1. **8 BIT ADDITION:**

LDA 4500

MOV B, A

LDA 4501

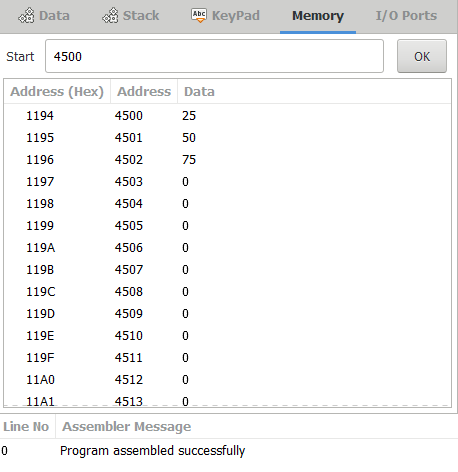
ADD B

STA 4502

RST 1

Hlt

**OUTPUT:**



1. **8 BIT SUBTRACTION:**

LDA 8000

MOV B, A

LDA 8001

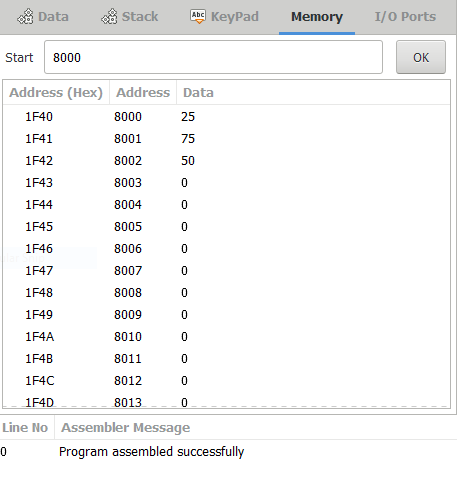
SUB B

STA 8002

RST 1

Hlt

**OUTPUT:**



1. **BIT MULTIPLICATION:**

LDA 4200

MOV E,A

LDA 4202

MOV B,A

LXI H,0000H

MVI D,00H

NEXT: DAD D

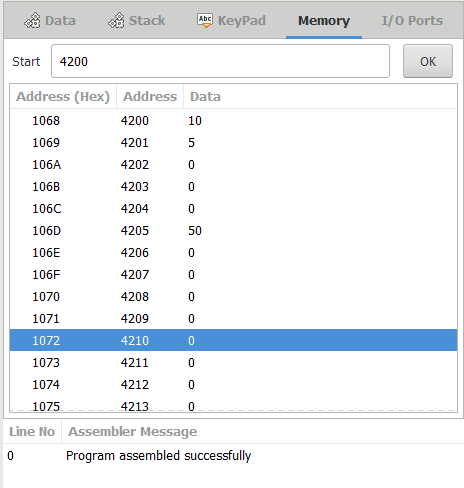
DCR B

JNZ NEXT

SHLD 4205h

hlt

**OUTPUT:**



1. **8 BIT DIVISION:**

LDA 4201

MOV B,A

LDA 4200

MVI C,00H

AGAIN: CMP B

JC STORE

SUB B

INR C

JMP AGAIN

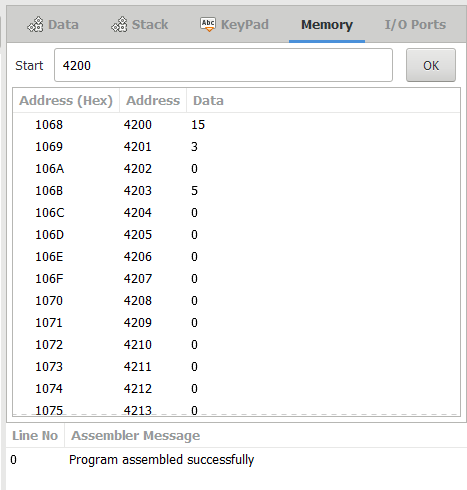
STORE: STA 4202

MOV A,C

STA 4203

Hlt

**OUTPUT:**



**5.16 BIT ADDITION:**

LDA 3050

MOV B,A

LDA 3051

ADD B

STA 3052

LDA 3053

MOV B,A

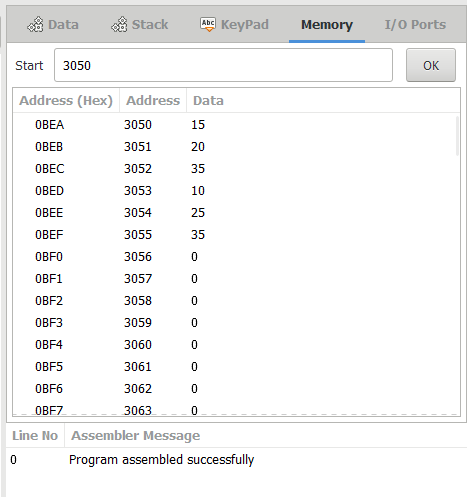
LDA 3054

ADC B

STA 3055

HLT

**OUTPUT:**



**6. 16 BIT SUBTRACTION:**

LDA 3050

MOV B,A

LDA 3051

SUB B

STA 3052

LDA 3053

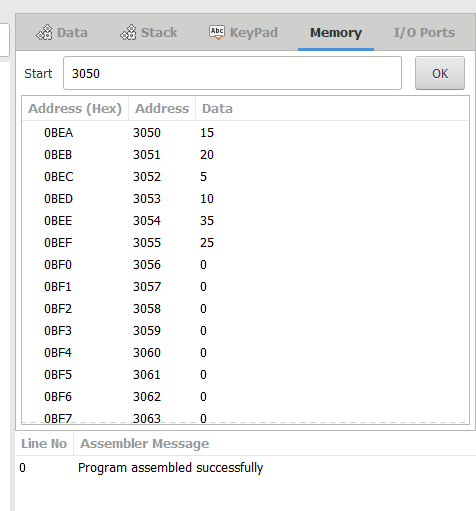
MOV B,A

LDA 3054

SBB B

STA 3055

HLT



**7. 16 BIT MULTIPLICATION:**

LHLD 2050

SPHL

LHLD 2052

XCHG

LXI H,0000H

LXI B,0000H

AGAIN: DAD SP

JNC START

INX B

START: DCX D

MOV A,E

ORA D

JNZ AGAIN

SHLD 2054

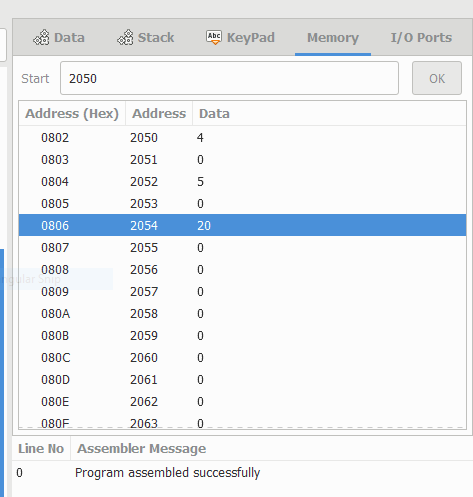
MOV L,C

MOV H,B

SHLD 2055

HLT

**OUTPUT:**



**8. 16 BIT DIVISION:**

LDA 8501

MOV B,A

LDA 8500

MVI C,00

LOOP:CMP B

JC LOOP1

SUB B

INR C

JMP LOOP

STA 8503

DCR C

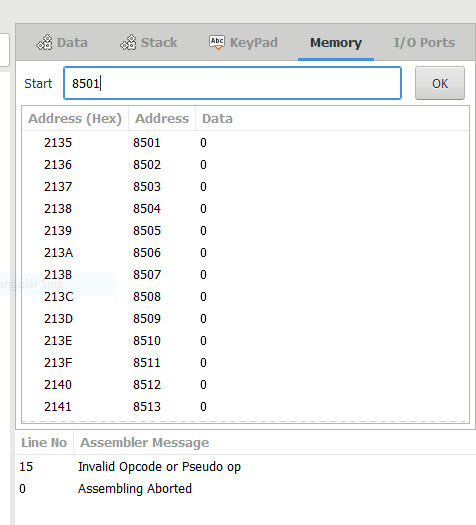
MOV A,C

LOOP1: STA 8502

RST 1

hlt

**OUTPUT:**



**9.FACTORIAL:**

LDA 2001

MOV B,A

MVI C,#01

MVI E,#01

LOOP: MOV D,C

MVI A,00H

LP: ADD E

DCR D

JNZ LP

MOV E,A

INR C

DCR B

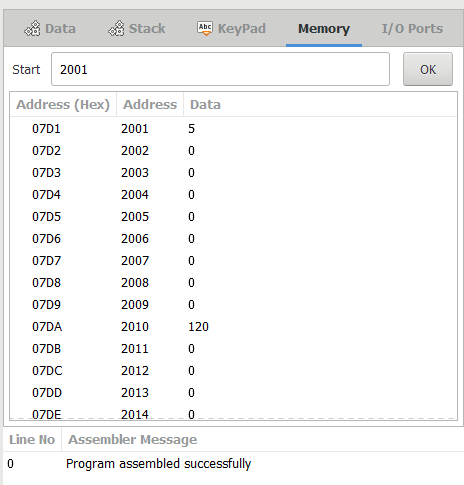
JNZ LOOP

MOV A,E

STA 2010

HLT

**OUTPUT:**



**10.LARGEST NUMBER:**

LXI H,2050

MOV C,M

DCR C

INX H

MOV A,M

LOOP1: INX H

CMP M

JNC LOOP

MOV A,M

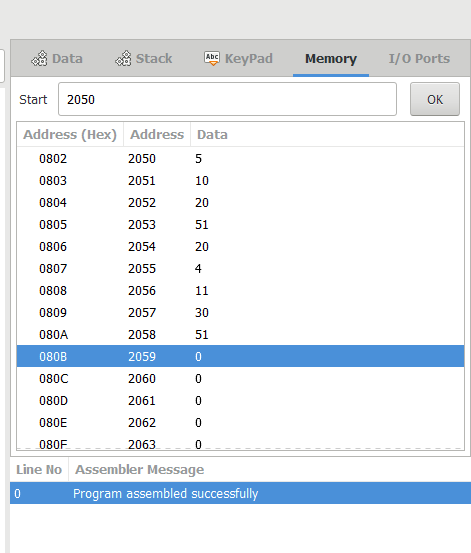
LOOP: DCR C

JNZ LOOP1

STA 2058

HLT

**OUTPUT:**



**11. SMALLEST NUMBER:**

LXI H,2050

MOV C,M

DCR C

INX H

MOV A,M

LOOP1: DCX H

CMP M

JNC LOOP

MOV A,M

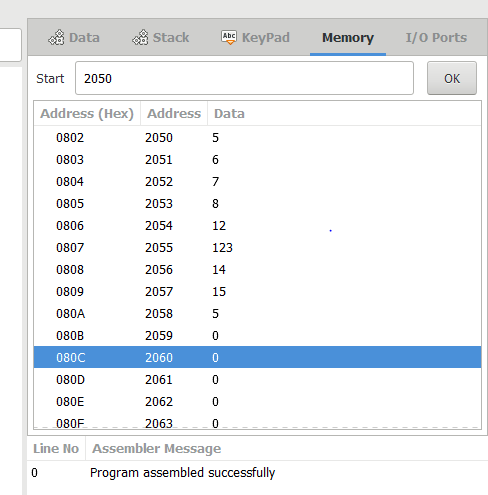
LOOP: INR C

JNZ LOOP1

STA 2058

HLT

**OUTPUT:**



**12.ASCENDING ORDER:**

LOOP: LXI H,3500

MVI D,00

MVI C,05

LOOP1: MOV A,M

INX H

CMP M

JC LOOP2

MOV B,M

MOV M,A

DCX H

MOV M,B

INX H

MVI D,01

LOOP2: DCR C

JNZ LOOP1

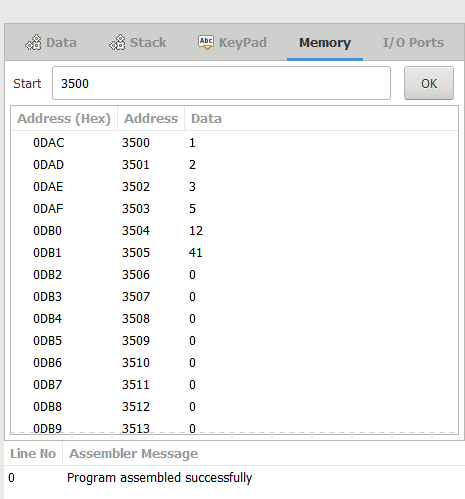
MOV A,D

RRC

JC LOOP

HLT

**OUTPUT:**



**13.DESCENDING ORDER:**

LOOP: LXI H,3500

MVI D,00

MVI C,05

LOOP1: MOV A,M

INX H

CMP M

JNC LOOP2

MOV B,M

MOV M,A

DCX H

MOV M,B

INX H

MVI D,01

LOOP2: DCR C

JNZ LOOP1

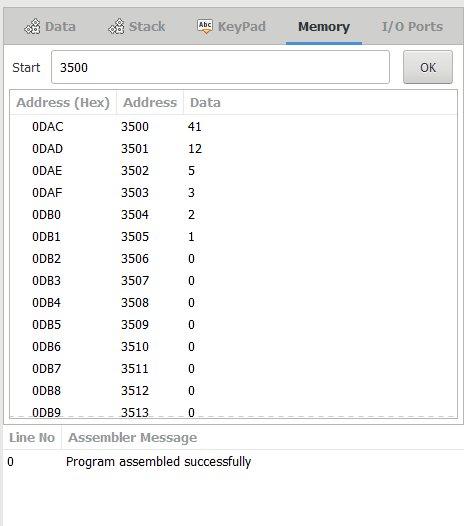
MOV A,D

RRC

JC LOOP

HLT

**OUTPUT:**



**14.ADDITION OF N NUMBERS:**

LXI H,8000

MOV C,M

MVI A,00

MOV B,A

LOOP: ADD C

JNC SKIP

INR B

SKIP: DCR C

JNZ LOOP

LXI H,8007

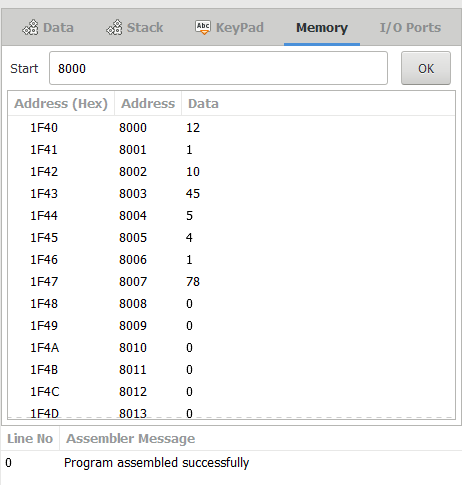
MOV M,A

INX H

MOV M,B

HLT

**OUTPUT:**



**15. SWAPPING OF NUMBERS:**

LDA 2001

MOV B,A

LDA 2002

MOV C,A

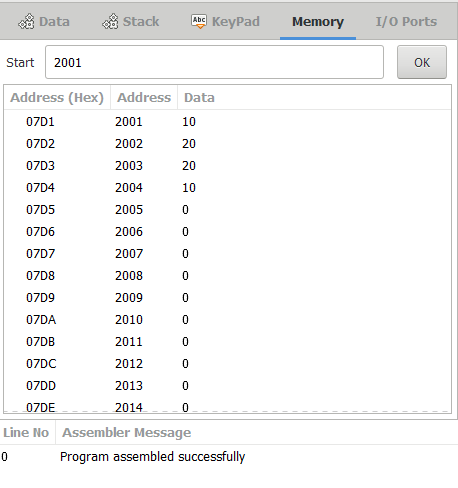
STA 2003

MOV A,B

STA 2004

HLT

**OUTPUT:**



**16. SQUARE OF NUMBERS:**

LXI H,4000

XRA A

MOV B,M

LOOP: ADD M

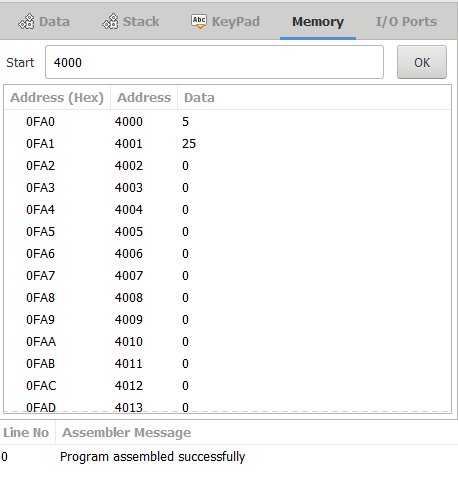
DCR B

JNZ LOOP

STA 4001

HLT

**OUTPUT:**



**18.ONES COMPLEMENT:**

LDA 3000

CMA

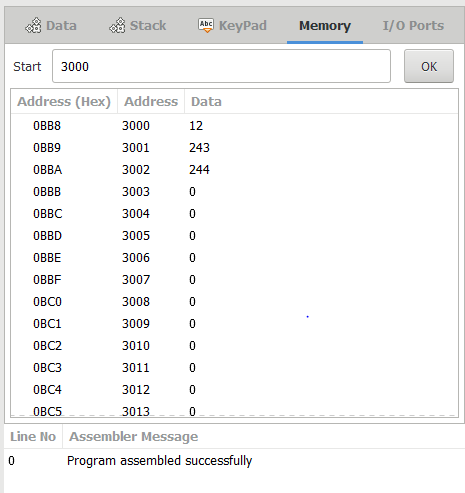
STA 3001

ADI 01

STA 3002

HLT

**OUTPUT:**



**19.ROTATE LEFT:**

MVI A,02

RLC

RLC

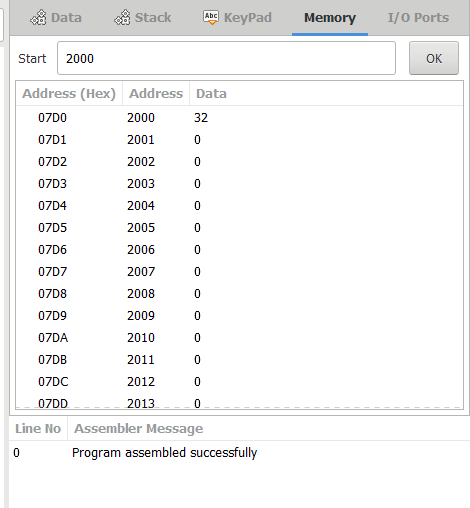
RLC

RLC

STA 2000

HLT

**OUTPUT:**



**20.ROTATE RIGHT:**

MVI A,03

RRC

RRC

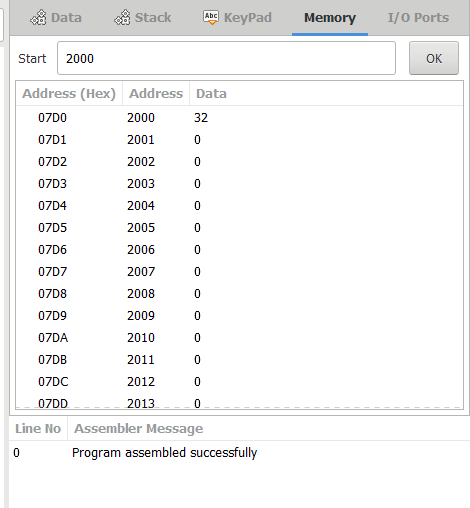
RRC

RRC

STA 2000

HLT

**OUTPUT:**



**LOGICAL OPERATIONS:**

**AND OPERATION:**

MVI A,06

MVI B,04

ANA B

STA 2500

HLT

**OR OPERATION:**

MVI A,07

MVI B,06

ORA B

STA 2000

HLT

**XOR OPERATION:**

MVI A,03

MVI B,04

XRA B

STA 2000

HLT