Tutorial 4

Solution to Problem 2:

Line Number	Instructions	Notes
1	LOAD 0, R11	Load first value of Fibonacci series in first register
2	LOAD 1, R12	Load second value of Fibonacci series in first register
3	MOV R21, R11	Initialize R21 with first value
4	MOV R22, R12	Initialize R22 with second value
5	LOAD 12, R5	R5 would point to current register for which we are calculating value. Since we have filled R12, its value is initialized to 12
6	LOAD 20, R6	R6 puts an upper limit to index register R5
7	INC R5	From here starts our loop. Increment R5 to point to next register to be filled in
8	ADD R23, R22, R21	Add 2 values in R21 to R22 to get next value in the series
9	MVI R(R5), R23	Store the newly calculated value in the register pointed to by the index register
10	MOV R21, R22	Update R21 with next value
11	MOV R22, R23	Update R22 with next value
12	CMP R5, R6	Check if we are done filling last register
13	If Lx = 0 Jump 7	If not, lets keep calculating

Solution to Problem 3:

Line Number	Instructions	Notes
1	LOAD 11, R5	R5 would hold index of the register we are currently checking for uniqueness
2	LOAD 15, R6	R6 serves as the upper limit
3	LOAD 5, R10	R10 stores the final result. Assuming initially that all 5 values are uniq. As we get to know there exists a duplicate value
4	LOAD 1, R8	Just holding value 1 for decrementing since we don't have a DEC function
5	MOV R7, R5	R7 would act as iterator to check values against the value currently being checked. Initializing it with current R5, to against only values after the index
6	MOV R20, R(R5)	R20 holds value that is currently tested
7	INC R7	From here starts the inner loop to check against every value after our current value
8	MOV R21, R(R7)	R21 holds the value which we are checking against current value
9	SUB R22, R21, R20	To check for equality, we check if the difference of the 2 numbers is 0
10	If L2=1 Jump 14	If L2 flag is set, it indicates result was 0, hence they are same. Thus we have found a duplicate value and hence we should decrement R10 and conclude current value is not unique and hence proceed with next.

11	CMP R7, R6	If this value is not same, we need to check we are done checking all the values
12	If L3 = 0, Jump 7	If we have not yet completed all values, lets continue the loop
13	If L3 = 1, Jump 15	If we have checked all the values, lets test next R5
14	SUB R10, R8, R10	Decrementing number of unique values
15	INC R5	Incrementing for next register index value
16	CMP R5, R6	Checking if we are done with all the values
17	If L3 = 0, Jump 5	Keep looping if not all values checked

Solution to Problem 4:

#	Instructions	State Machine State	Data Bus	A4	A3	A2	A1	A0	RD	WR	LD	AI	C1	C0	Flag Register
A	INC R5	MOV R5 to R0	xxxx xxxx	0	0	1	0	1	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	0	0	0	0	1	0	0	X	X	xxxx xxxx
		Set ALU to INC	xxxx xxxx	X	X	X	X	X	0	0	0	0	1	1	xxxx xxxx
		MOV R2 to R5	xxxx xxxx	0	0	0	1	0	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	1	0	1	0	1	0	0	X	X	xxxx xxxx
B	ADD R23, R22, R21	MOV R22 to R0	xxxx xxxx	1	0	1	1	0	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	0	0	0	0	1	0	0	X	X	xxxx xxxx
		MOV R21 to R1	xxxx xxxx	1	0	1	0	1	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	0	0	1	0	1	0	0	X	X	xxxx xxxx
		Set ALU to ADD	xxxx xxxx	X	X	X	X	X	0	0	0	0	0	0	xxxx xxxx
		MOV R2 to R23	xxxx xxxx	0	0	0	1	0	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	1	0	1	1	1	0	1	0	0	X	X	xxxx xxxx
C	MVI R(R5), R2	Get data in R(R5) on data bus	xxxx xxxx	0	0	1	0	1	1	0	0	1	X	X	xxxx xxxx
		Write to R2	xxxx xxxx	0	0	0	1	0	0	1	0	0	X	X	xxxx xxxx

D	SUB R21, R22, R21	MOV R21 to R0	xxxx xxxx	1	0	1	0	1	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	0	0	0	0	1	0	0	X	X	xxxx xxxx
		MOV R22 to R1	xxxx xxxx	1	0	1	1	0	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	0	0	0	0	1	0	1	0	0	X	X	xxxx xxxx
		Set ALU to SUB	xxxx xxxx	X	X	X	X	X	0	0	0	0	1	0	xxxx xxxx
		MOV R2 to R23	xxxx xxxx	0	0	0	1	0	1	0	0	0	X	X	xxxx xxxx
			xxxx xxxx	1	0	1	1	1	0	1	0	0	X	X	xxxx xxxx