1. Exchange of two 16 bit numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | B,02 | Setup B as a byte counter. To exchange two 16 bit numbers b is loaded with 02 |
|  | LXI | H,8050 | Setup HL as a memory pointer 8050 memory location |
|  | LXI | D,8070 | Setup DE as a memory pointer 8070 memory location |
| NEXT | LDAX | D | Move the contents of memory pointer specified by the DE register pair to accumulator |
|  | MOV | C,A | Save the contents of accumulator in C register |
|  | MOV | A,M | Move the data byte of memory pointer specified by the HL register pair to accumulator |
|  | STAX | D | Transfer the data byte of HL register pair memory pointer to DE register pair memory pointer |
|  | MOV | M,C | Transfer the data byte of DE register pair memory pointer to HL register pair memory pointer |
|  | INX | H | Increment the source memory pointer to get the next data byte |
|  | INX | D | Increment the destination memory pointer to get the next data byte |
|  | DCR | B | Decrement the counter by 1, if one data byte is exchanged |
|  | JNZ | NEXT | If counter is not zero, go back to NEXT to exchange next data byte |
|  | HLT |  | Terminate program execution |

Exchange the data bytes 4535H and B6ACH

|  |  |  |  |
| --- | --- | --- | --- |
| Input(Before execution) | | Output(after execution) | |
| Source memory | Destination memory | Source memory | Destination memory |
| 8050 – 35H | 8070 – ACH | 8050 – ACH | 8070 – 35H |
| 8051 – 45H | 8071 – B6H | 8051 – B6H | 8071 – 45H |

1. (a) addition of two 8 bit hex numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Initialize the HL register pair as memory pointer |
|  | MVI | C,00 | Clear C register to store carry |
|  | MOV | A,M | Move the first data byte (2FH) to accumulator |
|  | INX | H | Increment the memory pointer by one (8501) to store second data byte (45H) |
|  | ADD | M | Add the content of [A]+[M]=[A] |
|  | JNC | NEXT | If carry means, go to label NEXT |
|  | INR | C | If no carry means, increment the c register by one |
| NEXT | STA | 8502 | Store the sum in 8502 |
|  | MOV | A,C | Move carry status to accumulator |
|  | STA | 8503 | Store the carry in 8503 |
|  | HLT |  | Terminate program execution |

|  |  |  |
| --- | --- | --- |
| Input | Output | Explanation |
| 8500 – 2FH(data1) | 8502 – 74H(sum) | 2FH=001011112  45H=010001012  74H=011101002 |
| 8501 – 45H(data2) | 8503 – 00H(carry) |

(b) subtraction of two 8 bit hex numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Initialize the HL register pair as memory pointer |
|  | MVI | C,00 | Clear C register to store carry |
|  | MOV | A,M | Move the first data byte (minuend) to accumulator |
|  | INX | H | Increment the memory pointer to load subtrahend |
|  | SUB | M | Subtract the subtrahend from minuend |
|  | JNC | NEXT | If carry means, go to label NEXT |
|  | INR | C | If no carry means, increment the c register by one |
| NEXT | STA | 8502 | Store the difference in 8502 |
|  | MOV | A,C | Move carry status to accumulator |
|  | STA | 8503 | Store the carry in 8503 |
|  | HLT |  | Terminate program execution |

Subtract 4CH from 2AH

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 4CH(minuend) | 8502 – 22H(difference) |
| 8501 – 2AH(subtrahend) | 8503 – 00H(borrow) |

1. Addition of two ‘N’ byte numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,04 | Setup C as a byte counter. To –byte two 4-bytes of data, register c is loaded with 04H |
|  | LXI | H,8500 | Setup HL as a memory pointer with 8500 memory location to store 1st byte of first number |
|  | LXI | D,8600 | Setup DE as a memory pointer with 8600 memory location to store 1st byte of second number |
|  | XRA | A | Clear accumulator |
| NEXT | LDAX | D | Get the data byte of 2nd number to accumulator |
|  | ADC | M | Add 1st and 2nd number with carry |
|  | MOV | M,A | Save the sum in memory pointer specified by HL register pair |
|  | INX | H | Increment the memory pointer 8500 by 1 to store the next data byte of first number |
|  | INX | D | Increment the memory pointer 8600 by 1 to store the next data byte of second number |
|  | DCR | C | Decrement the counter by 1, if one addition is completed |
|  | JNZ | NEXT | If counter is not zero, go back to NEXT to add next data byte |
|  | MVI | A,00 | Clear accumulator to save carry |
|  | RAL |  | Rotate the contents of accumulator through carry to move the generated carry during addition to accumulator |
|  | MOV | M,A | Save carry in 8504 memory location |
|  | HLT |  | Terminate program execution |

Add B9681429H and 65554535H

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| Memory location to store 1st 4-byte numbers | Memory location to store 2nd 4-byte numbers | Memory location to store the result of two 4-byte numbers and carry |
| 8500 – 35H | 8600 – 29H | 8500 – 5EH |
| 8501 – 45H | 8601 – 14H | 8501 – 59H |
| 8502 – 55H | 8602 – 68H | 8502 – BDH |
| 8503 – 65H | 8603 – B9H | 8503 – 1EH |
|  |  | 8504 – 01H(carry) |

1. ‘N’ decimal number addition

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,0A | Setup C register with a counter value OA to add 10 BCD numbers |
|  | XRA | A | Clear accumulator |
|  | MOV | B,A | Clear B register to save generated carry |
|  | LXI | H,8050 | Setup HL register pair as memory pointer which memory address 8050 to store first data bytes |
| LOOP | ADD | M | Data byte is added to accumulator |
|  | DAA |  | Adjust the sum for decimal value |
|  | JNC | NEXT | If no carry go to NEXT |
|  | INR | B | If carry Increment B register by 1 |
| NEXT | INR | H | Increments the memory pointer to store the next data byte for addition |
|  | DCR | C | Decrement the counter by 1, if one addition is completed |
|  | JNZ | LOOP | If counter is not zero, go to LOOP to add next data byte |
|  | STA | 8600 | Stores the sum in 8600 memory location |
|  | MOV | A,B | Moves the contents of B register (carry contents) to accumulator |
|  | STA | 8601 | Store the carry in 8601 memory location |
|  | HLT |  | Terminate program execution |

Add the BCD numbers 25,31,64,80,15,42,05,76,98,56. The result is 496

|  |  |  |
| --- | --- | --- |
| Input | Output | |
| 8050 – 25 | | 8600 – 96(sum)  8601 – 04 (carry) |
| 8051 – 31 | |
| 8052 – 64 | |
| 8053 – 80 | |
| 8054 – 15 | |
| 8055 – 42 | |
| 8056 – 05 | |
| 8057 – 76 | |
| 8058 – 98 | |
| 8059 – 56 | |

1. Multiplication of two digit BCD numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,08 | Get multiplicand in C register |
|  | MVI | D,04 | Get multiplier in D register |
|  | XRA | A | Clear accumulator to save sum |
|  | MOV | B,A | Clear B register to save generated carry |
| LOOP | ADD | C | Add multiplicand to accumulator |
|  | DAA |  | Adjust the sum for decimal value |
|  | JNC | NEXT | If no carry go to NEXT |
|  | INR | B | If carry Increment B register by 1 |
| NEXT | DCR | D | Decrement the D register by 1, if the multiplicand is added to accumulator |
|  | JNZ | LOOP | If counter is not zero, go to LOOP to add multiplicand again |
|  | STA | 8600 | Stores the product in 8600 memory location |
|  | MOV | A,B | Moves the contents of B register (carry contents) to accumulator |
|  | STA | 8601 | Store the carry in 8601 memory location |
|  | HLT |  | Terminate program execution |

Multiply 08 and 04. The product is 2 digits BCD(32). Thus the content of 8600 is 32 and the content of 8601 memory location is 00

|  |  |
| --- | --- |
| Input | Output |
| 08D(multiplicand) | 8600 – 32D(product) |
| 04D(multiplier) | 8601 -00D(carry) |

1. Addition of two 16 bit numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,00 | Clear C register to save carry |
|  | LHLD | 8500 | Load the HL register pair with the contents of 8500 and 8501 locations |
|  | XCHG |  | Exchange the contents of HL and DE pair |
|  | LHLD | 8502 | Load the HL register pair with the contents of 8502 and 8503 |
|  | DAD | D | Add the contents of HL and DE register pair and result is in HL pair |
|  | JNC | LOOP | If no carry after the addition , then jump to LOOP |
|  | INR | C | If there is carry, after the addition increment C register by 1 |
| LOOP | SHLD | 8504 | Store the contents of L register in 8504 and H register contents in 8505 memory locations |
|  | MOV | A,C | Move the carry to accumulator |
|  | STA | 8506 | Store the carry in 8506 memory locations |

Add 3125H and 8064H

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 25H(least significant byte of 1st number) | 8504 – 89H(least significant byte of sum) |
| 8501 – 31H(most significant byte of 1st number) | 8505 – B1H(most significant byte of sum) |
| 8502 – 64H(least significant byte of 2nd number) | 8506 – 00H(carry) |
| 8503 – 80H(most significant byte of 2nd number) |  |

1. 4-digit BCD addition

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,00 | Clear C register to save carry |
|  | LHLD | 8500 | Load the HL register pair with the contents of 8500 and 8501 locations to get 1st 4-digit BCD number |
|  | XCHG |  | Exchange the contents of HL and DE pair to move the 1st 4-diit BCD number to DE pair |
|  | LHLD | 8502 | Load the HL register pair with the contents of 8502 and 8503 locations to get 2nd 4-digit BCD number |
|  | MOV | A,E | Move the lower byte of 1st 4-digit BCD number to accumulator |
|  | ADD | L | Add the lower byte of 1st number with the lower byte of 2nd number |
|  | DAA |  | Adjust the sum for decimal value |
|  | JNC | LOOP1 | If no carry jump to LOOP1 |
|  | INR | D | If carry increment D by 1 |
| LOOP1 | STA | 8600 | Store the lower byte BCD sum in8600 memory locations |
|  | MOV | A,D | Load the higher byte of 1st 4-digit BCD number to accumulator |
|  | ADD | H | Add the higher byte of 1st number with the higher byte of 2nd number |
|  | DAA |  | Adjust the sum for decimal value |
|  | JNC | LOOP2 | If no carry jump to LOOP2 |
|  | INR | C | If carry increment C by 1 |
| LOOP2 | STA | 8601 | Store the higher byte BCD sum in 8601 memory locations |
|  | MOV | A,C | Move the carry to A |
|  | STA | 8602 | Store the carry in 8602 memory locations |

The register pair DE contents is 3456 and HL contents is 7890.the result is 11346

|  |  |
| --- | --- |
| Input | output |
| 8500 – 56D(E register contents) | 8600 – 46D(lower byte sum) |
| 8501 – 34D(D register contents) | 8601 – 13D(higher byte sum) |
| 8502 – 90D(L register contents) | 8602 – 01D(carry or 3rd byte of BCD) |
| 8503 – 78D(H register contents) |  |

1. Subtraction of two 16 bit numbers

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,00 | Clear C register to save carry |
|  | LHLD | 8500 | Load the HL register pair direct with the contents of 8500 and 8501 locations |
|  | XCHG |  | Exchange the contents of HL and DE pair |
|  | LHLD | 8502 | Load the HL register pair direct with the contents of 8502 and 8503 locations |
|  | MOV | A,E | Move the lower byte of minuend to accumulator |
|  | SUB | L | Subtract the lower byte of data |
|  | JNC | LOOP1 | If [E] > [L] go to LOOP1 |
|  | DCR | D | If [E] < [L] decrement D by 1, since barrow is needed |
| LOOP1 | STA | 8600 | Store the lower byte difference in8600 memory locations |
|  | MOV | A,D | Load the higher byte minuend to accumulator |
|  | SUB | H | Subtract the higher byte of data |
|  | JNC | LOOP2 | If [D] >[H] go to LOOP2 |
|  | INR | C | If [D] < [H] increment C by 1 |
| LOOP2 | STA | 8601 | Store the higher byte difference in 8601 memory locations |
|  | MOV | A,C | Move the carry to A |
|  | STA | 8602 | Store the carry in 8602 memory locations |

Subtract 2894H from F436H

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 36H(least significant byte of 1st number) | 8600 – A2H(least significant byte difference) |
| 8501 – F4H(most significant byte of 1st number) | 8601 – CBH(most significant byte difference) |
| 8502 – 94H(least significant byte of 2nd number) | 8602 – 00H(borrow) |
| 8503 – 28H(most significant byte of 2nd number) |  |

1. Block transfer

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Setup HL as a memory pointer for the source memory |
|  | LXI | D,8600 | Setup DE as a memory pointer for the destina tion memory |
|  | MVI | B,0A | Setup B as a byte counter. To transfer bytes of data to new memory location B is loaded with 0A |
| NEXT | MOV | A,M | Move the data byte of source memory to accumulator |
|  | STAX | D | Transfer the data byte to the destination memory |
|  | INX | H | Increment the source memory pointer to get the next data byte |
|  | INX | D | Increment the destination memory pointer to get the next data byte |
|  | DCR | B | Decrement the counter by 1, if one data byte is transferred |
|  | JNZ | NEXT | If counter is not zero, go back to NEXT to transfer next data bytes |
|  | HLT |  | Termination |

|  |  |  |  |
| --- | --- | --- | --- |
| Input(Before execution) | | Output(after execution) | |
| Source memory | Destination memory | Source memory | Destination memory |
| 8500 – 35H | 8600 – 00H | 8500 – 35H | 8600 – 35H |
| 8501 – 45H | 8601 – 00H | 8501 – 45H | 8601 – 45H |
| 8502 – 55H | 8602 – 00H | 8502 – 55H | 8602 – 55H |
| 8503 – 65H | 8603 – 00H | 8503 – 65H | 8603 – 65H |
| 8504 – 75H | 8604 – 00H | 8504 – 75H | 8604 – 75H |
| 8505 – 85H | 8605 – 00H | 8505 – 85H | 8605 – 85H |
| 8506 – 95H | 8606 – 00H | 8506 – 95H | 8606 – 95H |
| 8507 – 25H | 8607 – 00H | 8507 – 25H | 8607 – 25H |

1. Block transfer of data bytes in reverse order (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Setup HL as a memory pointer for the source memory |
|  | LXI | D,8600 | Setup DE as a memory pointer for the destina tion memory |
|  | MVI | B,0A | Setup B as a byte counter. To transfer bytes of data to new memory location B is loaded with 0A |
| NEXT | MOV | A,M | Move the data byte of source memory to accumulator |
|  | STAX | D | Transfer the data byte to the destination memory |
|  | INX | H | Increment the source memory pointer to get the next data byte |
|  | DCR | D | Decrement the destination memory pointer to get the next data byte |
|  | DCR | B | Decrement the counter by 1, if one data byte is transferred |
|  | JNZ | NEXT | If counter is not zero, go back to NEXT to transfer next data bytes |
|  | HLT |  | Termination |

|  |  |  |  |
| --- | --- | --- | --- |
| Input(Before execution) | | Output(after execution) | |
| Source memory | Destination memory | Source memory | Destination memory |
| 8500 – 35H | 8600 – 00H | 8500 – 35H | 8600 – 25H |
| 8501 – 45H | 8601 – 00H | 8501 – 45H | 8601 – 95H |
| 8502 – 55H | 8602 – 00H | 8502 – 55H | 8602 – 85H |
| 8503 – 65H | 8603 – 00H | 8503 – 65H | 8603 – 75H |
| 8504 – 75H | 8604 – 00H | 8504 – 75H | 8604 – 65H |
| 8505 – 85H | 8605 – 00H | 8505 – 85H | 8605 – 55H |
| 8506 – 95H | 8606 – 00H | 8506 – 95H | 8606 – 45H |
| 8507 – 25H | 8607 – 00H | 8507 – 25H | 8607 – 35H |

1. 1’s complement of 8 bit number (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LDA | 8500 | Load the 8 bit data to accumulator |
|  | CMA |  | Complement the content of accumulator |
|  | STA | 8501 | Store the result in 8501 memory location |
|  | HLT |  | Termination |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 7C H | 8501 – 83 H |

1. 1’s complement of 16 bit number (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Setup HL register pair as memory pointer with memory address 8500 to get data bytes |
|  | MOV | A,M | Move the lower data byte to accumulator |
|  | CMA |  | Complement the lower data byte |
|  | STA | 8502 | Store the result in 8502 memory location |
|  | INX | H | Increment the pointer to get the higher data byte |
|  | MOV | A,M | Move the higher data byte to accumulator |
|  | CMA |  | Complement the higher data byte |
|  | STA | 8503 | Store the result in 8503 memory location |
|  | HLT |  | Termination |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 8B H | 8502 – 74 H |
| 8501 – 36 H | 8503 – C9 H |

1. 2’s complement of 8 bit number (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,00 | Clear C register to store carry |
|  | LDA | 8500 | Load the 8 bit data to accumulator |
|  | CMA |  | Complement he content of accumulator |
|  | ADI | 01 | Add 01 to the accumulator to get 2’s complement of a number |
|  | JNC | LOOP | If no carry, go to LOOP label |
|  | INR | C | If there is carry, increment C register by 1 |
| LOOP | STA | 8501 | Store the result in memory location 8501 |
|  | MOV | A,C | Move the carry to accumulator |
|  | STA | 8502 | Store the carry in memory location 8502 |
|  | HLT |  | Termination |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 9F H | 8501 – 61 H |
|  | 8502 – 00 H |

1. Fibonacci series (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,0A | Setup a counter by loading count 0A to C register |
|  | LXI | H,8500 | Setup HL as memory pointer with memory address 8500 |
|  | MOV | A,M | Get 00 too accumulator |
|  | INX | H | Increment the pointer to get next data |
|  | MOV | D,M | Get 01 to D register |
| LOOP | ADD | D | Add content of accumulator with the content of D register |
|  | DAA |  | Decimal adjust accumulator |
|  | INX | H | Increment pointer to store sum |
|  | MOV | M,A | Move accumulator comtent to next location |
|  | MOV | A,D | Move D register to accumulator |
|  | MOV | D,M | Move content of memory location to D register |
|  | DCR | C | Decrement C register by 1 |
|  | JNZ | LOOP | If counter value is not zero, go to LOOP |
|  | HLT |  | Termination |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 00 H | 8500 – 00H |
| 8501 – 01 H | 8501 – 01H |
|  | 8502 – 02H |
|  | 8503 – 03H |
|  | 8504 – 05H |
|  | 8505 – 08H |
|  | 8506 – 13H |
|  | 8507 – 21H |
|  | 8508 – 34H |
|  | 8509 – 55H |

1. Convert two digit packed BCD to two unpacked BCD (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LDA | 8500 | Get e packed BCD number stored in 8500 memory location to accumulator |
|  | MOV | B,A | Move the packed BCD to B register |
|  | ANI | 0F | Mask higher nibble to get the digit BCD1 |
|  | STA | 8600 | Store the unpacked least significant BCD digit (BCD1) in 8600 memory location |
|  | MOV | A,B | Move the original packed BCD number again to accumulator to get BCD2 |
|  | ANI | F0 | Mask lower nibble to convert packed BCD into unpacked BCD2 |
|  | RRC |  | Adjust higher BCD2 digit as a lower digit |
|  | RRC |  |
|  | RRC |  |
|  | RRC |  |
|  | STA | 8601 | Store the unpacked most significant BCD digit (BCD2) in 8601 memory location |
|  | HLT |  | Termination |

The packed BCD number is 58 stored in8500. The unpacked BCD1 is 08 andBCD2 is 05 stored in 8600 and 8601

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 58 D (BCD) | 8600 – 08 D(BCD1) |
| 8601 – 05 D(BCD2) |

1. Convert two digit unpacked BCD to two packed BCD (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LDA | 8500 | Get the least significant BD digit (04) is stored in 8500 memory location to accumulator |
|  | MOV | B,A | Move the least significant BCD digit to B register |
|  | LDA | 8501 | Get the most significant BD digit (09) is stored in 8501 memory location to accumulator |
|  | RLC |  | Adjust the position of the second digit (09 is changed to 90) |
|  | RLC |  |
|  | RLC |  |
|  | RLC |  |
|  | ADD | B | Add BCD2 and BCD1 to get BCD |
|  | STA | 8600 | Store the packed BCD 8601 memory location |
|  | HLT |  | Termination |

The packed BCD number is 58 stored in8500. The unpacked BCD1 is 08 andBCD2 is 05 stored in 8600 and 8601

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 04 D (BCD1)  8501 – 09 D (BCD2) | 8600 – 94 D(BCD) |
|  |

1. To find largest of two numbers (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LXI | H,8500 | Setup Hl as memory pointer with memory address 8500 to store 1st data byte |
|  | MOV | A,M | Moves the data byte stored in 8051memory location addressed by HL register pair to accumulator |
|  | INX | H | Increments the memory pointer by 1 (8501) to store the 2nd data byte |
|  | CMP | M | Compares the two numbers |
|  | JNC | NEXT | If no carry larger number is in accumulator then go to NEXT |
|  | MOV | A,M | If carry, the larger number in 8501 memory location the move the larger number to accumulator |
| NEXT | STA | 8600 | Stores the larger number in 8600 |
|  | HLT |  | Termination |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 35 H | 8600 – 45 H |
| 8501 – 45 H |

18 .to find smallest of ‘N’ byte numbers (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | C,0A | Setup C register with counter value 0A to find thesmallest of ten data bytes |
|  | LXI | H,8500 | Setup HL register pair as memory pointer with memory address 8500 |
|  | MOV | A,M | Move the data byte stored in 8500 memory location to accumulator |
|  | DCR | C | Decrement counter by1. If the 1sr byte is loaded to accumulator |
| LOOP | INX | H | Increments the memory pointer by 1 to store the next data byte |
|  | CMP | M | Compares the two numbers |
|  | JC | NEXT | If carry , smaller number is in accumulator then go to NEXT |
|  | MOV | A,M | If no carry, the smaller number is in memory pointer then move the smaller number to accumulator |
| NEXT | DCR | C | Decrement the counter by 1, if one data byte is checked |
|  | JNZ | LOOP | If counter is not zero, go to LOOP to find the smallest of the reamaining data bytes |
|  | STA | 8600 | Store the smallest in 8600 memory location |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 35H | 8600 – 25H |
| 8501 – 45H |
| 8502 – 55H |
| 8503 – 65H |
| 8504 – 75H |
| 8505 – 85H |
| 8506 – 95H |
| 8507 – 25H |
| 8508 – 45 H |
| 1. – 45 H |

19.Sorting of array in ascending order

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | MVI | D,05 | Setup D register as counter 1 with a counter value 05 to arrange ten bytes of numbers in ascending order |
| LOOP3 | LXI | H,8500 | Setup HL as memory pointer [M] with memory address 8500 |
|  | MVI | C,05 | Setup C register as counter 2 with a counter value 05 to arrange ten bytes of numbers in ascending order |
| LOOP2 | MOV | A,M | Move the data byte stored in memory pointer to accumulator |
|  | INX | H | Increment he memory pointer by 1 |
|  | CMP | M | Compare the contents of memory pointer with accumualator contents |
|  | JC | LOOP1 | If [A]<[M] jump to LOOP1 . otherwise exchange the contents of accumulator and memory pointer |
|  | MOV | B,M | If [A] >[M] Exchange the contents of memory pointer [M] and [M-1] |
|  | MOV | M,A |
|  | DCX | H |
|  | MOV | M,B |
|  | INX | H |
| LOOP1 | DCR | C | Decrement counter by one if the content of [M-1] <[M] |
|  | JNZ | LOOP2 | If counter is not zero go back to LOOP2 to keep smaller number in memory pointer |
|  | DCR | D | Decrement the D counter by one |
|  | JNZ | LOOP3 | If the counter is not zero go back to LOOP3 to repeat the same operation to arrange the numbers in ascending order |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 92 H | 8500 – 50H |
| 8501 – 50 H | 8501 – 55H |
| 8502 – 55 H | 8502 – 75H |
| 8503 – AA H | 8503 – 92H |
| 8504 – 75 H | 8504 – AAH |

20. Program to convert ASCII to HEX (PART-B)

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Mneumonics | Operand | Comment |
|  | LDA | 8500 | Load the input data in the memory location 8500 |
|  | SUI | 30 | Subtract 30 with the contents of accumulator |
|  | CPI | 0A | Compare the contents of accumulator with 0A |
|  | JC | LOOP | If [A] <0A then jump to LOOP |
|  | SUI | 07 | Subtract 07 with the contents of accumulator |
| LOOP | STA | 8501 | Store the result in 8501 memory location |

|  |  |
| --- | --- |
| Input | Output |
| 8500 – 31 H | 8501 – 0B D |