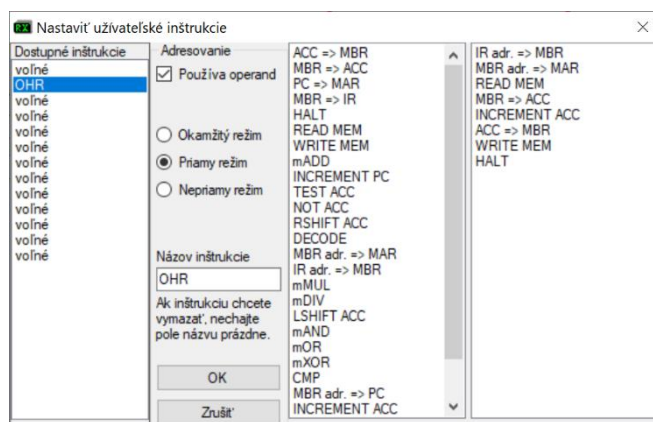
Ak je X rovné 1 a zároveň Y rovné 2, program skončí inštrukciou OHR. (červené políčko). Ak je X rovné 1 a Y nie je rovné 2, tak sa pripočíta 4. (žlté políčka).

	1	2	3	4	5
1				•	
2			•	V	•
3				•	
4				•	
5				•	

Inštrukcia OHR - Ohrozenie:

Inštrukcia OHR má za úlohu robiť dve veci.

1. Podľa zadania pred ukončením programu vloží do pamäte na riadku 302 hodnotu 1 ako znak predčasného ukončenia programu z dôvodu ohrozenia.
2. Ukončí program.



Vytvorenie inštrukcie:

Používa operand aby bola variabilnejšia z prípadu iného budúceho využitia, inak by to bolo „zamknuté“ na pamäť [302].

Pamäť ktorá bola zadaná pri volaní inštrukcie sa inkrementuje, tak z 0 prejde na 1 a zadanie je splnené. Potom sa uloží na rovnakú pamäť.

Roland Vdovják, ID: 110912
piatok, 20. novembra 2020 17:29

Simulácia:

Daná je postupnosť: 1 2 2 1 1 4 4 4 3 3 3 2 0

Program túto postupnosť vykoná a pôjde podľa tabuľky zo začiatočného políčka s písmenom „Z“ a dostane po zadanej postupnosti späť na toto políčko.

Na pamäti 303 je ukazovateľ na 305, tento ukazovateľ sa postupne inkrementuje.

Pri zmene nejakého kľučku vedeného ku ohrozeniu sa program ukončí a v pamäti 302 sa zapíše 1 alebo zostane 0.

Príklad: 1 2 2 1 1 4 4 4 3 3 3 2 0

Prvá 4 sa zmení na jednotku.

	1	2	3	4	5
1	↑ Z			•	→
2		■	•	V	•
3	←	←	←	•	↓
4			↑	•	↓
5	→	→	↑	•	↓

299	-----
300	1
301	1
302	0
303	305
304	-----
305	1
306	2
307	2
308	1
309	1
310	4
311	4
312	4
313	3
314	3
315	3
316	2
317	0
318	-----
319	-----
320	-----