Machine Codes

Instruction	Machine code
Int number	11001101 number Nop:10010000
Mov reg, number	1011wreg number w:0 for byte 1 for word
Mov DH,23 is (10110110)(23)	Mov CX,1000 is (10111001)(232)(3) Hint:1000=3*256+232

000	001	010	011	100	101	110	111
AL AX	CL CX	DL DX	BL BX	AH SP	CH BP	DH SI	BH DI

.model small
.code
Mov AH,2
Mov DL,70
Int 33
 	
Mov AH,76
Int 33
End
	.model small
	.code
	Mov AH,2
	Mov DL,70
	db 11001101b
	db 33d
	Mov AH,76
	Int 33
	End
		Mov AH,2
		Mov DL,65
		L:Int 33
		Inc DL
		Cmp DL,70
		JL L
		
		Mov AH,76
		Int 33
			Mov AH,2
			Mov DL,65
			L:Int 33
			Inc DL
			Cmp DL,70
			db 01111100b
			db -9
			Mov AH,76
			Int 33		

Add reg, number	100000sw 11000reg number [w=0 $\rightarrow$ s=0][w=1, -128 $\leq$ num<128 $\rightarrow$ s=1]
Sub reg, number	100000sw 11101reg number Cmp reg,num:100000sw 11111reg number
Mov reg_1,reg_2	1000101w 11reg_1,reg_2 Add:0000001w Sub:0010101w Cmp:0011101w
Mul reg	1111011w 11100reg Div:11110reg imul:11101reg idiv:11111reg
Jmp(short) 11101011 disp	Jmp (long within segment) 11101001 offset (two bytes) Call:11101000
Jmp conditional: 0111xxxx [Example JL=01111100]	JNL=JGE JZ=JE JC=JB

JO:0000	JB:0010	JE:0100	JBE:0110	JS:1000	JP:1010	JL:1100	JLE:1110
JNO:0001	JAE:0011	JNE:0101	JA:0111	JNS:1001	JNP:1011	JGE:1101	JG:1111

Mov reg,r/m	100010dw mdregr/m Add:000000dw Sub:001010dw Cmp:001110dw
d:0 for opposite 1 for correct	md: 11(reg) 00(memory) 01(mem+byte) 10(mem+word)

000	001	010	011	100	101	110 md00	110 md01,10	111
[BX+SI]	[BX+DI]	[BP+SI]	[BP+DI]	[SI]	[DI]	[Direct]	[BP]	[BX]

Mov BL,[SI] is (10001010)(00011100)	Mov [SI],BL is (10001000)(00011100) d changes
Mov BX,[SI] is (10001011)(00011100)	Mov BL,[SI+52] is (10001010)(01011100)(52)
Mov r/m,number	1100011w md000r/m number
Add r/m,number	100000sw md000r/m number sub:(2^{nd} byte)md101r/m cmp:md111r/m
Inc r/m	1111111w md000r/m Dec(2^{nd} byte):md001r/m
Mov seg-reg,r/m	100011d0 md0sgr/m sg is CS,DS,ES,SS m/c 01,11,00,10 respectively.
Push r/m	11111111 md110r/m Pop: 10001111 md000r/m
Push seg-reg	000sg110 Pop:000sg111 PushF:10011100 PopF:10011101
Neg	1111011w md011r/m Not:md010r/m
Shift/rotate r/m,_	110100cw mdxyzreg [Shr bh,1 has cxyz=0101][Rcl dl,cl 1010][Sar y=1]
c=0:count=1 c=1:CL	x=0:rotate y=0:logical shift or rotate without carry z=0:left
When r/m is used in place of reg, first two bits of second byte become md in many instructions.	
Following Instructions provide efficiency.	
Add AL, number	0000010w number (AX can be used) Sub:0010110w Cmp:0011110w
Mov AL,memory	1010000w addr Mov memory,AL: 1010001w addr (AX may be used)
Inc reg (word) 01000reg Dec:01001reg	Push reg 01010reg Pop: 01011reg
Add BX,50 (10000011)(11000011)(50)	Add BX,500 (10000001)(11000011)(244)(1)

When segment register for memory is other then DS (default) then 001sg110 is put as first byte. Mov BL,ES:[BX+SI] (00100110)(10001010)(00011000) (For BP the default is SS)		
Add [BX+40],word ptr 50 (10000011)(01000111)(40)(50) Add [BX+400],word ptr 50 (10000011)(10000111)(144)(1)(50) Add [BX+40],word ptr 500 (10000001)(01000111)(40)(244)(1) Add [BX+40],byte ptr 50 (10000000)(01000111)(40)(50) Mov DL,DS:[200] (10001010)(00010110)(11001000)(0) Mov DL,SS:[BP+200] (10001010)(10010110)(11001000)(0)		
Jmp (intersegment)	11101010 offset(2 byte) seg(2 byte)	Call:10011010
Jmp (indirect within segment)	11111111 md100r/m Call:md010r/m	Intersegment 101,011
Ret: 11000011(within segment)	Ret: 11001011(intersegment)	

1. Program to print 'G'. Do not use Int in that program.
2. Program to print 'F'. Do not use DL in that program.
3. Program that reads a letter and prints the next letter. Do not use AL.
4. Program that reads a letter and prints the next letter. Do not use Add or Inc.
5. Read a letter. Output A is ascci code is bigger than 100. B otherwise. No Cmp.
6. Read two letters. Print bigger. No Cmp.
7. Read two letters. Print bigger. No Jmp. Use Int 33 only 3 times.
8. Read two letters. Print bigger. No jump of any type (JL, JG etc should not be used).
9. Print ABCD….Z. No jump of any type (JL, JG etc should not be used).