**ASSIGNMENT 2**

**EXP 1:-**

***Write an assembly program to take two numbers as input and output their sum using registers. // Write an 8085-assembly program that’s take two 8- bit numbers as input, each stored in a separate register and output their sum.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A, 25H | 3E 25 |
| 0802 | MVI B, 1AH | 06 1A |
| 0804 | ADD B | 80 |
| 0805 | STA 3000H | 32 00 30 |
| 0808 | HLT | 76 |

**OUTPUT:-** THE FINAL VALUE OF THE ACCUMLATOR AFTER EXECUTION WILL BE 3FH I.E (63)10 WILL BE STORED IN MEMORY LOCATION 3000H

**EXP 2:-**

***Write an assembly program to take two numbers as input and output their sum using memory.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H, 2000H | 21 00 20 |
| 0803 | MOV B,M | 46 |
| 0804 | LXI H,2001H | 21 01 20 |
| 0807 | MOV C,M | 4E |
| 0808 | MOV A,B | 78 |
| 0809 | ADD C | 91 |
| 080A | STA 3000H | 32 00 30 |
| 080D | HLT | 76 |

**INPUT:-** AT 2000H => 02

AT 2001H => 03

**OUTPUT:-** AT 3000H => 05

**EXP 3:-**

***Write an assembly program to take two numbers as input and output their difference using registers.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A,35H | 3E 35 |
| 0802 | MVI B,15H | 06 15 |
| 0804 | SUB B | 90 |
| 0805 | STA 3000H | 32 00 30 |
| 0808 | HLT | 76 |

**OUTPUT:-** AT 3000H =>20H

**EXP 4:-**

***Write an assembly program to take two numbers as input and output their difference using memory.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H, 2000H | 21 00 20 |
| 0803 | MOV B,M | 46 |
| 0804 | LXI H,2001H | 21 01 20 |
| 0807 | MOV C,M | 4E |
| 0808 | MOV A,B | 78 |
| 0809 | SUB C | 91 |
| 080A | STA 3000H | 32 00 30 |
| 080D | HLT | 76 |

**INPUT:-** AT 2000H => 05

AT 2001H => 03

**OUTPUT:-** AT 3000H => 02

**EXP 5:-**

***Write an assembly program to take two numbers as input and output their product using registers.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A,00H | 3E 00 |
| 0802 | MVI B,04H | 06 04 |
| 0804 | MVI C,03H | 0E 03 |
| 0806 | MVI D,00H | 16 00 |
| 0808 | LOOP2: ADD B | 80 |
| 0809 | JNC LOOP | D2 0D 08 |
| 080C | INR D | 14 |
| 080D | LOOP: DCR C | 0D |
| 080E | JNZ LOOP2 | C2 08 08 |
| 0811 | STA 3000H | 32 00 30 |
| 0814 | HLT | 76 |

**OUTPUT:-** AT 3000H =>0C H

**EXP 6:-**

***Write an assembly program to take two numbers as input and output their product using memory.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H, 2000H | 21 00 20 |
| 0803 | MOV B,M | 46 |
| 0804 | MVI A,00H | 3E 00 |
| 0806 | MVI D,00H | 16 00 |
| 0808 | INX H | 23 |
| 0809 | MOV C,M | 4E |
| 080A | LOOP2: ADD B | 80 |
| 080B | JNC LOOP | D2 01 08 |
| 080E | INR D | 14 |
| 080F | LOOP: DCR C | 0D |
| 0810 | JNZ LOOP2 | C2 0A 08 |
| 0813 | STA 2005H | 32 05 20 |
| 0816 | MOV A,D | 7A |
| 0817 | STA 2006H | 32 06 20 |
| 081A | HLT | 76 |

**INPUT:-** AT 2000H => 04

AT 2001H => 03

**OUTPUT:-** AT 2005H => 0C H

**EXP 7:-**

***Write an assembly program to take two numbers as input and output their quotient using registers.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A,00H | 3E 00 |
| 0802 | MVI B,06H | 06 06 |
| 0804 | MVI C,03H | 0E 03 |
| 0806 | MVI D,00H | 16 00 |
| 0808 | MOV A,B | 78 |
| 0809 | LOOP: CMP C | B9 |
| 080A | JC LOOP2 | DA 12 08 |
| 080D | INR D | 14 |
| 080E | SUB C | 91 |
| 080F | JMP LOOP | C3 09 08 |
| 0812 | LOOP2:MOV A,D | 7A |
| 0813 | STA 3000H | 32 00 30 |
| 0816 | HLT | 76 |

**OUTPUT:-** AT 3000H =>02 H

**EXP 8:-**

***Write an assembly program to take two numbers as input and output their quotient using memory.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H, 2000H | 21 00 20 |
| 0803 | MOV B,M | 46 |
| 0804 | INX H | 23 |
| 0805 | MOV C,M | 4E |
| 0806 | MVI A,00H | 3E 00 |
| 0808 | MVI D,00H | 16 00 |
| 080A | MOV A,B | 78 |
| 080B | LOOP: SUB C | 91 |
| 080C | INR D | 14 |
| 080D | CMP C | B9 |
| 080E | JNC LOOP | D2 0B 08 |
| 0811 | STA 2001H | 32 02 20 |
| 0814 | MOV A,D | 7A |
| 0815 | STA 2003H | 32 03 20 |
| 0818 | HLT | 76 |

**INPUT:-** AT 2000H => 06

AT 2001H => 03

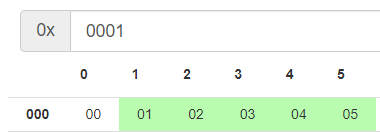
**OUTPUT:-** AT 2003H => 02H

**ASSIGNMENT 3**

**EXP 1:-**

***Write an assembly program to display numbers from 1 to 5 using a loop.***

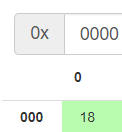
|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A, 01H | 3E 01 |
| 0802 | MVI B, 01H | 06 01 |
| 0804 | MVI C, 05H | 0E 05 |
| 0806 | LXI D,0001H | 11 01 00 |
| 0809 | LOOP: STAX D | 12 |
| 080A | INX D | 13 |
| 080B | ADD B | 80 |
| 080C | DCR C | 0D |
| 080D | JNZ LOOP | C2 09 08 |
| 0810 | HLT | 76 |

**OUTPUT:-**

**EXP 2:-**

***Write an assembly program to calculate the factorial of a given number.***

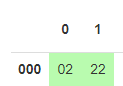
|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI B, 04H | 06 04 |
| 0802 | MVI D, 01H | 16 01 |
| 0804 | FACT: CALL MULT | CD 10 08 |
| 0807 | DCR B | 05 |
| 0808 | JNZ FACT | C2 04 08 |
| 080B | LXI H,000H | 21 00 00 |
| 080C | MOV M,D | 72 |
| 080F | HLT | 76 |
| 0810 | MULT: MOV E,B | 58 |
| 0811 | MVI A,00H | 3E 00 |
| 0813 | LOOP: ADD D | 82 |
| 0814 | DCR E | 10 |
| 0815 | JNZ LOOP | C2 13 08 |
| 0818 | MOV D,A | 57 |
| 0819 | RET | C9 |

**OUTPUT:-**

**EXP 3:-**

***Develop an assembly program to check if a given number is odd or even using branching statements***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LDA 0000H | 06 04 |
| 0802 | RAR | 16 01 |
| 0804 | JC LOOP | CD 10 08 |
| 0807 | MVI A,22H | 05 |
| 0808 | STA 0001H | C2 04 08 |
| 080B | HLT | 21 00 00 |
| 080C | LOOP: MVI A,11H | 72 |
| 080F | STA 0001H | 76 |
| 0810 | HLT | 58 |

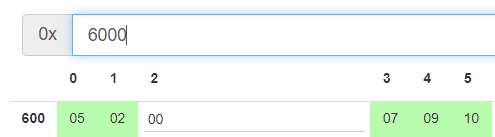
**INPUT:-** AT 0000H => 02

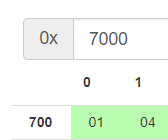
**OUTPUT:-** AT 0001H=> 22 H

**EXP 4:-**

***Develop an assembly program to determine if a given number is positive or negative using branching.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H,6000H | 21 00 60 |
| 0803 | MOV C,M | 4E |
| 0804 | MVI B,00H | 06 00 |
| 0806 | MVI D,00H | 16 00 |
| 0808 | MVI E,00H | 1E 00 |
| 080A | LOOP3: INX H | 23 |
| 080B | MOV A,M | 7E |
| 080C | ADI 00H | C6 00 |
| 080E | JZ LOOP1 | CA 15 08 |
| 0811 | JP LOOP2 | F2 19 08 |
| 0814 | INR E | 1C |
| 0815 | LOOP1: INR D | 14 |
| 0818 | JMP LOOP4 | C3 1A 08 |
| 0819 | LOOP2: INR B | 04 |
| 081C | LOOP4: DCR C | 0D |
| 081D | JNZ LOOP3 | C2 0A 08 |
| 081E | MOV A,D | 7A |
| 0821 | STA 7000H | 32 00 70 |
| 0824 | MOV A,B | 78 |
| 0825 | STA 7001H | 32 01 70 |
| 0828 | STA 2002H | 32 02 20 |
| 082B | HLT | 76 |

**INPUT:-**



**OUTPUT:-**

**EXP 5:-**

***Write an assembly language program to find the smaller of two numbers stored in memory locations 2501 H and 2502 H. Store the result in 2503H memory location.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LXI H,2501H | 21 01 25 |
| 0803 | MOV A,H | 7E |
| 0804 | LXI H,2502H | 21 02 05 |
| 0807 | MOV B,M | 46 |
| 0808 | CMP B | B8 |
| 0809 | JC LOOP | DA 0D 08 |
| 080C | MOV A,B | 78 |
| 080D | LOOP:STA 2503H | 32 03 25 |
| 0810 | HLT | 76 |

**INPUT:-** AT 2501=>13

AT 2502=>12

**OUTPUT:-** AT 2503=>12

**EXP 6:-**

***Write an assembly program to determine whether a given year is a leap year or not using conditional branches.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | LDA 2500H | 3A 00 25 |
| 0803 | RAR | 1F |
| 0804 | JC LOOP | DA 11 08 |
| 0807 | RAR | 1F |
| 0808 | JC LOOP | DA 11 08 |
| 080B | MVI A,02H | 3E 02 |
| 080D | STA 2501H | 32 01 25 |
| 0810 | HLT | 76 |
| 0811 | LOOP: MVI A,01H | 3E 01 |
| 0813 | STA 2501H | 32 01 25 |
| 0816 | HLT | 76 |

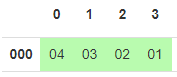
**INPUT:-** AT 2500=>2050

**OUTPUT:-** AT 2501=>02

**EXP 8:-**

***WRITE AN ASSEMBLY PROGRAM TO USE CONDITIONAL JUMP INSTRUCTION FOR PROGRAM FLOW.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | MVI A, 04H | 3E 04 |
| 0802 | LXI D, 0000H | 11 00 00 |
| 0805 | LOOP: STAX D | 12 |
| 0806 | INX D | 13 |
| 0807 | DCR A | 3D |
| 0808 | JNZ LOOP | C2 05 08 |
| 080B | HLT | 76 |



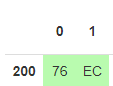
**ASSIGNMENT 4**

**ASSIGNMENT 4**

**EXP 1:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO SHIFT LEFT AN 8 BIT NUMBER BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RAL | 17 |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

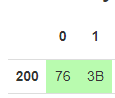
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => EC

**EXP 2:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO SHIFT RIGHT AN 8 BIT NUMBER BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RAR | 1F |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

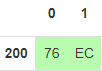
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => 3B

**EXP 3:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO ROTATE BITS TO THE LEFT BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RLC | 07 |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

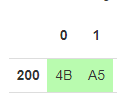
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => EC

**EXP 4:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO ROTATE BITS TO THE right BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RRC | 0F |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

**INPUT:-** AT 2000H => 4B

**OUTPUT:-** AT 2001H => A5

**EXP 5:-**

***Write an assembly program to generate a series of odd numbers up to a specified limit.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LXI H,2000H | 21 00 20 |
| 0803 | MOV C,M | 4E |
| 0804 | MVI A,01H | 3E 01 |
| 0806 | MVI B,02H | 06 02 |
| 0808 | LXI D,2001H | 11 10 20 |
| 080B | LOOP: STAX D | 12 |
| 080C | ADD B | 80 |
| 080D | INX D | 13 |
| 080E | CMP C | B9 |
| 080F | JC LOOP | DA 0B 08 |
| 0812 | HLT | 76 |

**INPUT:-** AT 2000H => 09

**OUTPUT:-** AT 2001H => 1

AT 2001H => 3

AT 2001H => 5

AT 2001H => 7

**EXP 6:-**

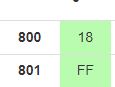
***Write an assembly program to generate a series of prime numbers within a specified range.***

|  |  |  |
| --- | --- | --- |
|  |  |  |

**EXP 7:-**

***Write an assembly program to implement a function to check if a given number is a palindrome.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 8000H | 3A 00 80 |
| 080A | MOV M,A | 77 |
| 080B | MVI C,08H | 0E 08 |
| 080C | LOOP: MOV A,H | 7C |
| 080D | RLC | 07 |
| 0806 | MOV M,A | 77 |
| 0807 | MOV A,D | 7A |
| 0808 | RAR | 1F |
| 0809 | MOV D,A | 57 |
| 080C | DCR C | 0D |
| 081D | JNZ LOOP | C2 06 08 |
| 0810 | MOV A,H | 7C |
| 0811 | CMP D | BA |
| 0812 | JZ TRUE | CA 1a 08 |
| 0815 | MVI A,00H | 3E 00 |
| 0817 | JMP EXIT | C3 1c 08 |
| 081A | TRUE:MVI A,FFH | 3E ff |
| 081C | EXIT:STA 8010H | 32 10 80 |
| 081F | HLT | 76 |

**INPUT:-** AT 8000H => 18

**OUTPUT:-** AT 8010H => FF

**EXP 8:-**

***Write an assembly program to implement a function to find the greatest common divisor (GCD) of two numbers.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | MVI A,09H | 3E 09 |
| 0802 | MVI B,03H | 06 07 |
| 0804 | CMP B | B8 |
| 0805 | JZ DOWN | CA 10 08 |
| 0808 | JNZ SHIFT | D2 0E 08 |
| 080B | MOV C,A | 4F |
| 080C | MOV A,B | 78 |
| 080D | MOV B,C | 41 |
| 080E | SHIFT: SUB B | 90 |
| 080F | CMP B | B8 |
| 0810 | JZ MOVE | CA 1C 08 |
| 0813 | JNZ SHIFT | D2 0E 08 |
| 0816 | MOV C,A | 4F |
| 0817 | MOV A,B | 78 |
| 0818 | MOV B,C | 41 |
| 0819 | JMP SHIFT | C3 0E 08 |
| 081C | MOVE: MOV A,B | 78 |
| 081D | DOWN: STA 0000H | 32 00 0 |
| 081E | HLT | 76 |

**OUTPUT:-** AT 0000H => 03

**EXP 9:-**

***Write an assembly program to find the least common multiple (LCM) of two numbers using a function.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LXI H,8000H | 21 00 80 |
| 0803 | MOV C,M | 4E |
| 0804 | MVI B,00H | 06 00 |
| 0806 | INX H | 23 |
| 0807 | MOV A,M | 7E |
| 0808 | CMA | 2F |
| 0809 | MOV E,A | 5F |
| 080A | MVI D,FFH | 16 FF |
| 080C | INX D | 13 |
| 080D | LXI H,0000H | 21 00 00 |
| 0810 | NEXT: DAD B | 09 |
| 0811 | SHLD 8050H | 22 50 80 |
| 0814 | LOOP: DAD D | 19 |
| 0815 | JNC SKIP | D2 20 08 |
| 0820 | MOV A,H | 7C |
| 0823 | ORA L | BF |
| 081A | JZ EXIT | CA 26 08 |
| 081D | JMP LOOP | C3 14 08 |
| 0820 | SKIP: LHLD 8050H | 2A 50 80 |
| 0823 | JMP NEXT | C3 10 08 |
| 0826 | EXIT: LHLD 8050H | 2A 50 80 |
| 0829 | HLT | 76 |

**INPUT**:- AT 8000H => 03 AT 8001H=> 06

**OUTPUT:-** AT 8050H => 06

**EXP 10:-**

***Write an assembly program to develop a function to perform the*** ***bubble sort algorithm on an integer array.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | START: LXI H,0040H | 21 40 00 |
| 0803 | MVI D,00H | 16 00 |
| 0805 | MOV C,M | 4E |
| 0806 | DCR C | 0D |
| 0807 | INX H | 23 |
| 0808 | FLAG: MOV A,M | 7E |
| 0809 | INX H | 23 |
| 080A | CMP M | BE |
| 080B | JC NEXT | DA 18 00 |
| 080E | JZ NEXT | CA 18 00 |
| 0811 | MOV B,M | 46 |
| 0812 | MOV M,A | 77 |
| 0813 | DCX H | 2B |
| 0814 | MOV M,B | 70 |
| 0815 | INX H | 23 |
| 0816 | MVI D,01H | 16 01 |
| 0818 | NEXT: DCR C | 0D |
| 0819 | JNZ FLAG | C2 08 00 |
| 081C | MOV A,D | 7A |
| 081D | CPI 01H | FE 01 |
| 081F | JZ START | CA 00 00 |
| 0822 | HLT | 76 |

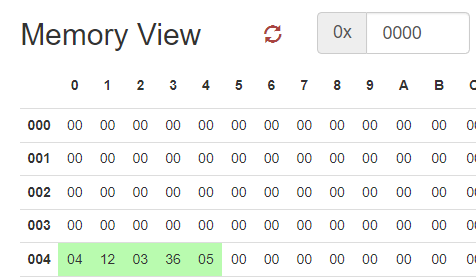
**INPUT**:- AT 0040H => 04

AT 0041H=> 12

AT 0042H=> 03

AT 0043H=> 36

AT 0044H=> 05

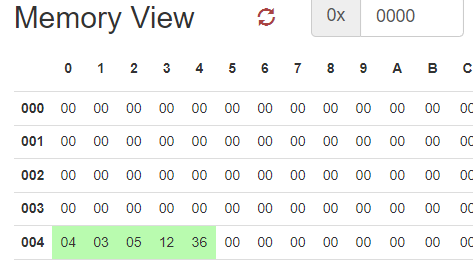
****

**OUTPUT:-** AT 0041H=> 03

AT 0042H=> 05

AT 0043H=> 12

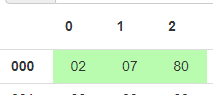
AT 0044H=> 36



**EXP 11:-**

***Write an assembly program to develop a function to compute the power of a number (x^n).***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LXI H,0000H | 21 00 00 |
| 0803 | MOV B,M | 46 |
| 0804 | INX H | 23 |
| 0805 | MOV C,M | 4E |
| 0806 | MVI D,01H | 16 01 |
| 0808 | POWERLOOP: CALL MULTIPLY | CD 12 08 |
| 080B | DCR C | 0D |
| 080C | JNZ POWERLOOP | C2 08 08 |
| 080F | INX H | 23 |
| 0810 | MOV M,D | 72 |
| 0811 | HLT | 76 |
| 0812 | MULTIPLY:MOV E,B | 58 |
| 0813 | MVI A,00H | 3E 00 |
| 0815 | MULTIPLYLOOP:ADD D | 82 |
| 0816 | DCR E | 1D |
| 0817 | JNZ MULTIPLYLOOP | C2 15 08 |
| 081A | MOV D,A | 57 |
| 081B | RET | C9 |

**INPUT:-** AT 0000H => 02

AT 0001H => 07

**OUTPUT:-** AT 0002H => 80

**EXP 12:-**

***Write an assembly program to calculate the sum of digits in a number using a function.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 0000H | 3A 00 00 |
| 0803 | MOV C,A | 4F |
| 0804 | ANI 0FH | E6 0f |
| 0806 | MOV B,A | 47 |
| 0807 | MOV A,C | 79 |
| 0808 | RRC | E6 0f |
| 080A | RRC | 0F |
| 080B | RRC | 0F |
| 080C | RRC | 0F |
| 080D | ANI 0FH | 0F |
| 080E | ADD B | 80 |
| 080F | STA 0001H | 32 01 00 |
| 0812 | HLT | 76 |

**INPUT:-** AT 0000H => 34

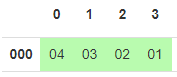
**OUTPUT:-** AT 0001H =>07

**ASSIGNMENT 3**

**EXP 8:-**

***WRITE AN ASSEMBLY PROGRAM TO USE CONDITIONAL JUMP INSTRUCTION FOR PROGRAM FLOW.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMONIES** | **OP CODE** |
| 0800 | MVI A, 04H | 3E 04 |
| 0802 | LXI D, 0000H | 11 00 00 |
| 0805 | LOOP: STAX D | 12 |
| 0806 | INX D | 13 |
| 0807 | DCR A | 3D |
| 0808 | JNZ LOOP | C2 05 08 |
| 080B | HLT | 76 |

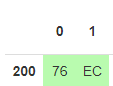


**ASSIGNMENT 4**

**EXP 1:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO SHIFT LEFT AN 8 BIT NUMBER BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RAL | 17 |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

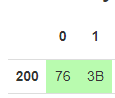
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => EC

**EXP 2:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO SHIFT RIGHT AN 8 BIT NUMBER BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RAR | 1F |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

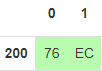
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => 3B

**EXP 3:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO ROTATE BITS TO THE LEFT BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RLC | 07 |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

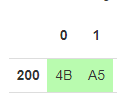
**INPUT:-** AT 2000H => 76

**OUTPUT:-** AT 2001H => EC

**EXP 4:-**

***WRITE AN 8085 ASSEMBLY PROGRAM TO ROTATE BITS TO THE right BY 1 BIT***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RRC | 0F |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

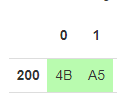
**INPUT:-** AT 2000H => 4B

**OUTPUT:-** AT 2001H => A5

**EXP 5:-**

***WRITE AN 8085 ASSEMBLY PROGRAM THAT GENERATES A SERIES OF ODD NUMBER STARTING FROM 1 AND CONTINUING UP TO A USER SPECUIFIC LIMIT.***

|  |  |  |
| --- | --- | --- |
| **MEMORY LOCATION** | **MEMORIES** | **OP CODE** |
| 0800 | LDA 2000H | 3A 00 20 |
| 0803 | RRC | 0F |
| 0804 | STA 2001H | 32 01 20 |
| 0807 | HLT | 76 |

**INPUT:-** AT 2000H => 4B

**OUTPUT:-** AT 2001H => A5