

## Úloha 23 – Artur Kozubov

|   | 1 | 2 | 3 |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   |   |   |
| 3 |   |   |   |
| 4 |   |   |   |
| 5 |   |   |   |
| 6 |   |   |   |
| 7 |   |   |   |
| 8 |   |   |   |

### Zadanie

Napište 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.

V pamäti údajov (PÚ) uchovávalte riadkovú súradnicu hráča na adrese **a0h** a stĺpcovú na adrese **b0h**. Od adresy **0h** so 4-bajtovými rozostupmi (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.

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

### Riešenie

#### Nápad

Ide o to, aby sa do každej bunky zapísalo desiatinné číslo, ktoré v binárnom kóde bude reprezentovať túto logiku: (**HORE.VPRAVO.DOLE.VĽAVO**). každá cifra bude uchovávať pamäť o tom, či v danom smere existuje stena (**1**) alebo nie (**0**). Potom to všetko prevedieme do desiatkovej sústavy a zapíšeme do registra (**64h + X.Y**). Šifrovaná mapa je zobrazená nižšie

#### Príklad kódovania poľa

|   | 1    | 2    | 3    |
|---|------|------|------|
| 1 | 1001 | 1000 | 1100 |
| 2 | 0001 | 0000 | 0100 |
| 3 | 0001 | 0000 | 0100 |
| 4 | 0001 | 0000 | 0100 |
| 5 | 0001 | 0000 | 0100 |
| 6 | 0011 | 0110 | 0101 |
| 7 | 1000 | 1100 | 0101 |
| 8 | 0011 | 0010 | 0110 |

⇒

| x<br>y | 1 | 2  | 3  |
|--------|---|----|----|
| 1      | 9 | 8  | 12 |
| 2      | 1 | 0  | 4  |
| 3      | 1 | 0  | 4  |
| 4      | 1 | 0  | 4  |
| 5      | 1 | 0  | 4  |
| 6      | 3 | 6  | 5  |
| 7      | 8 | 12 | 5  |
| 8      | 3 | 2  | 6  |

## Prečo?

Jednoduché zistenie, či sa v určitej koordináte nachádza stena. Napríklad sa nachádzame v bode (1h, 1h) (**1001**), ak chceme zistiť, či je hore stena, použijeme **AND** 8h (**1000**): 1001 AND 1000 = 1h - stena existuje.

## Pamäť programu

| Adresa | Label    | Inštrukcia          | Komentár  |
|--------|----------|---------------------|---|
| 0h     | LOADX    | LW \$11,0070(\$0)   | načítame začiatočnú riadkovú <b>X</b> súradnicu z PÚ z adresy 70h do registra R11   |
| 4h     | LOADY    | LW \$12,0080(\$0)   | načítame začiatočnú stĺpcovú <b>Y</b> súradnicu z PÚ z adresy 80h do registra R12   |
| 8h     | LOADV    | LW \$14,0000(\$15)  | do registra R14 načítame prvok postupnosti z PÚ z adresy, na ktorú ukazuje ukazovateľ v registri R15 (Op je počiatočná)   |
| ch     | INCS     | ADDI \$15,\$15,0004 | zvážšime ukazovateľ v registri R15 o 4h, aby ukazoval na ďalší prvok postupnosti v poradí   |
| 10h    | CHECKING | BEQ \$11,\$0,WIN    | kontrola súradníc (na výstup z labyrintu), či súradnica nie je mimo hraníc mapy 3 x 9<br><br>ak áno - skok na WIN, aby sa zapísalo (exit found) a potom nasleduje HALT, ktorý zapíše konečné súradnice späť do pamäte |
| 14h    |          | BEQ \$12,\$0,WIN    |   |
| 18h    |          | BEQ \$11,\$4,WIN    |   |
| 1ch    |          | BEQ \$12,\$9,WIN    |   |
| 20h    | POINTER  | MUL \$13,\$11,\$10  | vytvorenie ukazovateľa pomocou súradníc ( <b>64h + X.Y</b> ), uloženie a prečítanie kódovanej bunky na registra \$13  |
| 24h    |          | NOP                 |   |
| 28h    |          | NOP                 |   |
| 2ch    |          | ADD \$13,\$13,\$12  |   |
| 30h    |          | NOP                 |   |
| 34h    |          | NOP                 |   |
| 38h    |          | MUL \$13,\$13,\$4   |   |
| 3ch    |          | NOP                 |   |
| 40h    |          | NOP                 |   |
| 44h    |          | LW \$13,0064(\$13)  |   |
| 48h    | MOVING   | BEQ \$14,\$1,UP     | ak je načítaný prvok postupnosti v reg. R14 rovný 1 (konštantu 1 máme uloženú v reg. R1) skoč na podprogram pre vykonanie pohybu hore ktorý sa nachádza na libely „UP“  |
| 4ch    |          | BEQ \$14,\$2,RIGHT  | ak je načítaný prvok postupnosti v reg. R14 rovný 2 (konštantu 2 máme uloženú v reg. R2) skoč na podprogram pre vykonanie pohybu vpravo ktorý sa nachádza na libely „RIGHT“   |
| 50h    |          | BEQ \$14,\$3,DOWN   | ak je načítaný prvok postupnosti v reg. R14 rovný 3 (konštantu 3 máme uloženú v reg. R3) skoč na podprogram pre vykonanie pohybu dole ktorý sa nachádza na libely „DOWN“  |
| 54h    |          | BEQ \$14,\$4,LEFT   | 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 libely „LEFT“   |
| 58h    |          | BEQ \$0,\$0,HALT    | a skočíme niekam na   |
| ...    |          |                     |   |

|      |       |                     |  |
|------|-------|---------------------|--|
| 70h  | UP    | ANDI \$13,\$13,0008 | uložiť výsledok, či je hore stena ?                              |
| 74h  |       | NOP                 |  |
| 78h  |       | NOP                 |  |
| 7ch  |       | BNEQ \$13,\$0,LOADV | ak áno, nemôžeme sa už pohnúť hore, takže ideme na ďalší prvok   |
| 80h  |       | NOP                 |  |
| 84h  |       | NOP                 |  |
| 88h  |       | SUBI \$12,\$12,0001 | ak nie, posuň sa hore (zmenši riadkovú súradnicu o 1)            |
| 8ch  |       | BEQ \$0,\$0,LOADV   | a ideme na ďalší prvok   |
| ...  |       |                     |  |
| a0h  | RIGHT | ANDI \$13,\$13,0004 | uložiť výsledok, či je vpravo stena?                             |
| a4h  |       | NOP                 |  |
| a8h  |       | NOP                 |  |
| ach  |       | BNEQ \$13,\$0,LOADV | ak áno, nemôžeme sa už pohnúť vpravo, takže ideme na ďalší prvok |
| b0h  |       | NOP                 |  |
| b4h  |       | NOP                 |  |
| b8h  |       | ADDI \$11,\$11,0001 | ak nie, posuň sa vpravo (zväčši stĺpcovú súradnicu o 1)          |
| bch  |       | BEQ \$0,\$0,LOADV   | a ideme na ďalší prvok   |
| ...  |       |                     |  |
| d0h  | DOWN  | ANDI \$13,\$13,0002 | uložiť výsledok, či je dole stena ?                              |
| d4h  |       | NOP                 |  |
| d8h  |       | NOP                 |  |
| dch  |       | BNEQ \$13,\$0,LOADV | ak áno, nemôžeme sa už pohnúť dole, takže ideme na ďalší prvok   |
| e0h  |       | NOP                 |  |
| e4h  |       | NOP                 |  |
| e8h  |       | ADDI \$12,\$12,0001 | ak nie, posuň sa dole (zväčši riadkovú súradnicu o 1)            |
| ech  |       | BEQ \$0,\$0,LOADV   | a ideme na ďalší prvok   |
| ...  |       |                     |  |
| 100h | LEFT  | ANDI \$13,\$13,0001 | uložiť výsledok, či je vľavo stena ?                             |
| 104h |       | NOP                 |  |
| 108h |       | NOP                 |  |
| 10ch |       | BNEQ \$13,\$0,LOADV | ak áno, nemôžeme sa už pohnúť vľavo, takže ideme na ďalší prvok  |
| 110h |       | NOP                 |  |
| 114h |       | NOP                 |  |
| 118h |       | SUBI \$11,\$11,0001 | ak nie, posuň sa vľavo (zmenši stĺpcovú súradnicu o 1)           |
| 11ch |       | BEQ \$0,\$0,LOADV   | a ideme na ďalší prvok   |
| ...  |       |                     |  |
| 140h | WIN   | SW \$1,0060(\$0)    |  |
| 144h | HALT  | SW \$11,0070(\$0)   |  |
| 148h |       | SW \$12,0080(\$0)   |  |

## Register

| Register | Value    |
|----------|----------|
| R0       | 00000000 |
| R1       | 00000001 |
| R2       | 00000002 |
| R3       | 00000003 |
| R4       | 00000004 |
| R5       | 00000000 |
| R6       | 00000000 |
| R7       | 00000000 |
| R8       | 00000000 |
| R9       | 00000009 |
| R10      | 0000000a |
| R11      | 00000000 |
| R12      | 00000000 |
| R13      | 00000000 |
| R14      | 00000000 |
| R15      | 00000000 |
| R16      | 00000000 |
| R17      | 00000000 |
| R18      | 00000000 |
| R19      | 00000000 |
| R20      | 00000000 |
| R21      | 00000000 |
| R22      | 00000000 |
| R23      | 00000000 |
| R24      | 00000000 |
| R25      | 00000000 |
| R26      | 00000000 |
| R27      | 00000000 |
| R28      | 00000000 |
| R29      | 00000000 |
| R30      | 00000000 |
| R31      | 00000000 |

| Register    | Údaj      | Komentár  |
|-------------|-----------|---|
| R1 - R4, R9 | 1h-4h, 9h | pre porovnanie  |
| R10         | ah        | na vytvorenie ukazovateľa   |
| R11, R12    | ...       | X, Y  |
| R14         |           | sem sa bude načítavať prvok postupnosti z pamäte údajov           |
| R15         |           | ukazovateľ do postupnosti prvkov, na začiatku ukazuje na 1. prvok |

## Data memory

| address  | data memory |          |          |          |
|----------|-------------|----------|----------|----------|
| 00000000 | 00000001    | 00000002 | 00000002 | 00000002 |
| 00000010 | 00000002    | 00000003 | 00000003 | 00000003 |
| 00000020 | 00000003    | 00000003 | 00000003 | 00000003 |
| 00000030 | 00000003    | 00000003 | 00000001 | 00000003 |
| 00000040 | 00000004    | 00000004 | 00000001 | 00000004 |
| 00000050 | 00000000    | 00000000 | 00000000 | 00000000 |
| 00000060 | 00000000    | 00000000 | 00000000 | 00000000 |
| 00000070 | 00000001    | 00000000 | 00000000 | 00000000 |
| 00000080 | 00000001    | 00000000 | 00000000 | 00000000 |
| 00000090 | 00000009    | 00000001 | 00000001 | 00000001 |
| 000000a0 | 00000001    | 00000003 | 00000008 | 00000003 |
| 000000b0 | 00000000    | 00000000 | 00000008 | 00000000 |
| 000000c0 | 00000000    | 00000000 | 00000000 | 00000006 |
| 000000d0 | 0000000c    | 00000002 | 00000000 | 00000000 |
| 000000e0 | 0000000c    | 00000004 | 00000004 | 00000004 |
| 000000f0 | 00000004    | 00000005 | 00000005 | 00000006 |
| 00000100 | 00000000    | 00000000 | 00000000 | 00000000 |
| 00000110 | 00000000    | 00000000 | 00000000 | 00000000 |

| Register  | Komentár                                   |
|-----------|--|
| 0h – 5ch  | postupnosť krokov                          |
| 6h        | výsledok, exit alebo nie na konci programu |
| 70h, 80h  | X, Y                                       |
| 90h – fch | ZAŠIFROVANÁ KARTA                          |

Nasimulujeme postupnosť

1.

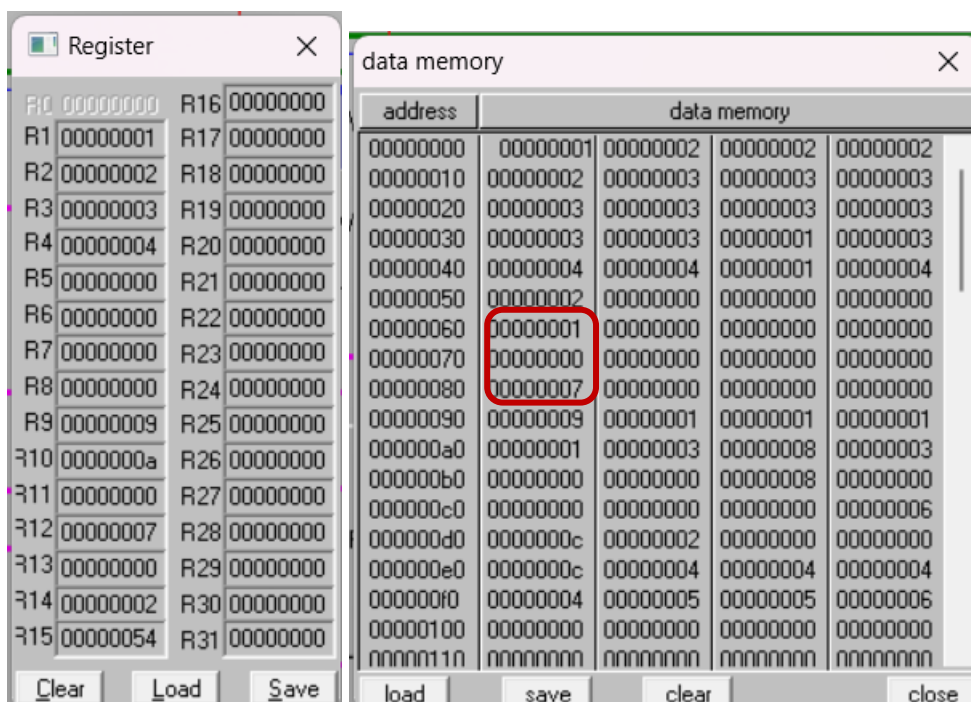
Počiatkový bod: ( 1, 1 ) ( **X**, **Y** )

Postupnosť: 1 2 2 2 2 3 3 3 3 3 3 3 3 1 3 4 4 1 4 2 0

*Pamäť údajov pred spustením programu*

**\*PRESNE ROVNAKÉ AKO PÔVODNÉ (VYŠŠIE)**

Pamäť údajov po skončení programu



Tak sa našlo východisko a výsledkom je jeden.

## Záver

Tento typ šifrovania sa dá použiť aj vtedy, keď máme k dispozícii viac možností pohybu, napríklad súradnicu Z, a je tiež vhodné a jednoduché šifrovať a dešifrovať informácie, ktoré potrebujeme.

Tento spôsob zápisu čísel do registra ([offset].X.Y) - nás obmedzuje vo veľkosti mapy 9 na 9, pretože nie je možné zapisovať do registra ([offset].X.12), pretože súradnica musí byť jednociferná.