

## Riešenie 4. zadania

### UPRATOVACÍ ROBOT

#### Zadanie:

Napíšte program, ktorý bude simulovať pohyb upratovacieho robota po miestnosti reprezentovanej ako pole podľa obrázka. Ľavé horné políčko má súradnice (riadok, stĺpec) = (1,1) a pravé spodné políčko má súradnice (5,5). Robot môže začínať na ľubovoľnom políčku, okrem prekážok a môže vykonávať kroky o 1 políčko smerom na niektorú svetovú stranu. Políčko, ktoré robot navštíví, je následne považované za upratané (vrátane štartovacieho). Políčko s prekážkou sa vždy počíta ako neupratané. Na niektorých políčkach už robot bol (viď. obrázok), avšak môže na nich opätovne vstúpiť aj viac krát.

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

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

Hodnota 0 reprezentuje ukončenie postupnosti. Môžete predpokladať, že robot sa nikdy nepokúsi vstúpiť na políčko s prekážkou, ani sa nepokúsi opustiť miestnosť. Zistíte, či po vykonaní zadanej postupnosti krokov robot upratol aspoň zadanú časť miestnosti (viď. konkrétne úlohy). Ak áno, na adresu 302 uložte hodnotu 1. Ak nie, na adresu 302 uložte hodnotu 0. Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu robota uchovávalte na adresách 300 resp. 301.

#### Úloha 54

	1	2	3	4	5
1	✓	✓	✓	✓	✓
2	✓	■		■	
3	✓				
4		■		■	
5					

■ - prekážka, ✓ - už upratané  
cieľ: 14 políčk

## Riešenie

### Pamäť programu

ADRESA	INŠTRUKCIA	KOMENTÁR
0	LOAD[mem(303)]	načítaj prvok postupnosti, ukazovateľ je uložený na adrese 303
1	CMP 1	porovnáva načítaný prvok s jednotkou
2	JZ 20	ak áno, skočí na <b>podprogram pre pohyb hráča hore</b> na adrese 20
3	CMP 2	porovnáva načítaný prvok s dvojkou
4	JZ 30	ak áno, skočí na <b>podprogram pre pohyb hráča doprava</b> na adrese 30
5	CMP 3	porovnáva načítaný prvok s trojkou
6	JZ 40	ak áno, skočí na <b>podprogram pre pohyb hráča dole</b> na adrese 40
7	CMP 4	porovnáva načítaný prvok s štvorkou
8	JZ 50	ak áno, skočí na <b>podprogram pre pohyb hráča doľava</b> na adrese 50
9	HALT	ak nie je splnené ani jedno porovnávanie, ukončí sa program
...	...	...
20	DEC [301]	zniž hodnotu na adrese 301 v pamäti údajov (y--; pohyb hore)
21	JMP 99	skočí na adresu 99( <b>podprogram na kontrolovanie, či je políčko upratané</b> )
...	...	...
30	INC[300]	pohyb doprava
31	JMP 99	-//-
...	...	...
40	INC[301]	pohyb dole
41	JMP 99	-//-
...	...	...
50	DEC[300]	pohyb doľava
51	JMP 99	-//-
...	...	...
99	INC [303]	zvýši hodnotu ukazovateľa ukazujúceho na prvok postupnosti
100	LOAD [mem(304)]	načíta hodnotu na ukazovateli, ktorá je uložená na adrese 304 (tam začínajú súradnice uprataných bodov)
101	CMP [300]	porovná s hodnotou uloženou na adrese 300 (porovná x-sovú súradnicu aktuálnej pozície robota s x-sovou súradnicou uprataného políčka)
102	JZ 115	ak sa zhodujú, skočí na <b>podprogram, ktorý kontroluje, či aj ypsilonové súradnice robota a uprataného bodu sa zhodujú</b>
103	LOAD [mem(304)]	načíta znova (tú istú) hodnotu na ukazovateli, ktorá je uložená na adrese 304, pre potrebu porovnávania
104	CMP 0	Porovnanie s 0, keď príde na koniec zoznamu uložených (uprataných) bodov nájde 0, čiže ho bude treba uložiť (bod nebol uprataný, ale už bude)
105	JZ 130	skočí na <b>podprogram na pridávanie bodu medzi uložené body</b> (upratané)
106	DINC [304]	double increment - vlastná inštrukcia zvýši hodnotu na adrese 304 (posúvanie sa po x-sových súradniciach uložených bodov) o 2

107	JMP 100	prechod na ďalšiu iteráciu cyklu
...	...	...
115	INC [304]	zvyšuje hodnotu na adrese 304 (prejde z adresy x na adresu y daného bodu)
116	LOAD [mem(304)]	načíta hodnotu na ukazovateli, ktorá je uložená na adrese 304 (y hodnota bodu)
117	CMP [301]	porovná s hodnotou uloženou na adrese 301 (y hodnota robota)
118	JZ 125	ak sa zhodujú, skočí na <b>podprogram na resetovanie hodnoty na adrese 304</b> (naspäť na začiatok zoznamu uprataných políček)
119	INC [304]	zvyšuje hodnotu na adrese 304 (prejde z adresy y na adresu x ďalšieho bodu)
120	JMP 100	skočí na adresu 100 (na podprogram na kontrolovanie, či je políčko upratané)
...	...	...
125	LOAD [299]	načítanie hodnoty na adrese 299 kde je uložená adresa začiatku pamäťového poľa obsahujúceho upratané body (400)
126	STORE [304]	vlož načítanú hodnotu na adresu 304 (lebo hodnota 400 na adrese 304 sa inkrementovala, a chceme ju nastaviť naspäť na 400)
127	JMP 0	vrátenie sa na začiatok programu
...	...	...
130	LOAD [300]	načíta hodnotu na adrese 300 (x-ová súradnica robota)
131	STORE [mem(304)]	uloží ju na adresu, na ktorú ukazuje ukazovateľ uložený na adrese 304 (adresa nuly)
132	INC [304]	inkrementuje hodnotu na adrese 304 (lebo pridávame bod – zatiaľ x-ová súradnica nového uprataného bodu)
133	LOAD [301]	načíta hodnotu na adrese 301 (y súradnica robota)
134	STORE [mem(304)]	uloží ju na adresu, na ktorú ukazuje ukazovateľ uložený na adrese 304
135	INC [304]	inkrementuje hodnotu na adrese 304 (lebo pridávame bod – y súradnica nového uprataného bodu)
136	LOAD [290]	načíta nulu, ktorou chceme ukončiť postupnosť uprataných bodov
137	STORE [mem(304)]	uloží nulu na adresu na ktorú ukazuje ukazovateľ, ktorý je uložený na adrese 304
138	LOAD [299]	načítanie hodnoty na adrese 299 kde je uložená adresa začiatku pamäťového poľa obsahujúceho upratané body (400)
139	STORE [304]	vlož načítanú hodnotu na adresu 304 (lebo hodnota 400 na adrese 304 sa inkrementovala, a chceme ju nastaviť naspäť na 400)
140	INC [285]	inkrementuje počet nájdených bodov (uprataných) na adrese 285
141	CMP 14	kontroluje či sa počet uprataných bodov zhoduje s minimálnym počtom uprataných bodov
142	JZ 150	ak ano skoč na podprogram zapisujúci hodnotu 1 na adresu 302
143	JMP 0	ak nie vrat sa na začiatok programu
...	...	...
150	LOAD [289]	načítaj číslo 1 do registra MBR

151	STORE[302]	vlož obsah registra MBR (číslo 1 načítane z adresy 289) na adresu 302
152	JMP 0	Vráť sa na začiatok programu

### Pamäť údajov pred spustením programu

ADRESA	ÚDAJ	KOMENTÁR
285	?	počet vopred uprataných políčok
...	...	...
289	1	pomocná jednotka
290	0	pomocná nula
...	...	...
299	400	adresa začiatku zoznamu súradníc uprataných bodov (statická)
300	?	začiatočná stĺpcová súradnica hráča - x
301	?	začiatočná riadková súradnica hráča - y
...	...	...
303	305	ukazovateľ na momentálne spracúvaný prvok postupnosti
304	400	ukazovateľ na zoznam súradníc uprataných bodov (dynamická)
305	?	postupnosť krokov
306	?	
307	?	
308	?	
309	?	
310	?	
311	?	
312	?	
313	?	
314	?	
315	?	
316	?	
317	?	
318	?	
319	?	
320	?	
321	0	koniec postupnosti
...	...	...
400	$X_1$	súradnice uložených (uprataných) bodov
401	$Y_1$	
402	$X_2$	
403	$Y_2$	
404	$X_3$	
405	$Y_3$	
406	$X_4$	
407	$Y_4$	
408	$X_5$	
409	$Y_5$	
410	0	Koniec zoznamu

### Pamäť údajov po skončení programu

ADRESA	ÚDAJ	KOMENTÁR
285	?	konečný počet uprataných bodov
...	...	...
300	5	konečná stĺpcová súradnica hráča
301	1	konečná riadková súradnica hráča
303	321	ukazovateľ na posledný prvok postupnosti
...	...	...
{400; ([280]+1)}		pamäťové pole, kt. drží súradnice bodov uprataných robotom ako postupne prechádzal postupnosťou inštrukcií pridával neznámy počet bodov podľa toho, koľko ich našiel čiže neviem presne definovať pamäťový rozsah , platí že je to rozsah od adresy 400 až po súradnicu 400+hodnota na adrese 285 +1

Ostatné hodnoty v pamäti údajov ostanú bez zmeny.

## Zhodnotenie

Na riešenie zadania sme využili 9 rozdielnych inštrukcií (nie len predprogramovaných) a niekoľko cyklov – celkovo 53 inštrukcií. Pre optimalizovanie programu sme vytvorili vlastnú funkciu DINC (double increment).

INŠTRUKCIA	POČET POUŽITÍ
LOAD	10
CMP	8
JZ	8
HALT	1
DEC	2
JMP	9
INC	8
STORE	6
DINC	1

Veľkosť zabranej pamäte údajov je 33, čo sa môže líšiť podľa aktuálnej série pokynov pre program.