1.PROGRAM TO ADD TWO 8-BIT DATA

MVI C,00

LDA 4200H

MOV B, A

LDA 4201H

ADD B

JNC SKIP

INR C

SKIP: STA 4202H

MOV A,C

STA 4203H

HLT

2. 16-BIT ADDITION WITH CARRY

MVI B,00

LHLD 4200H

XCHG

LHLD 4202H

DAD D

JNC SKIP

INR B

SKIP: SHLD 4204H

MOV A, B

STA 4206H

HLT

3. 8-BIT SUBTRACTION WITH BORROW USING DIRECT ADDRESSING

LDA 4201H

MOV B, A

LDA 4200H

SUB B

JNC SKIP

INR C

CMA

INR A

SKIP: STA 4202H

MOV A,C

STA 4203H

HLT

4.8-BIT MULTIPLICATION OPERATIONS USING 8085 MICROPROCESSOR

LDA 4200H

MOV E,A

LDA 4201H

MOV B,A

LXI H,0000H

MVI D,00H

NEXT: DAD D

DCR B

JNZ NEXT

SHLD 4202H

HLT

5.8bit division operation using 8085 microprocessor

LDA 4201H

MOV B,A

HDA 4200H

MVI C,00H

AGAIN: CMP B

JC STORE

SUB B

INR C

JMP AGAIN

STORE: STA 4203H

MOV A,C

STA 4202H

HLT

6.SORT AN ARRAY OF DATA IN ASCENDING ORDER

LDA 4200H

MOV B,A

DCR B

LOOP 2 LXI  H,4200H

MOV C,M

DCR C

INX  H

LOOP 1  MOV A,M

INX H

CMP M

JC AHEAD

MOV D,M

MOV M,A

DCX H

MOV M,H

DCX H

MOV M,H

INX H

AHEAD DCR C

JNZ LOOP 1

DCR B

JNZ LOOP 2

HLT

7.SORT AN ARRAY OF DATA IN DESCENDING ORDER

LDA 4200H

MOV B,A

DCR B

LOOP2 LXI H,4200H

MOV C,M

DCR C

INX H

LOOP1 MOV A,M

INX H

CMP M

JNC AHEAD

MOV D,M

MOV M,A

DCX H

MOV M,D

INX H

AHEAD DCR C

JNZ LOOP 1

DCR B

JNZ LOOP 2

HLT

8.SEARCH THE SMALLEST NUMBER FROM AN ARRAY

ORG 4100H

LXI H,4200H

MCV B,M

INX H

MCV A,M

DCR B

LOOP INX H

CMP M

JC AHEAD

MOV A,M

AHEAD DCR B

JNZ LOOP

STA 4300H

HLT

9.SEARCH THE LARGEST NUMBER FROM AN ARRAY

ORG 4100H

LXI H,4200H

MCV B,M

INX H

MCV A,M

DCR B

LOOP: INX H

CMP M

JNC AHEAD

MOV A,M

AHEAD: DCR B

JNZ LOOP

STA 4300H

HLT

10.ASC11 CODE TO HEX CODE CONVERSION

LDA 4200H

SUI 30H

CPI 0AH

JC STORE

SUI 07H

STORE: STA 4201H

HLT

11.MASKING AND SETTING OF LOWER NIBBLES ON GIVEN DATA

LDA 4200H

ANI, 0FH

STA 4201H

HLT

12.ONES AND TWOS COMPLEMENT

LDA  4200H

CMA

STA 4201H

INR A

STA 4202

HLT

13.ADDITION OF 16 BIT NUMBERS WITH CARRY

MOV CX, 0000H

MOV AX, [1200]

MOV BX, [1202]

ADD AX, BX

JNC L1

INC CX

L1: MOV [1206], CX

MOV [1204],AX

HLT

14.SUBTRACTION OF 16 BIT NUMBERS WITH CARRY

MOV CX,0000H

MOV AX, [1300]

MOV BX,[1302]

SUB AX, BX

JNC SKIP

INC CX

NEG AX

SKIP: MOV [1306],CX

MOV [1304],AX

HLT

15.MULTIPLICATION OF 16 BIT NUMBERS

MOV AX, [1200]

MOV BX,[1202]

MUL BX

MOV [1204],AX

MOV [1206],DX

HLT

16.DIV OF 32 BIT BY 16 BIT NUMBERS

MOV DX,[1200H]

MOV AX,[1202]

MOV BX, [1204]

DIV CX

MOV [1206],AX

MOV [1209],DX

HLT

17.LOGICAL OPERATION

MOV AX,[1200]

AND AX, OFOFH

MOV [1202],AX

HLT

18.MOV DATA BLOCK WITHOUT OVERLAP

MOV SI, 1150H

MOV DI,1250H

MOV CX,0006 H

REPEAT :CLD

MOVSB

LOOP REPEAT

HLT

19.SUM OF N NUMBER IN WORD OF ARRAY

MOV DX,OH

MOV SI,1250H

MOV CX,03H

MOV AX, [SI]

A1:INC SI

INC SI

ADD AX, [SI]

JNC NEXT

INC DX

NEXT: LOOP A1

MOV [1300H], AX

MOV [1302H], DX

HLT

20.STEPPER MOTOR INTERFACING

START: LXI H, 4200

MVI C, 04

NEXT: MOV A, M

OUT CO

LXI D, 1010

LOOP: DCX D

MOV A,E

ORA D

JNZ loop

INX H

DCR C

JNZ NEXT

JMP START

HLT

TABLE: 09 05 06 0A

TABLE: OA 06 05 09

21.KEYBOARD AND DISPLAY

MOV SI,1200H

MOV CX,000FH

MOV AL,10

OUT C2,AL

MOV AL,0CC

OUT C2,AL

MOV AL,90

OUT C2,AL

NEXT: MOV AL,[SI]

OUT CO,AL

DELAY: MOV DX,0FFFFH

LOOP1: DEC DX

JNZ LOOP1

INC SI

LOOP NEXT

JMP START

HLT

22.INTERFACE SWITCHES WITH 8086 THROUGH 8255

MOV AL,90

OUT C2,AL

IN AL,C0

MOV[1250],AL

HLT

23.ADDITION OPERATION USING 8051 MICROCONTROLLER

MOV A,#data

MOV B,#data

ADDC A,B

MOV DPTR,#4500H

MOVX @ DPTR,A

STOP: SJMP STOP

24.SUB OPERATION USING 8051 MICROCONTROLLER

MOV A,#data

MOV B,#data

SUBB A,B

MOV DPTR,#4500H

MOVX @ DPTR,A

STOP: SJMP STOP

25.MUL OPERATION USING 8051 OPERATION

MOV A,#data

MOV B,#data

MUL AB

MOV DPTR,#4500H

MOVX @ DPTR,A

INC DPTR

MOV A,B

MOVX @ DPTR,A

STOP: SJMP STOP

26.DIV OPERATION USING 8051 OPERATION

MOV A,#CF

MOV B,#21

DIV AB

MOV DPTR,#4500H

MOVX @ DPTR,A

INC DPTR

MOV A,B

MOVX @ DPTR,A

STOP: SJMP STOP

27.LOGICAL OPERATION 8051

MOV A,#C3

MOV B,#0F

ORL A,B

MOV DPTR,#4500H

MOVX@DPTR,A

STOP: SJMP STOP

28.MASKING OF BITS USING 8051

MOV A,#4D

MOV B,#OF

ANL A,B

MOV DPTR,#4500H

MOVX @ DPTR,A

STOP: SJMP STOP