**CROSSWORD PUZZLE –III**

Designed by Pratibha Pai

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** |  |  | **3** |  | **4** |  |  | **5** |
|  |  |  |  |  | **6** |  |  |  |  |
|  |  |  | **7** |  |  |  | **8** |  |  |
| **9** |  |  |  |  |  |  |  |  |  |
|  |  |  | **10** |  |  |  | **11** |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **12** | **13** |  |  |  | **14** |  | **15** |  |  |
|  |  |  |  |  |  |  |  | **16** |  |
| **17** |  | **18** |  |  |  | **19** |  |  |  |
| **20** |  |  | **21** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Puzzle III MENU |  |  |  |  | |  |
|  |  |  |  |  |  | |  |
|  | **Across** |  | **Down** |  |  | |  |
| 1 | Move data from L register to A register(5) | 2 | Logically OR accumulator content with any register content(3) | | | | |
| 4 | Put the sum of accumulator content and L register content into accumulator(4) | 3 | Load HL pair direct | | | | |
| 7 | Rotate accumulator content left without carry(3) | 5 | Set stack pointer to memory(5) | | | | |
| 8 | Compare immediately 8 bit data with accumulator(3) | 6 | Add with carry immediately(3) | |  |  | |
| 10 | **Backwards** – compare A content with D content(4) | 9 | Load accumulator with data from memory(3) | | | | |
| 12 | Add E register with A register(4) | 11 | Set carry flag(3) | |  |  | |
| 14 | Complement carry flag(3) | 13 | Disable interrupt(2) | | | | |
| 15 | Complement accumulator(3) | 14 | Compare A content with memory location content(4) | | | | |
| 18 | Send data from accumulator to output port(3) | 16 | Store accumulator content in register pair(4) | |  |  | |
| 20 | No operation(3) | 17 | Jump if zero flag is reset(3) | |  |  | |
|  |  | 19 | Logically OR immediately(3) | |  |  | |
|  | 21 | **Upwards** – Subtract E register content from A register(4) | | | | |

Solution at the end

**CROSSWORD PUZZLE –IV**

Designed by Pratibha Pai

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** |  | **3** |  |  | **4** |  |  |
|  |  | **5** |  |  |  |  |  |  |  |  |
|  | **6** |  |  |  |  |  |  |  |  |  |
|  |  |  |  | **7** |  |  |  | **8** |  |  |
|  | **9** |  |  |  |  |  |  | **10** |  |  |
|  |  |  | **11** |  |  |  | **12** |  | **13** |  |
| **14** |  |  |  |  |  |  |  |  |  |  |
|  |  | **15** |  | **16** |  |  |  | **17** |  |  |
|  | **18** |  |  |  | **19** |  |  |  |  |  |
|  |  |  |  | **20** |  |  |  |  |  |  |
|  | **21** | **22** |  |  |  |  | **23** | **24** |  | **25** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Puzzle IV MENU |  |  |  |  |  |
|  | **Across** |  | **Down** |  |  |  |
| 1 | Copy data from E register to accumulator(5) | 2 | Logically OR immediately(3) | | | |
| 4 | Complement accumulator(3) | 3 | Double the data in accumulator using arithmetic instruction(3) | | | |
| 5 | **Backwards**- Send data from accumulator to output port(3) | 4 | Compare C register content with accumulator(4) | |  |  |
| 6 | Move 8 bit data immediately to C register(4) | 10 | Store HL pair direct(4) | | | |
| 7 | Decimal adjust addition(3) | 13 | Load 8bit data from memory to accumulator(3) | |  |  |
| 8 | Complement carry flag(3) | 16 | See 23 across | |  |  |
| 9 | **Backwards**- Jump if data is positive(2) | 17 | Disable interrupt(2) | | | |
| 9 | Jump if data is negative(2) | 19 | Jump if no carry is generated(3) | |  |  |
| 11 | Do not perform any operation(3) | 22 | **Upwards-** Send 8 bit data from accumulator to output port(3) | |  |  |
| 12 | Load HL pair direct(4) | 24 | ***Upwards****- Logically XOR immediately 8bit data(3)* | |  |  |
| 14 | Jump if zero flag is set(2) | 25 | **Upwards-** see 7 Across | | | |
| 15 | Rotate accumulator content left with carry(3) |  |  | | | |
| 17 | 16 bit addition of HL |  | | | |
| 18 | Store accumulator content to address stored in pair of registers(4) |  | | | |
| 20 | Load accumulator 8 bit data from a port(2) |  | | | |
| 21 | Transfer data from C register to D register(5) |  | | | |
| 23 | Use DE pair register as pointer(4) |  | | | |

**PUZZLE –III PUZZLE –IV**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1**  **M** | **2**  **O** | **V** | **A** | **3**  **L** |  | **4**  **A** | **D** | **D** | **5**  **L** |
|  | **R** |  |  | **H** | **6**  **A** |  |  |  | **X** |
|  | **A** |  | **7**  **R** | **L** | **C** |  | **8**  **C** | **P** | **I** |
| **9**  **L** |  |  | **R** | **D** | **I** |  |  |  | **S** |
| **D** | **P** | **M** | **10**  **C** |  |  |  | **11**  **S** |  | **P** |
| **A** |  |  |  |  |  |  | **T** |  |  |
| **12**  **A** | **13**  **D** | **D** | **E** |  | **14**  **C** | **M** | **15**  **C** | **M** | **A** |
|  | **I** |  | **B** |  | **M** |  |  | **16**  **S** |  |
| **17**  **J** |  | **18**  **O** | **U** | **T** | **P** | **19**  **O** | **R** | **T** |  |
| **20**  **N** | **O** | **P** | **21**  **S** |  | **M** | **R** |  | **A** |  |
| **Z** |  |  |  |  |  | **I** | **R** | **X** |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1**  **M** | **2**  **O** | **V** | **3**  **A** | **E** |  | **4**  **C** | **M** | **A** |
| **T** | **U** | **5**  **O** | **R** |  | **D** |  |  | **M** |  |  |
|  | **6**  **M** | **V** | **I** | **C** | **D** |  |  | **P** |  |  |
|  |  |  |  | **7**  **D** | **A** | **A** |  | **8**  **C** | **M** | **C** |
| **P** | **9**  **J** | **M** |  |  |  |  |  | **10**  **S** |  |  |
|  | **N** |  | **11**  **N** | **O** | **P** |  | **12**  **L** | **H** | **13**  **L** | **D** |
| **14**  **J** | **Z** |  |  |  |  |  |  | **L** | **D** |  |
|  |  | **15**  **R** | **A** | **16**  **L** |  |  |  | **17**  **D** | **A** | **D** |
|  | **18**  **S** | **T** | **A** | **X** | **19**  **J** |  |  | **I** |  | **A** |
|  |  | **U** |  | **20**  **I** | **N** |  |  | **R** |  | **A** |
|  | **21**  **M** | **22**  **O** | **V** | **D** | **C** |  | **23**  **L** | **24**  **X** | **I** | **25**  **D** |