***Name:- Mul without mul inst. 8086***

MOV AX,1000H

MOV DS,AX

MOV SI,2000H

MOV BX,2002H

MOV CX,[BX]

MOV DX,0000H

MOV BX,SI

MOV AX,0000H

L2:ADD AX,[BX]

JNC L1

INC DX

L1:DEC CX

JNZ L2

INC BX

INC BX

MOV [BX],AX

INC BX

INC BX

L3:MOV [BX],DX

HLT

***Name:- Div without div inst 8086***

MOV AX,1000H

MOV DS,AX

MOV BX,2000H

MOV AX,[BX]

INC BX

INC BX

MOV CX,0000H

L2: CMP AX,[BX]

JC HLT1

SUB AX,[BX]

INC CX

JMP L2

HLT1:INC BX

INC BX

MOV [BX],AX

INC BX

INC BX

MOV [BX],CX

HLT

***Name:- Avg. of n 16bit no.***

MOV AX,1000H

MOV DS,AX

MOV SI,5000H

MOV CL,[SI]

MOV BL,CL

MOV AX,0000H

MOV DX,0000H

L2:INC SI

INC SI

ADD AX,[SI]

JNC L1

INC DX

L1:DEC CL

JNZ L2

INC SI

INC SI

MOV [SI],AX

INC SI

INC SI

MOV [SI],DX

DIV BX

INC SI

INC SI

MOV [SI],AX

INC SI

INC SI

MOV [SI],DX

HLT

***Name:- No of 1’s in 8bit no***

MOV AX,1000H

MOV DS,AX

MOV DL,00H

MOV CL,00H

MOV AL,[5000H]

L2:MOV BL,AL

AND AL,01H

JZ L1

INC DL

L1:SHR AL,01H

DEC CL

JNZ L2

MOV [5001H],DL

HLT

***Name:- Mov a block of 16 bit data***

INTERSEGMENT

MOV AX,1000H

MOV BX,3000H//

MOV SI,2000H

MOV DI,5002H

MOV DS,AX

MOV CL,[SI]

L1:INC SI

INC SI

MOV DS,AX//

MOV DX,[SI]

MOV DS,BX//

INC DI

INC DI

MOV [DI],DX

DEC CL

JNZ L1

HLT

***Name:- largest among n***

.DATA

COUNT DB 04H

VALUE DB 0FH,09H,14H,10H

RES DB 0

.CODE

MAIN PROC

MOV AX,DATA

MOV DS,AX

MOV CL,COUNT

DEC CL

LEA SI,VALUE

MOV AL,[SI]

UP:INC SI

CMP AL,[SI]

JNC DWN//jc

MOV AL,[SI]

DWN:DEC CL

JNZ UP

LEA DI,RES

MOV [DI],AL

END MAIN

***Name:- Ascending order***

.DATA

COUNT DB 04H

VALUE DB 03H,02H,04H,01H

.CODE

MAIN PROC

MOV AX,DATA

MOV DS,AX

MOV CL,COUNT

DEC CL

UP2:MOV CH,CL

LEA SI,VALUE

UP1:MOV AL,[SI]

CMP AL,[SI+1]

JC DWN

MOV DL,[SI+1]

XCHG [SI],DL

MOV [SI+1],DL

DWN:INC SI

DEC CH

JNZ UP1

DEC CL

JNZ UP2

END MAIN

***Name:- ARM arithmetic and logical***

LDR R0,=0xe10100F00

LDR R1,[R0]

ADD R0,R0,#04

LDR R2,[R0]

ADD R3,R1,R2

ADD R0,R0,#04

STR R3,[R0]

SUB R4,R1,R2

ADD R0,R0,#04

STR R4,[R0]

MUL R5,R1,R2

ADD R0,R0,#04

STR R5,[R0]

AND R6,R1,R2

ADD R0,R0,#04

STR R6,[R0]

ORR R7,R1,R2

ADD R0,R0,#04

STR R7,[R0]

EOR R8,R1,R2

ADD R0,R0,#04

STR R8,[R0]

MVN R9,R1

ADD R0,R0,#04

STR R9,[R0]

MVN R10,R2

ADD R0,R0,#04

STR R10,[R0]

EXIT : B EXIT

***Name :- Largest ARM***

LDR RO,=count

LDR R1,[R0]

LDR R2,=array

LDR R3,[R2],#4

L1:SUB R1,R1,#1

CMP R1,#0

BEQ L2

LDR R4,[R2],#4

CMP R3,R4

BGT L1//blt

MOV R3,R4

B L1

L2:STR R3,[R2]

EXIT : B EXIT

.data

count:.word 0x05

array:.word 0x05,0xF1,0xAB,0x35,0x45

result:.word 0

***Name:- Separate Odd Even***

LDR RO,=count

LDR R1,[R0]

LDR R2,=array

LDR R3,=even

LDR R4,=odd

L3:LDR R5,[R2],#4

AND R6,R5,#1

CMP R6,#0

BEQ L1

STR R5,[R4],#4

B L2

L1:STR R5,[R3],#4

L2:SUB R1,R1,#1

CMP R1,#0

BNE L3

EXIT : B EXIT

.data

count:.word 0x05

array:.word 0x35,0xF1,0xAC,0x10,0x45

even:.word 0,0,0,0,0

odd:.word 0,0,0,0,0