NAME : **Sujan Biswas**

BATCH: BSCE 2ND YEAR , (LATERAL)

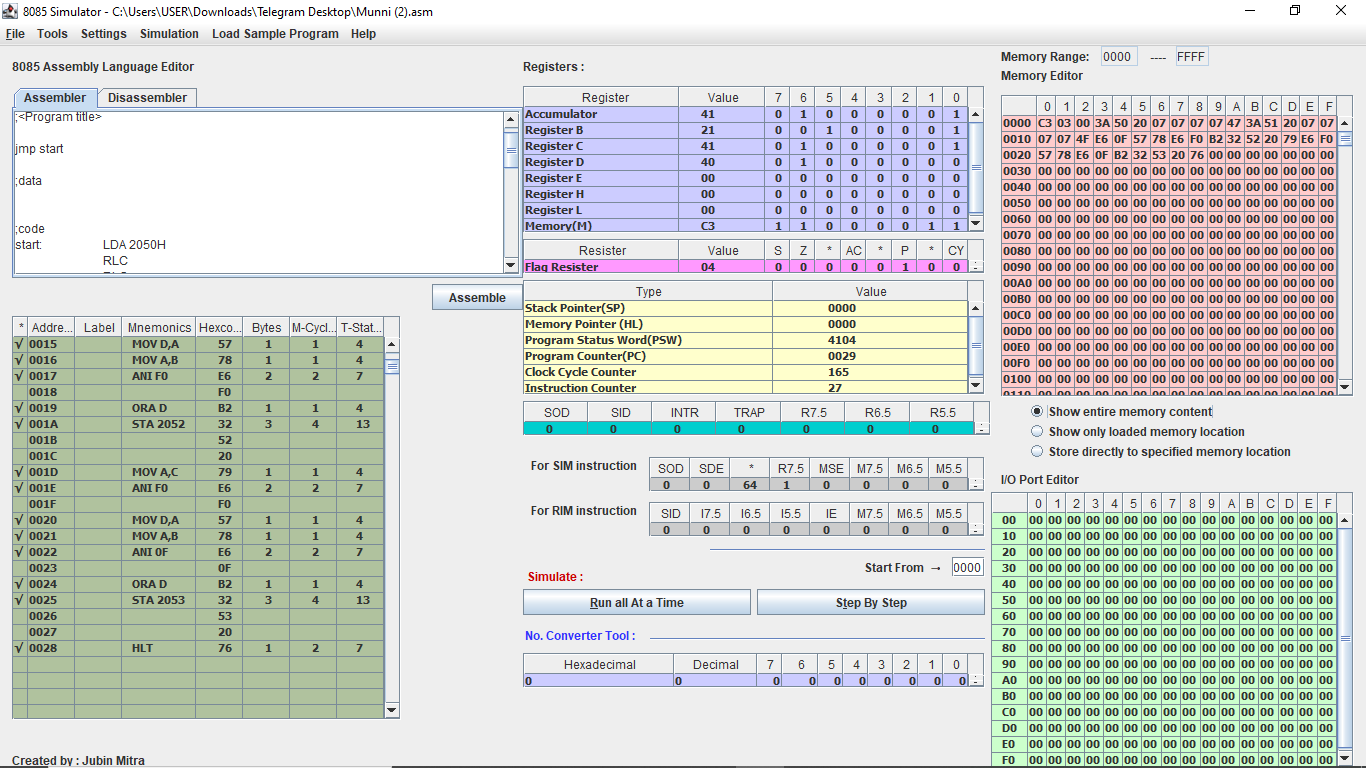
ROLL NO : 302010501003

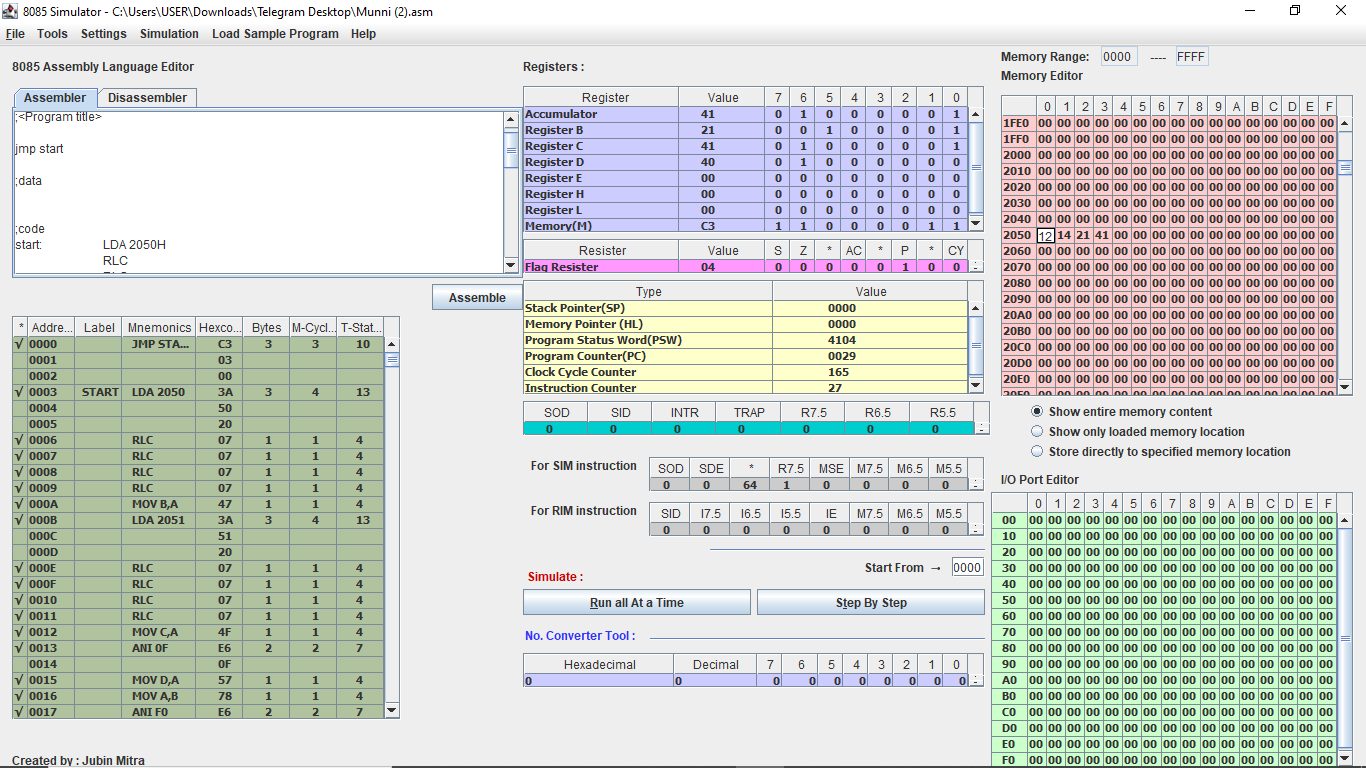
Microprocessor lab Assignment problem sheet #2

1. Two numbers MNH and KLH are stored in 2050 H and 2051 H , respectively . Write a program to assemble them a NKH and LMH store them in 2052 H and 2053 H.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 0000 | C3  03  00 |  | JMP START | Start the program |
| 0001 |
| 0002 |
| 2 | 0003 | 3A | START | LDA2050 H | Load accumulator direct memory location 2050 H [A=MN H] |
| 0004 | 50 |
| 0005 | 20 |
| 3 | 0006 | 07 |  | RLC | Rotate accumulator left without carry |
| 4 | 0007 | 07 |  | RLC | Rotate accumulator left without carry |
| 5 | 0008 | 07 |  | RLC | Rotate accumulator left without carry |
| 6 | 0009 | 07 |  | RLC | Rotate accumulator left without carry [After 4 rotation ,A=NM H] |
| 7 | 000A | 47 |  | MOV B,A | Move accumulator to B register [B=A=NM H] |
| 8 | 000B | 3A |  | LDA 2051 H | Load accumulator direct memory location 2051 H [A=KL H] |
| 000C | 51 |
| 000D | 20 |
| 9 | 000E | 07 |  | RLC | Rotate accumulator left without carry |
| 10 | 000F | 07 |  | RLC | Rotate accumulator left without carry |
| 11 | 0010 | 07 |  | RLC | Rotate accumulator left without carry |
| 12 | 0011 | 07 |  | RLC | Rotate accumulator left without carry [After 4 rotation, A= LK H] |
| 13 | 0012 | 4F |  | MOV C,A | Move accumulator to C register [C=A= LK H] |
| 14 | 0013 | E6 |  | ANI 0F | A=0K H |
| 0014 | 0F |
| 15 | 0015 | 57 |  | MOV D,A | Move accumulator to D register [D=A=0K H] |
| 16 | 0016 | 78 |  | MOV A,B | Move B register to accumulator [A=B=NM H] |
| 17 | 0017 | E6 |  | ANI F0 | A=N0 H |
| 0018 | F0 |
| 18 | 0019 | B2 |  | ORA D | A=N0 H|0K H=NK H |
| 19 | 001A | 32 |  | STA 2052 H | Load the contents of the accumulator in the address location 2052 H, M[2052= NK H] |
| 001B | 52 |
| 001C | 20 |
| 20 | 001D | 79 |  | MOV A,C | Move C register to accumulator [A=C=LK H] |
| 21 | 001E | E6 |  | ANI F0 | A=L0 H |
| 001F | F0 |
| 22 | 0020 | 57 |  | MOV D,A | Move accumulator to D register[A=D=L0 H] |
| 23 | 0021 | 78 |  | MOV A,B | Move B register  accumulator to[A=B=NM H] |
| 24 | 0022 | E6 |  | ANI 0F | A=0M H |
| 0023 | 0F |
| 25 | 0024 | B2 |  | ORA D | A=0M H|L0 H=LM H |
| 26 | 0025 | 32 |  | STA 2053 H | Load the contents of the accumulator in the address location 2053 H, M[2053=LM H] |
|  | 0026 | 53 |  |  |  |
| 0027 | 20 |  |
| 27 | 0028 | 76 |  | HLT | Terminate the program. |

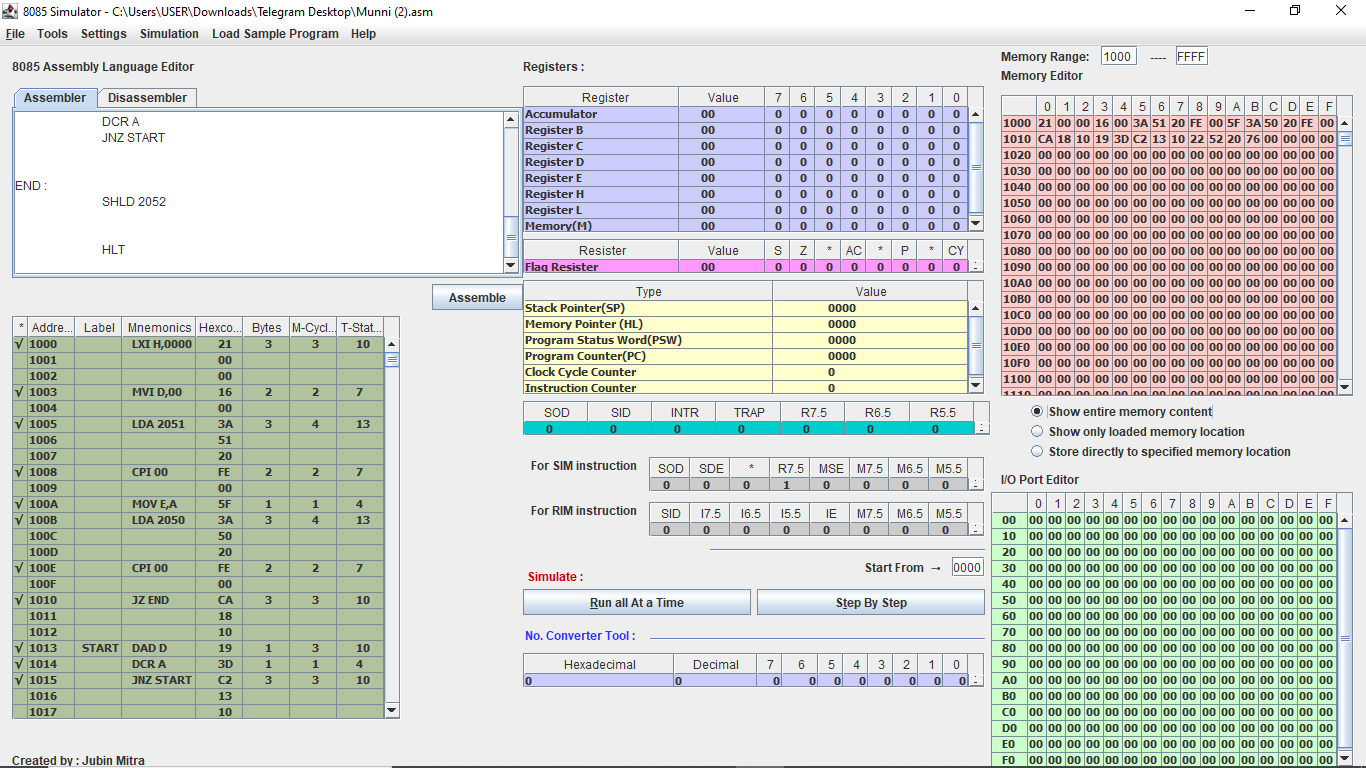
Simulator with loaded machine code:



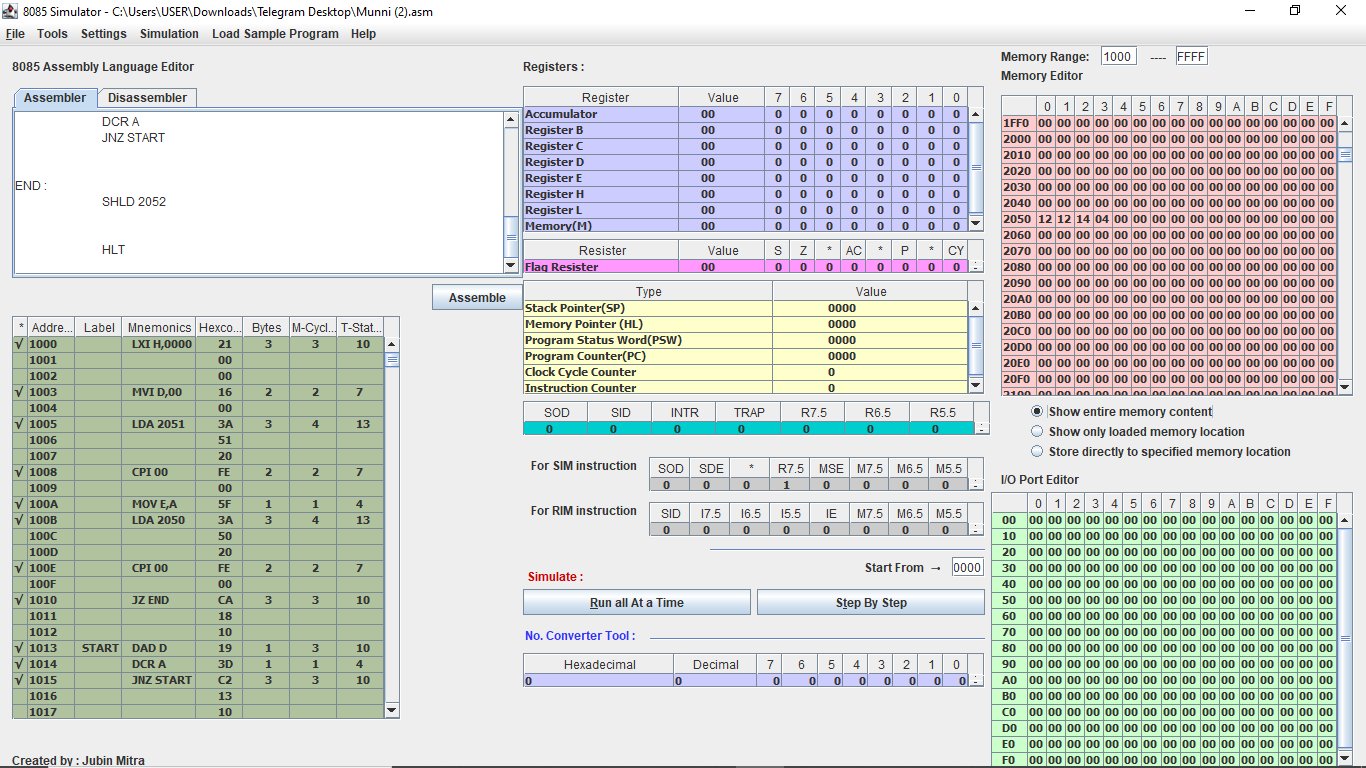
Sample input and output (M[2050 H]=12 H, M[2051 H]=14 H)

2. Two numbers A & B are stored in 2050 H and 2051 H , respectively . Write a program to perform A\*B and store the results in 2052 H and 2053 H.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 21 |  | LXI H,0000 | Contents of memory location 0000 H into HL register pair |
| 1001 | 00 |
| 1002 | 00 |
| 2 | 1003 | 16 |  | MVI D,00 | Move immediate to D register |
| 1004 | 00 |
| 3 | 1005 | 3A |  | LDA 2051 H | Load accumulator direct memory location 2051 H  Get B |
| 1006 | 51 |
| 1007 | 20 |
| 4 | 1008 | FE |  | CPI 00 H | If B=0? |
| 1009 | 00 |
| 5 | 100A | 5F |  | MOV E,A | Move accumulator to E register  E=A |
| 6 | 100B | 3A |  | LDA 2050 H | Load accumulator direct memory location 2050H  Get A |
| 100C | 50 |
| 100D | 20 |
| 7 | 100E | FE |  | CPI 00 H | Is A=0? |
| 100F | 00 |
| 8 | 1010 | CA |  | JZ END | If A=0,nothing to do |
| 1011 | 18 |
| 1012 | 10 |
| 9 | 1013 | 19 | START | DAD D | HL + DE |
| 10 | 1014 | 3D |  | DCR A | Decrement the A |
| 11 | 1015 | C2 |  | JNZ START | If DE has not been added A times ,add again |
| 1016 | 13 |
| 1017 | 10 |
| 12 | 1018 | 22 | END | SHLD 2052 H | Store results as specified |
| 1019 | 52 |
| 101A | 20 |
| 13 | 101B | 76 |  | HLT | Stop the program |

Simulator with loaded machine code:

Sample input and output (A= M[2050 H]=12 H,B=M[2051 H]=12 H, A\*B=144 H, M[2052 H]= 12 H, M[2053 H]=04 H).

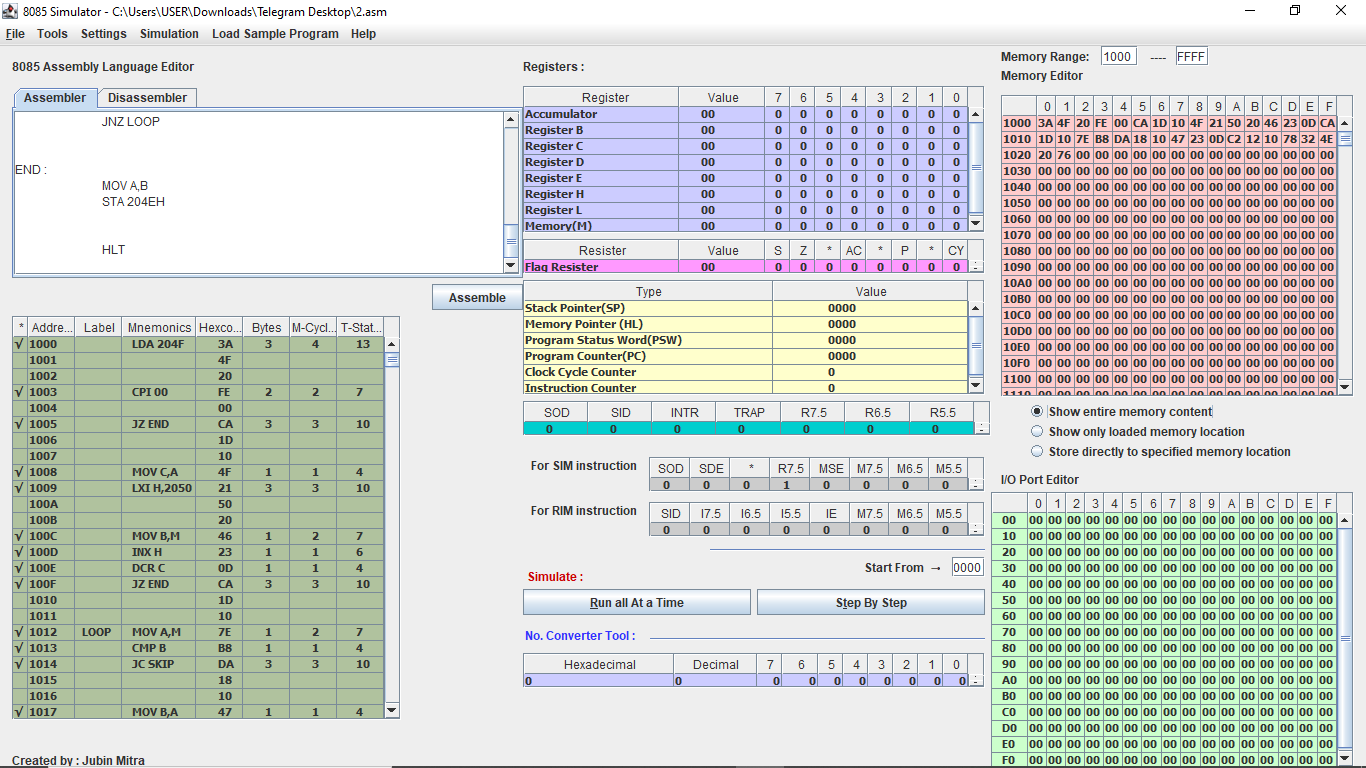


3.N numbers are stored in consecutive m/m location starting the from 2050 H. The value N is stored in 204F H.

I)Find maximum among the N numbers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 3A |  | LDA 204F H | Load accumulator direct memory location 204F H  [A=N H] |
| 1001 | 4F |
| 1002 | 20 |
| 2 | 1003 | FE |  | CPI 00 H | Is N =0? |
| 1004 | 00 |
| 3 | 1005 | CA |  | JZ END | If N =0,nothing to do |
| 1006 | 1D |
| 1007 | 10 |
| 4 | 1008 | 4F |  | MOV C,A | Move accumulator to C register |
| 5 | 1009 | 21 |  | LXI 2050 H | Contents of memory location 2050 H into HL register pair |
| 100A | 50 |
| 100B | 20 |
| 6 | 100C | 46 |  | MOV B,M | Move memory address to B register B is current maximum number |
| 7 | 100D | 23 |  | INX H | Increase the HL register ,then go the next number |
| 8 | 100E | 0D |  | DCR C | Decrement the C register , then check the number |
| 9 | 100F | CA |  | JZ END | If end , nothing to do ,store this |
| 1010 | 1D |
| 1011 | 7E |
| 10 | 1012 | 7E | LOOP | MOV A,M | Move the memory address to accumulator |
| 11 | 1013 | B8 |  | CMP B | Compare against current maximum |
| 12 | 1014 | DA |  | JC SKIP | If B>A, do nothing |
| 1015 | 18 |
| 1016 | 10 |
| 13 | 1017 | 47 |  | MOV B,A | Move accumulator to B register load a new maximum |
| 14 | 1018 | 23 | SKIP | INX H | Increase the HL ,then go the next number |
| 15 | 1019 | 0D |  | DCR C | Decrement the C then check the number |
| 16 | 101A | C2 |  | JNZ LOOP | If numbers left, continue checking |
| 101B | 12 |
| 101C | 10 |
| 17 | 101D | 78 | END | MOV A,B | Move B register to accumulator [A=Maximum] |
| 18 | 101E | 32 |  | STA 204E H | Store the maximum |
| 101F | 4E |  |
| 1020 | 20 |  |
| 19 | 1021 | 76 |  | HLT | Stop the program |

Simulator with loaded machine code:

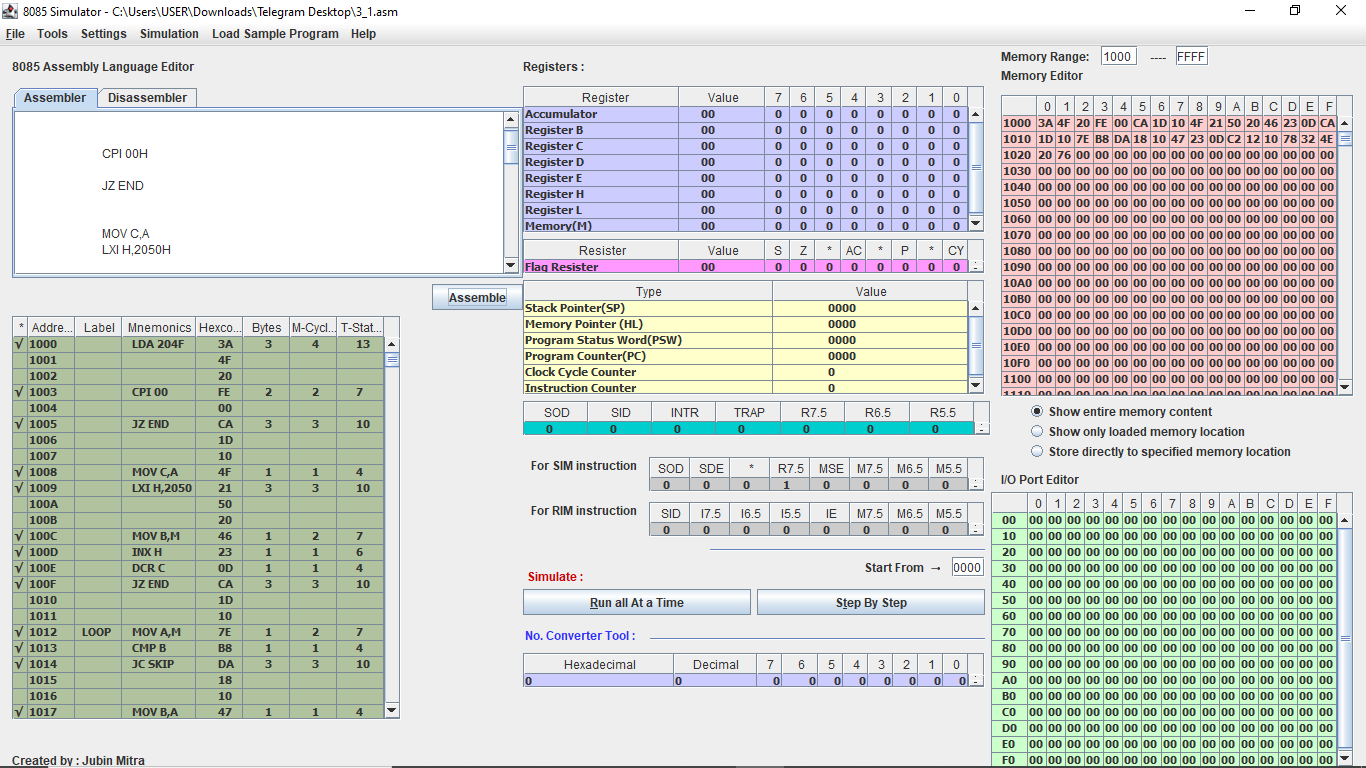


ii) Find the minimum among the N numbers.

We store the result (Minimum) in 204E H memory location.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 3A |  | LDA 204F H | Load accumulator direct memory location 204F H  [A=N H] |
| 1001 | 4F |
| 1002 | 20 |
| 2 | 1003 | FE |  | CPI 00 H | Is N=0? |
| 1004 | 00 |
| 3 | 1005 | CA |  | JZ END | If N =0,nothing to do |
| 1006 | 1D |
| 1007 | 10 |
| 4 | 1008 | 4F |  | MOV C,A | Move accumulator to C register |
| 5 | 1009 | 21 |  | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| 100A | 50 |
| 100B | 20 |
| 6 | 100C | 46 |  | MOV B,M | Move memory address to B register , B is current minimum number |
| 7 | 100D | 23 |  | INX H | Increase the HL register ,then go the next number |
| 8 | 100E | 0D |  | DCR C | Decrement the C register ,then check the number |
| 9 | 100F | CA |  | JZ END | If end ,nothing to do ,store this |
| 1010 | 1D |
| 1011 | 10 |
| 10 | 1012 | 7E | LOOP | MOV A,M | Move the memory address to accumulator |
| 11 | 1013 | B8 |  | CMP B | Compare against current minimum |
| 12 | 1014 | DA |  | JC SKIP | If B>A, do nothing |
| 1015 | 18 |
| 1016 | 10 |
| 13 | 1017 | 47 |  | MOV B,A | Move accumulator to B register, load a new minimum |
| 14 | 1018 | 23 | SKIP | INX H | Increase the HL ,then go to the next number |
| 15 | 1019 | 0D |  | DCR C | Decrement the C register ,then check the number |
| 16 | 101A | C2 |  | JNZ LOOP | If numbers left, continue checking |
| 101B | 12 |
| 101C | 10 |
| 17 | 101D | 78 | END | MOV A, B | Move B register to accumulator [A=Minimum] |
| 18 | 101E | 32 |  | STA 204E H | Store the minimum |
| 101F | 4E |
| 1020 | 20 |
| 19 | 1021 | 76 |  | HLT | Stop the program |

Simulator with loaded machine code:



Sample input and output (M[204F H]=06 H, 6 numbers 6A,10,45,7E,B6, and DF,Starting from M[2050 H],

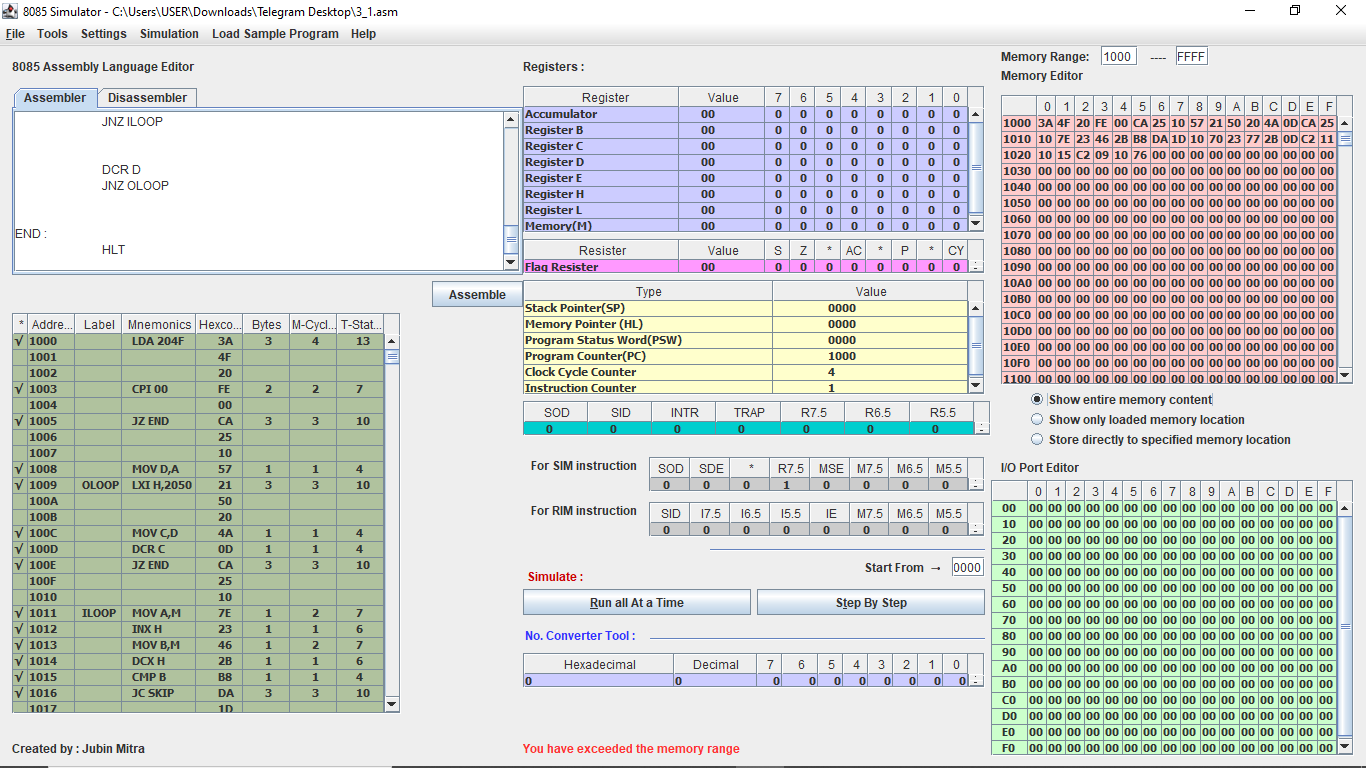
M[204E H]=10

iii)Sort the N numbers in ascending order.

We will be using bubble sort algorithm.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 3A |  | LDA 204F H | Get N |
| 1001 | 4F |
| 1002 | 20 |
| 2 | 1003 | FE |  | CPI 00 H | Is N =? |
| 1004 | 00 |
| 3 | 1005 | CA |  | JZ END | Is N =0,nothing to do |
| 1006 | 25 |
| 1007 | 10 |
| 4 | 1008 | 57 |  | MOV D,A | D=outer loop counter |
| 5 | 1009 | 21 | OLOOP | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| 100A | 50 |
| 100B | 20 |
| 6 | 100C | 4A |  | MOV C,D | C=inner loop counter |
| 7 | 100D | 0D |  | DCR C | Numbers of comparisons is 1 less than the length |
| 8 | 100E | CA |  | JZ END | If no comparisons are to be made ,do nothing |
| 100F | 25 |
| 1010 | 10 |
| 9 | 1011 | 7E | ILOOP | MOV A,M | Get first number |
| 10 | 1012 | 23 |  | INX H | Go to next number |
| 11 | 1013 | 46 |  | MOV B,M | Get second number |
| 12 | 1014 | 2B |  | DCX H | Go back to current position |
| 13 | 1015 | B8 |  | CMP B | Compare 2nd number against 1st number |
| 14 | 1016 | DA |  | JC SKIP | If 2nd number>1st number, do nothing |
| 1017 | 1D |
| 1018 | 10 |
| 15 | 1019 | 70 |  | MOV M,B | Put 2nd number first |
| 16 | 101A | 13 |  | INX H | Go to next location |
| 17 | 101B | 77 |  | MOV M,A | Put 1st number second |
| 18 | 101C | 2B |  | DCX H | Go back to previous location |
| 19 | 101D | 0D | SKIP | DCR C | Comparison done |
| 20 | 101E | C2 |  | JNZ ILOOP | Start from the next location |
| 101F | 11 |
| 1020 | 10 |
| 21 | 1021 | 15 |  | DCR D | One pass finished |
| 22 | 1022 | C2 |  | JNZ OLOOP | Go to the next pass |
| 1023 | 09 |  |
| 1024 | 10 |  |
| 23 | 1025 | 76 |  | HLT | Stop the program |

Simulator with loaded machine code:

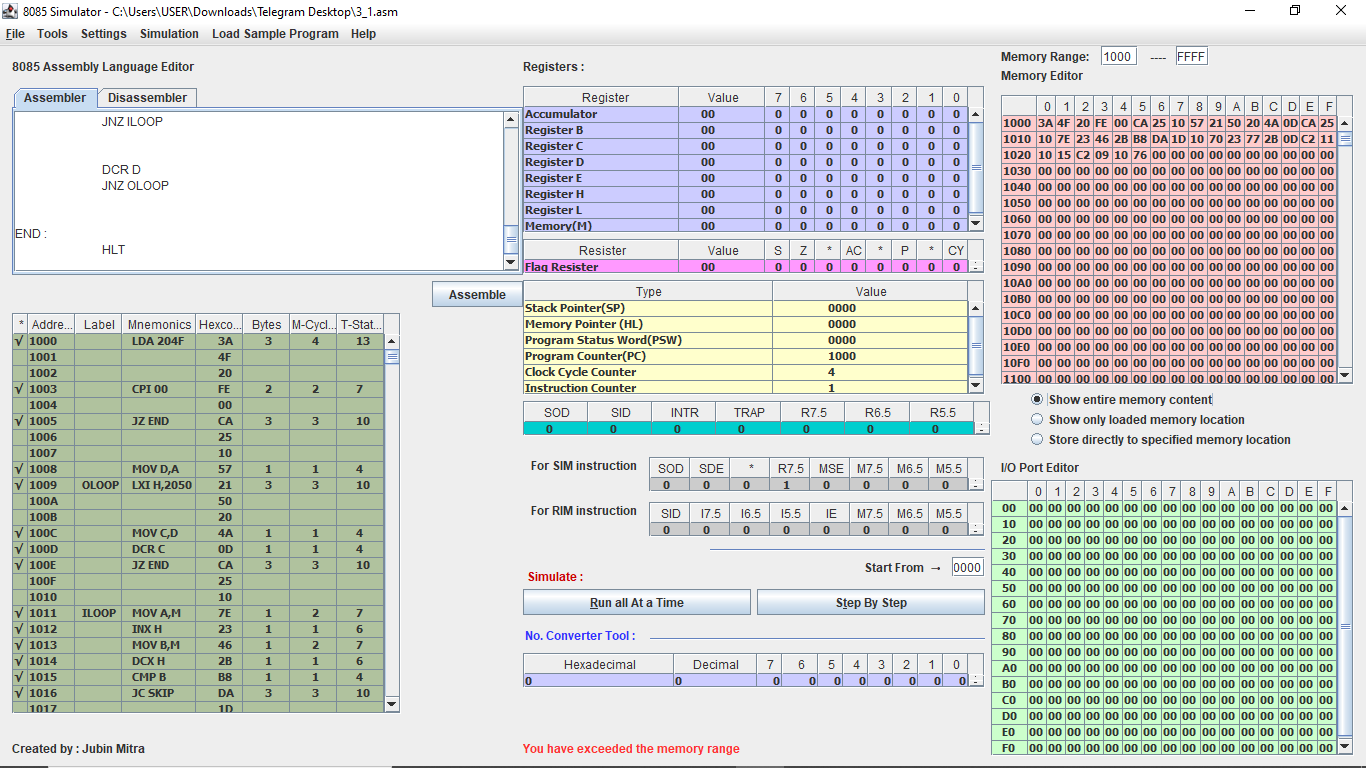


iv)Sort the N numbers in descending order.

We will be using bubble sort algorithm.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 3A |  | LDA 204F H | Get N |
| 1001 | 4F |
| 1002 | 20 |
| 2 | 1003 | FE |  | CPI 00 H | Is N =? |
| 1004 | 00 |
| 3 | 1005 | CA |  | JZ END | Is N =0,nothing to do |
| 1006 | 25 |
| 1007 | 10 |
| 4 | 1008 | 57 |  | MOV D,A | D=outer loop counter |
| 5 | 1009 | 21 | OLOOP | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| 100A | 50 |
| 100B | 20 |
| 6 | 100C | 4A |  | MOV C,D | C=inner loop counter |
| 7 | 100D | 0D |  | DCR C | Numbers of comparisons is 1 less than the length(D) |
| 8 | 100E | CA |  | JZ END | If no comparisons are to be made ,do nothing |
| 100F | 25 |
| 1010 | 10 |
| 9 | 1011 | 7E | ILOOP | MOV A,M | Get first number |
| 10 | 1012 | 23 |  | INX H | Go to next number |
| 11 | 1013 | 46 |  | MOV B,M | Get second number |
| 12 | 1014 | 2B |  | DCX H | Go back to current position |
| 13 | 1015 | B8 |  | CMP B | Compare 2nd number against 1st number |
| 14 | 1016 | DA |  | JC SKIP | If 2nd number<1st number, do nothing |
| 1017 | 1D |
| 1018 | 10 |
| 15 | 1019 | 70 |  | MOV M,B | Put 2nd number first |
| 16 | 101A | 13 |  | INX H | Go to next location |
| 17 | 101B | 77 |  | MOV M,A | Put 1st number second |
| 18 | 101C | 2B |  | DCX H | Go back to previous location |
| 19 | 101D | 0D | SKIP | DCR C | Comparison done |
| 20 | 101E | C2 |  | JNZ ILOOP | Start from the next location |
| 101F | 11 |
| 1020 | 10 |
| 21 | 1021 | 15 |  | DCR D | One pass finished |
| 22 | 1022 | C2 |  | JNZ OLOOP | Go to the next pass |
| 1023 | 09 |  |
| 1024 | 10 |  |
| 23 | 1025 | 76 |  | HLT | Stop the program |

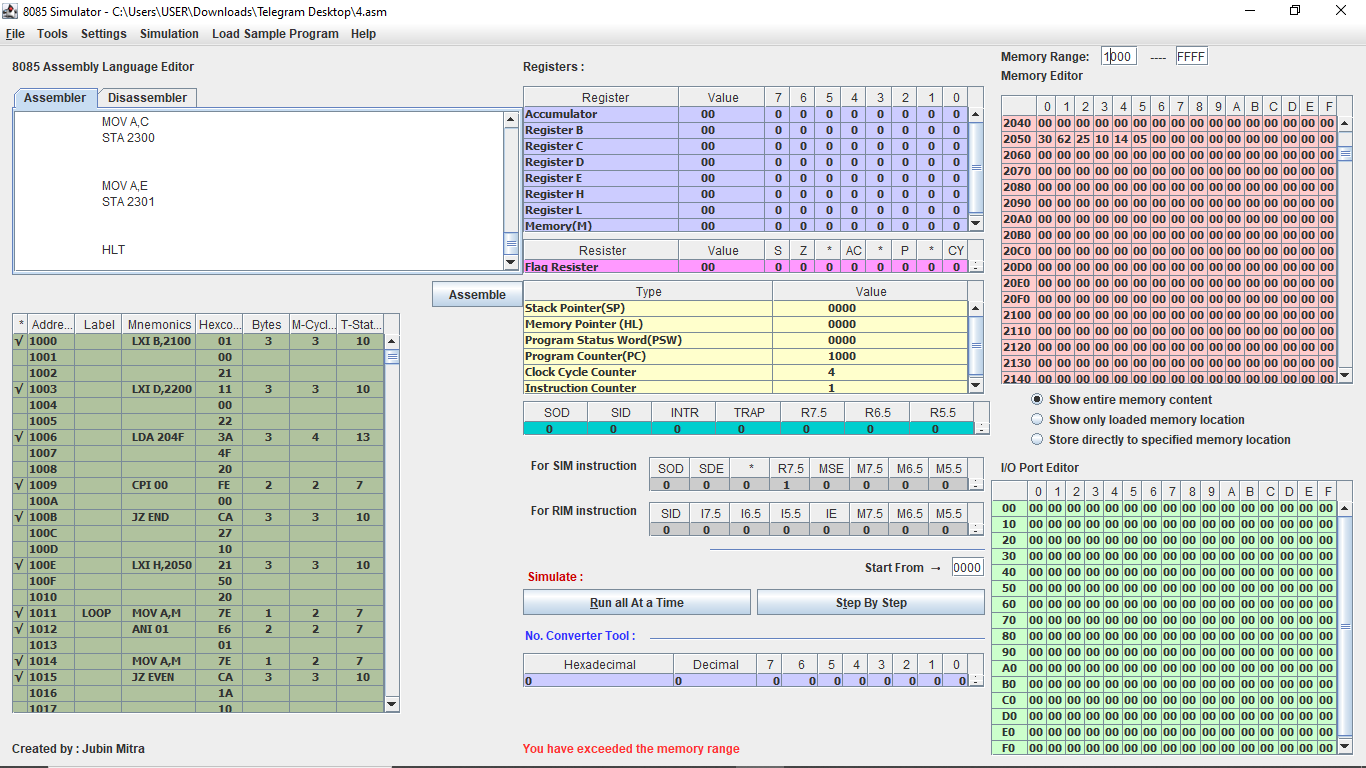
Simulator with loaded machine code:



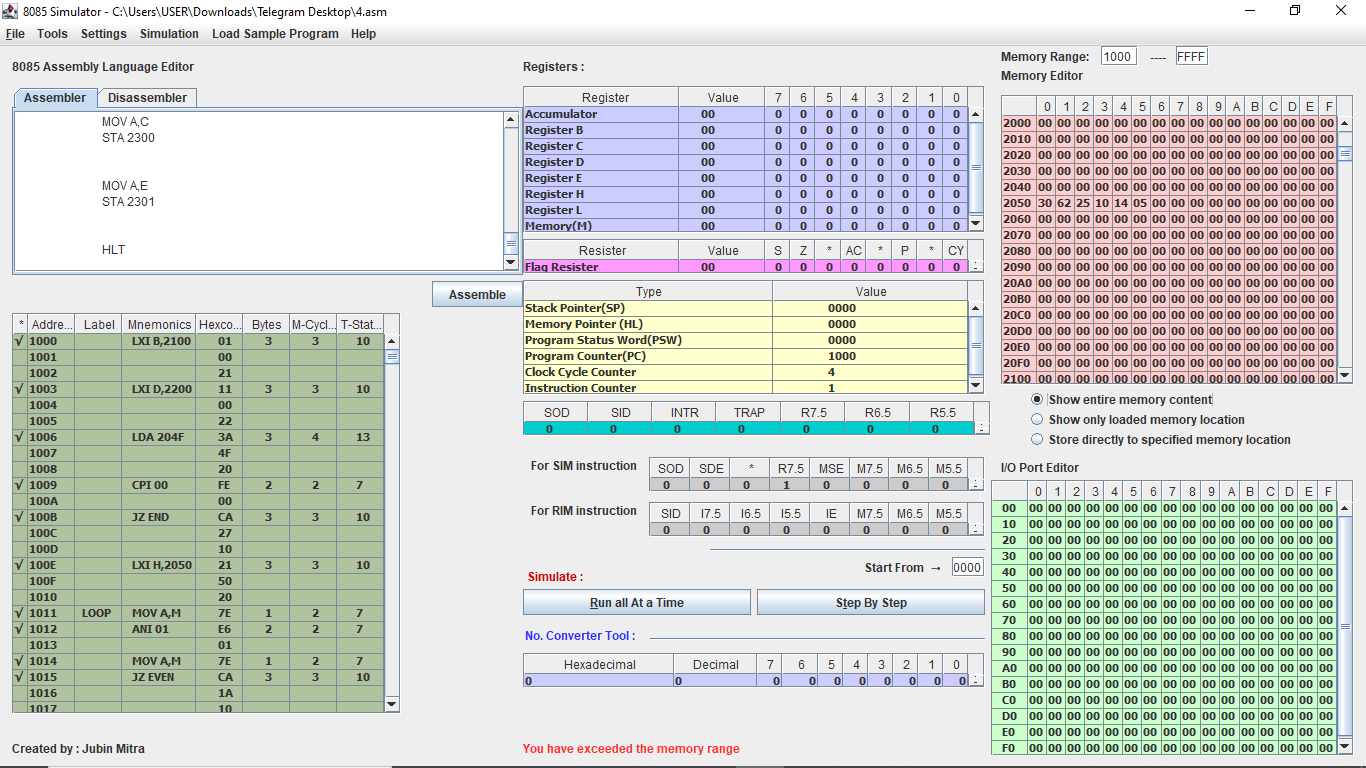
4.N numbers are stored in consecutive m/m location startingfrom 2050 H. The value N is stored in 204F H. Write a program to copy the even and odd numbers starting from 2100 H and 2200 H ,respectively. Store the total num of even and odd numbers in 2300 H and 2301 H , respectively

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 01 |  | LXI B , 2100 | Starting address of even numbers |
| 1001 | 00 |
| 1002 | 21 |
| 2 | 1003 | 11 |  | LXI D,2200 | Starting address of odd numbers |
| 1004 | 00 |
| 1005 | 22 |
| 3 | 1006 | 3A |  | LDA 204F | Get N |
| 1007 | 4F |
| 1008 | 20 |
| 4 | 1009 | FE |  | CPI 00 H | Is N =0? |
| 100A | 00 |
| 5 | 100B | CA |  | JZ END | If N=0, nothing to do |
|  | 100C | 27 |  |  |  |
| 100D | 10 |
| 6 | 100E | 21 |  | LXI H ,2050 | Contents of memory location 2050 H into HL register pair/input |
| 100F | 50 |
| 1010 | 20 |
| 7 | 1011 | 7E | LOOP | MOV A, M | Get current numbers |
| 8 | 1012 | E6 |  | ANI 01 | Check odd or not |
| 1013 | 01 |
| 9 | 1014 | 7E |  | MOV A, M | Restore the number |
| 10 | 1015 | CA |  | JZ EVEN | If 0, then even |
| 1016 | 1A |
| 1017 | 10 |
| 11 | 1018 | 12 |  | STAX D | This is an odd number |
| 12 | 1019 | 13 |  | INX D | One odd number added |
| 13 | 101A | 02 | EVEN | STAX B | This is an even number |
| 14 | 101B | 03 |  | INX B | One even number added |
| 15 | 101C | 23 | LEND | INX H | Go to the next address |
| 16 | 101D | 3A |  | LDA 204F | Get n |
| 101E | 4F |
| 101F | 20 |
| 17 | 1020 | 3D |  | DCR A | One number checked |
| 18 | 1021 | 32 |  | STA 204F | Store N for later use |
| 1022 | 4F |
| 1023 | 20 |
| 19 | 1024 | C2 |  | JNZ LOOP | If numbers left ,continue |
| 1025 | 11 |
| 1026 | 10 |
| 20 | 1027 | 79 | END | MOV A,C | A=C=number of even number because BC started from 2100 H |
| 21 | 1028 | 32 |  | STA 2300 H | Store numbers of even number |
| 1029 | 00 |
| 102A | 23 |
| 22 | 102B | 7B |  | MOV A,E | A=E=number of odd numbers because DE started from 2200 H |
| 23 | 102C | 32 |  | STA 2301 H | Store number of odd numbers |
| 102D | 01 |
| 102E | 23 |
| 24 | 102F | 76 |  | HLT | Stop the program |

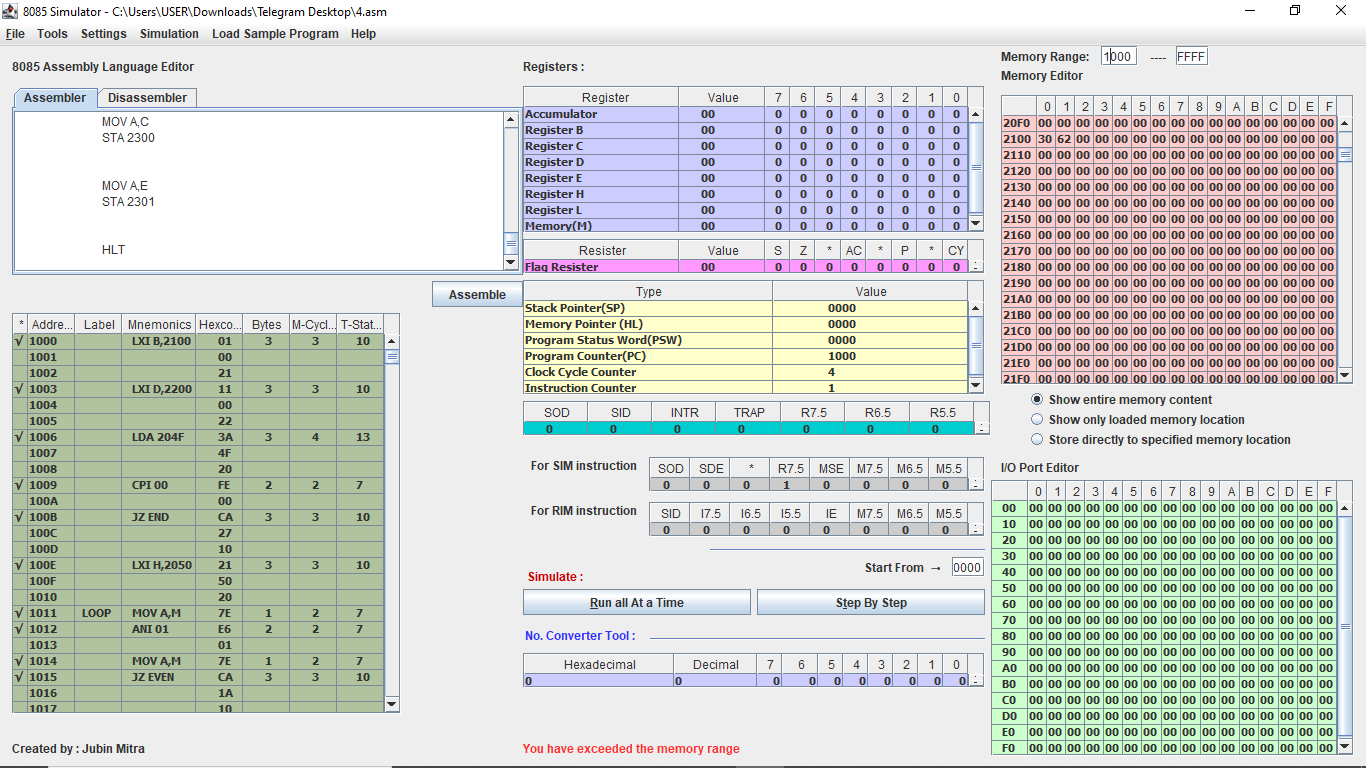
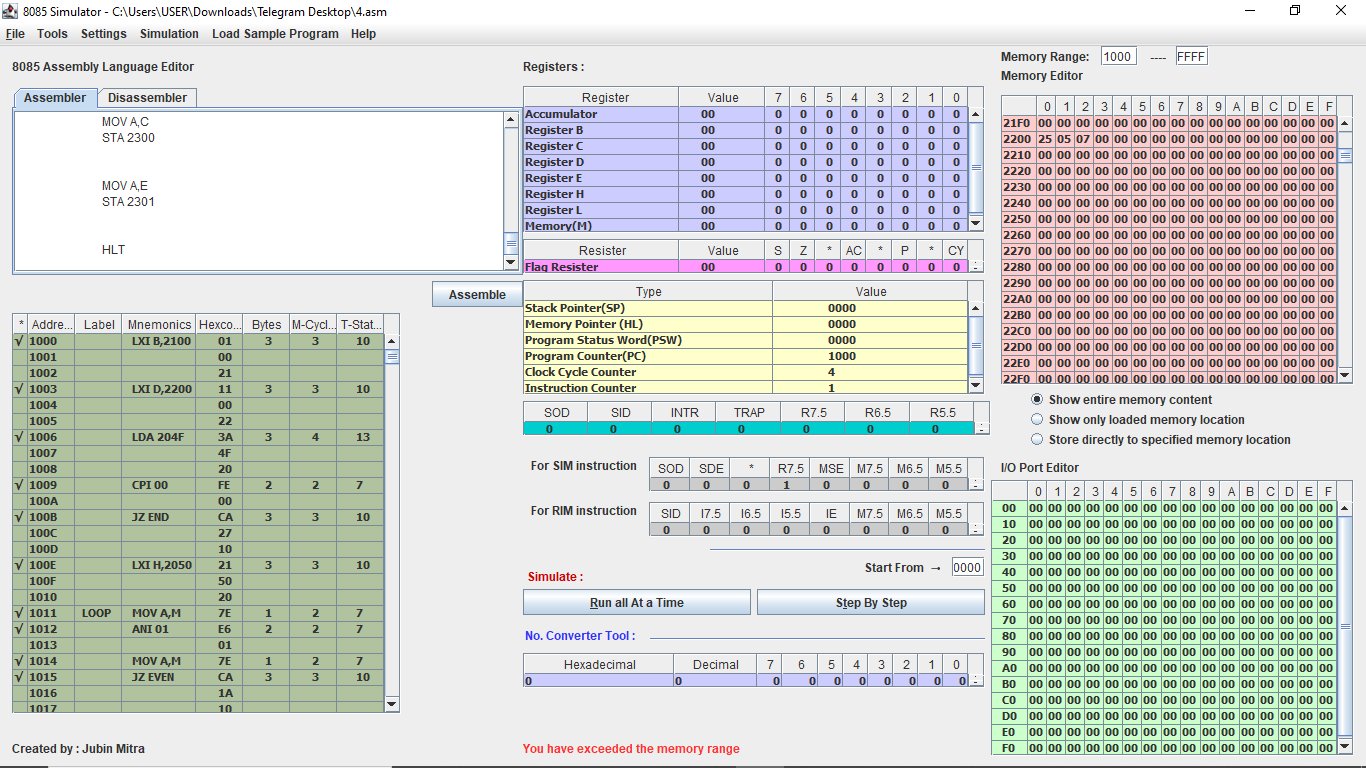
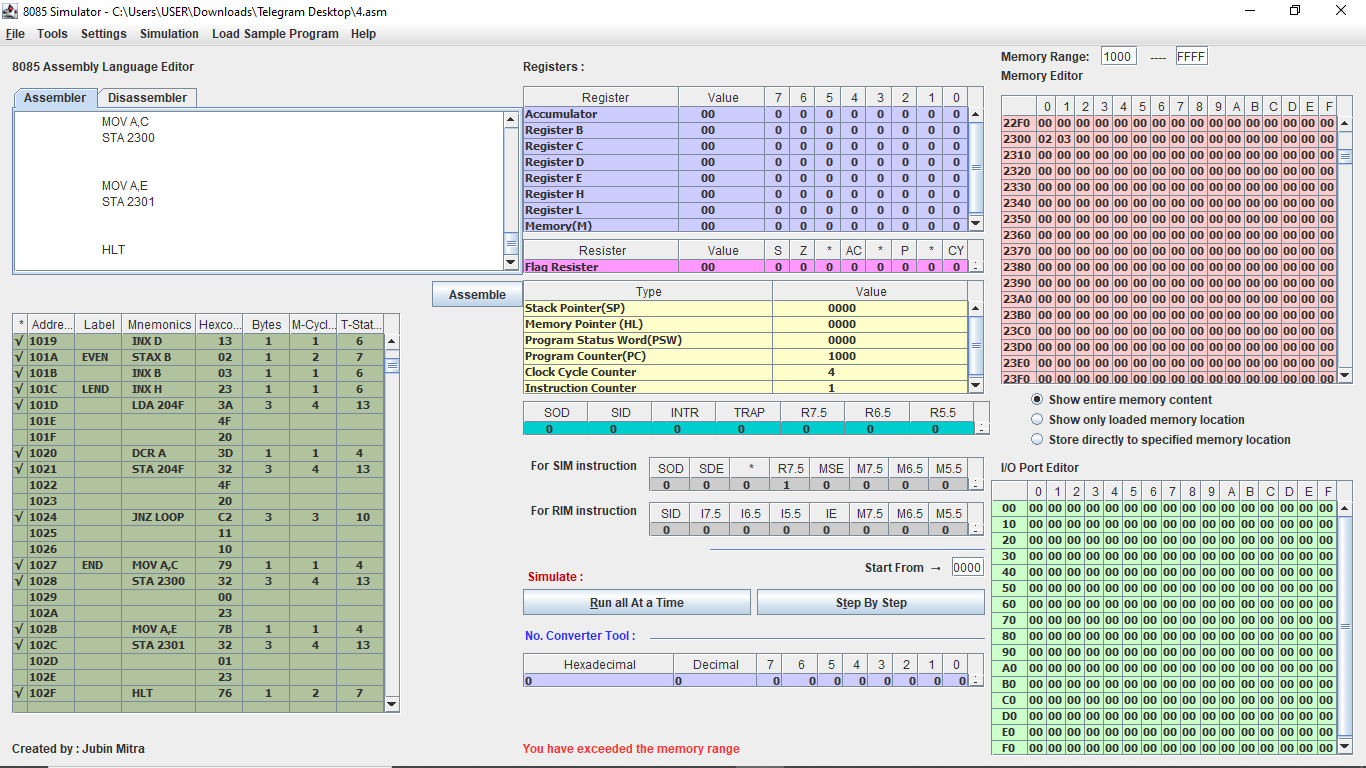
Simulator with loaded machine code:



Sample input (M[204F H]=06 H, 6 numbers 30 H,62 H,25 H,10 H,24 H and 05 H starting from M[2050 H]):



Sample output (2 even numbers 30 H, 62 H starting from M[2100 H], 3 odd numbers 25 H,05 H, 07 H starting from M[2200 H], M[2300 H]=02 H,M[2301]=03 H):

5. N numbers are stored in consecutive m/m location starting from 2050 H. The value N is stored in 204F H. Write a program to test whether a number stored in 204E H is present in the list. If present , store its position in the list at 204D H ;otherwise store FF H.

We will give position by a zero – based index.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO.** | **ADDRESS** | **OPCODE IN HEX CODE** | **LABEL** | **INSTRUCTIONS** | **COMMENTS** |
| 1 | 1000 | 21 |  | LXI H, 204E | B=Numbers to search (key) |
| 1001 | 4E |
| 1002 | 20 |
| 2 | 1003 | 46 |  | MOV B,M | Move memory location to B register |
| 3 | 1004 | 23 |  | INX H | HL=204F H |
| 4 | 1005 | 7E |  | MOV A,M | Get N |
| 5 | 1006 | FE |  | CPI 00 H | Is N=0? |
| 1007 | 00 |
| 6 | 1008 | CA |  | JZ END | If N=0,do not search |
| 1009 | 1E |
| 100A | 10 |
| 7 | 100B | 4F |  | MOV C,A | Move accumulator to C register [C=A=Counter] |
| 8 | 100C | 51 |  | MOV D,C | Move C register to D register[D=total no of items] |
| 9 | 100D | 23 |  | INX H | HL=2050 H=Starting address of input |
| 10 | 100E | 7E | LOOP | MOV A,M | Move memory address to accumulator, load current number |
| 11 | 100F | B8 |  | CMP B | Compare against key |
| 12 | 1010 | C2 |  | JNZ CONT | If unequal ,continue loop |
| 1011 | 19 |
| 1012 | 10 |
| 13 | 1013 | 7A |  | MOV A,D | A=D=Totals number of items |
| 14 | 1014 | 91 |  | SUB C | A=position of key in list |
| 15 | 1015 | 32 |  | STA 204D | Store found position |
| 1016 | 4D |
| 1017 | 20 |
| 16 | 1018 | 76 |  | HLT | We have nothing to do anymore |
| 17 | 1019 | 23 |  | INX H | Go to next number |
| 18 | 101A | 0D |  | DCR C | Checked one number |
| 19 | 101B | C2 |  | JNZ LOOP | If number left,continue |
| 101C | 0E |
| 101D | 10 |
| 20 | 101E | 3E | END | MVI A,FF | We haven’t found key |
| 101F | FF |
| 21 | 1020 | 32 |  | STA 204D | Store the number |
| 1021 | 4D |
| 1022 | 20 |
| 22 | 1023 | 76 |  | HLT | Stop the program. |

Simulator with loaded machine code:

