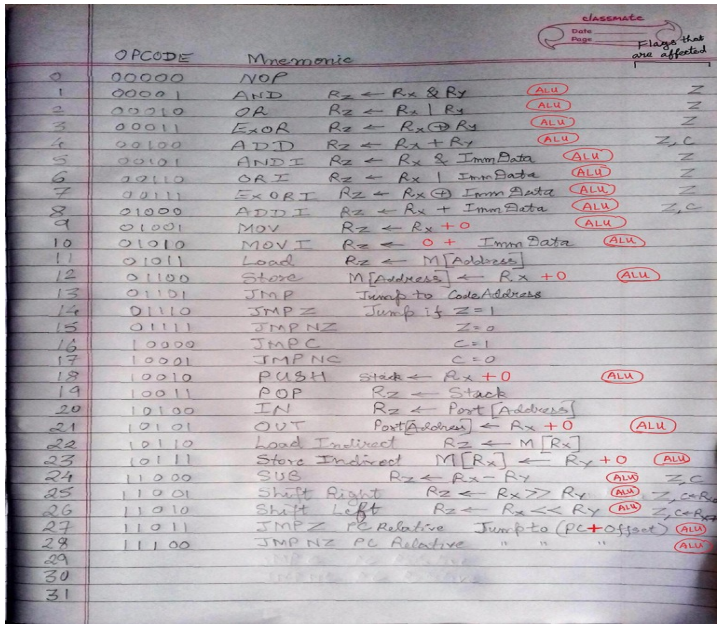

Sr. No.	OPCODE	
0	00000	NOP
1	00001	AND
2	00010	OR
3	00011	ExOR
4	00100	ADD
5	00101	ANDI
6	00110	ORI
7	00111	ExORI
8	01000	ADDI
9	01001	MOV
10	01010	MOVI
11	01011	Load
12	01100	Store
13	01101	JMP
14	01110	JMPZ
15	01111	JMPNZ
16	10000	JMPC
17	10001	JMPNC
18	10010	PUSH
19	10011	POP
20	10100	IN
21	10101	OUT
22	10110	Load Indirect
23	10111	Store Indirect
24	11000	SUB
25	11001	Shift Right
26	11010	Shift Left
27	11011	JMPZ Relative
28	11100	JMPNZ Relative
29	11101	
30	11110	
31	11111	

How to write in assembler? (Example given below)	Comments
NOP;	
AND, R12, R2, R5;	
OR, R3, R15, R6;	
EXOR, R3, R15, R6;	
ADD, R13, R15, R2;	
ANDI, R3, R15, 77;	
ORI, R3, R15, 88;	
EXORI, R3, R15, AB;	
ADDI, R10, R15, FC;	
MOV, R12, R7;	
MOVI, R14,AC;	
LOAD, R10,81;	Means LoadI R10 <--- M[81H]
STORE, 83,R10;	Means LoadI M[83H] <---R10
JMP, AB;	Means JMP to address ABH in the instruction mememory
JMPZ, AC;	
JMPNZ, AD;	
JMPC, AE;	
JMPNC, AF;	
PUSH, R12;	Means PUSH Stack <--- R12
POP, R13;	Means POP R13<--Stack
IN, R11,F1;	Means IN R11 <--- P[F1H]
OUT, F2,R14;	Means OUT P[F2H]<--R14
LOADI, R10, R2;	Means LoadI R10 <--- M[R2]
STOREI, R10, R2;	Means StoreI M[R2]<--R10
SUB, R12, R13, R14;	
SHIFTR, R12, R13, R14;	Means Shift Right R12<--R13>>R14
SHIFTL, R12, R13, R14;	
JMPPCRZ, AD;	
JMPPCRNZ, AE;	