ASIAN SCHOOL OF MANAGEMENT AND TECHNOLOGY

[Affiliated to Tribhuvan University]

ASSIGNMENT: KIT LAB REPORT OF MICROPROCESSOR

Submitted By

Name: Manalika Shrestha

Roll No: 13

Program: B.Sc. CSIT

Submission Date: / /2080

Registration No:

Submitted To:

Er. Manoj Giri

_____

Kathmandu, Nepal

Table of contents

1.Write a program to add 56H and 37H and store the sum in memory address 8050H.

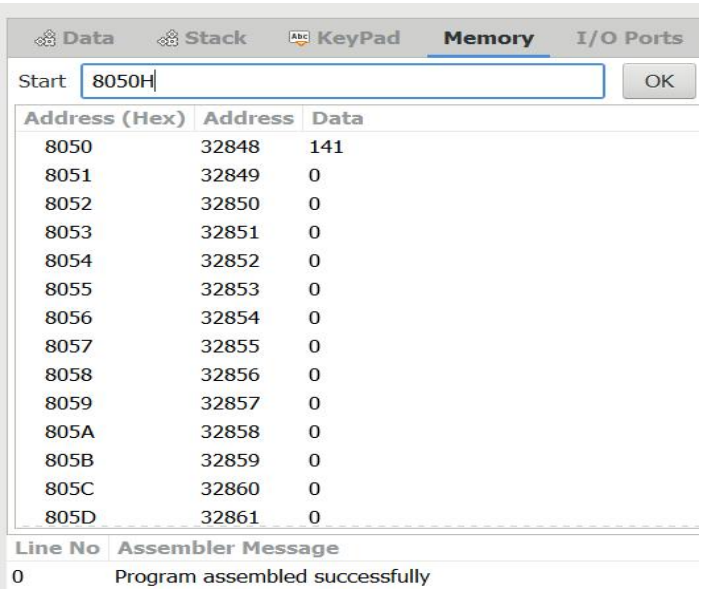
MVI A, 56H

ADI 37H

STA 8050H

HLT

OUTPUT:



2.Write a program to add two number store in 8050H and 8051H and store the sum in 8052H.

LDA 8050H

MOV H, A

LDA 8051H

ADD H

MOV L, A

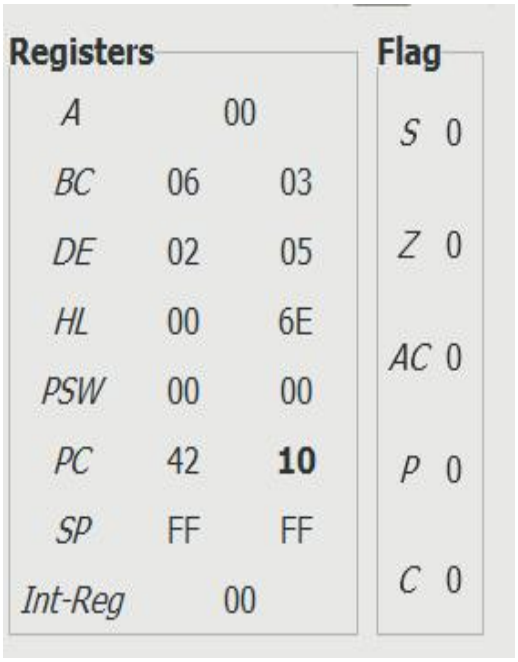
MVI A, 00

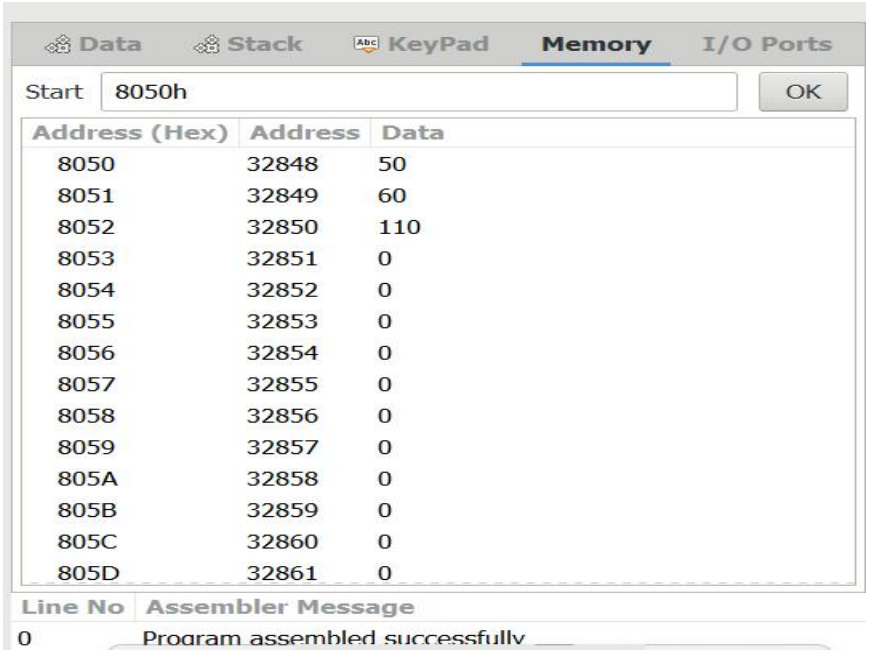
MOV H, A

SHLD 8052H

HLT

OUTPUT:





3.Write a program to subtract 37H from 52H and store the difference in memory address 8050.

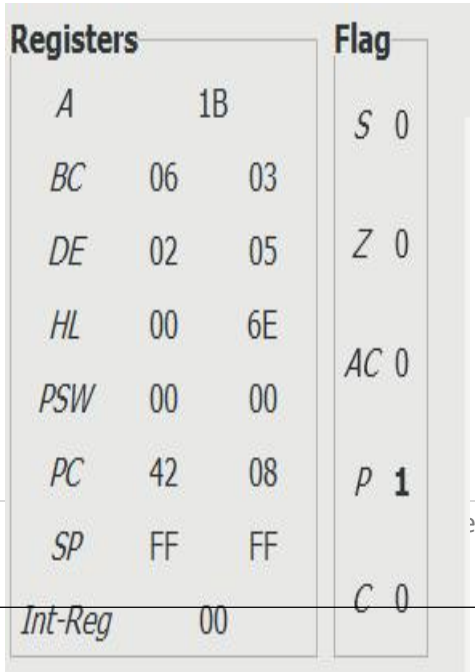
MVI A, 52H

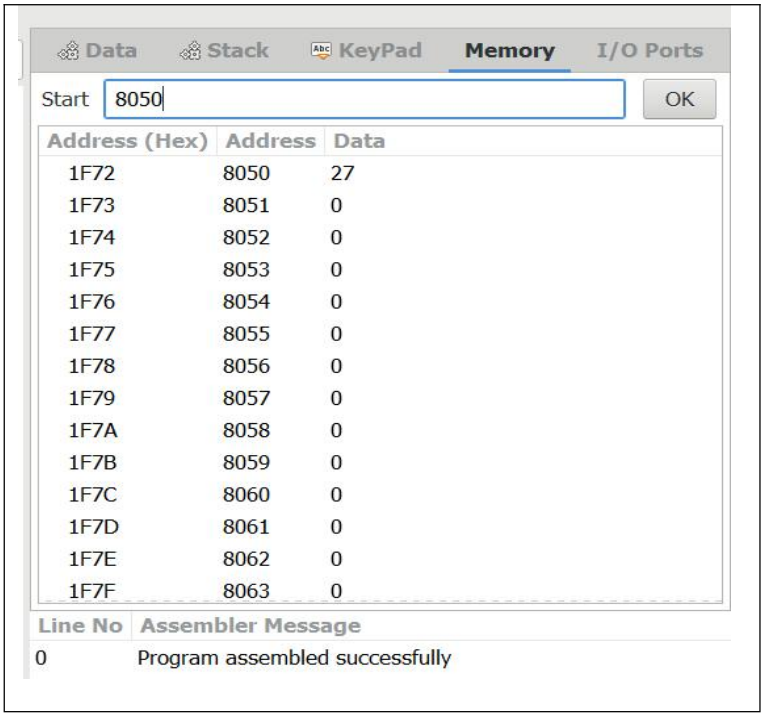
SUI 37H

STA 8050

HLT

OUTPUT:





4.Write a program to find the difference of number stored in location 9050 and 9051 and stored the difference in 9052.

LDA 9050

M0V B, A

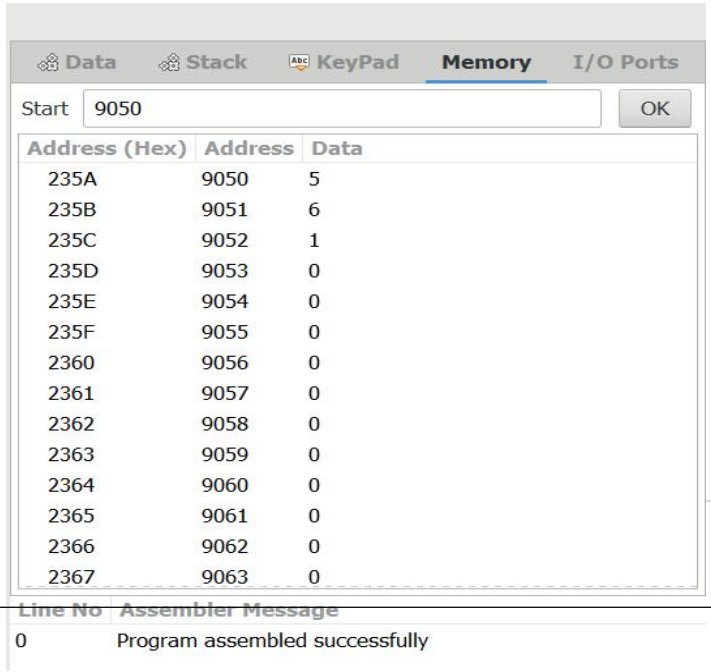
LDA 9051

SUB B

STA 9052

HLT

OUTPUT:



Registers

A	01	
BC	05	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	**0C**
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	0
AC	0
P	0
C	0

5.Write a program to find greater number between two number stored in location 8050H and 8051H and stored the bigger number in memory address 9050H.

LDA 8050H

MOV B, A

LDA 8051

CMP B

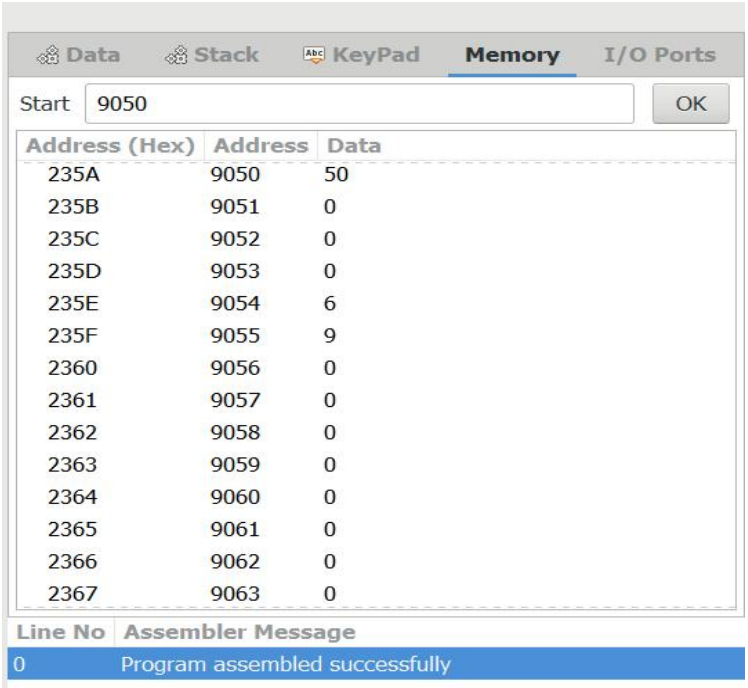
JNC GREATER

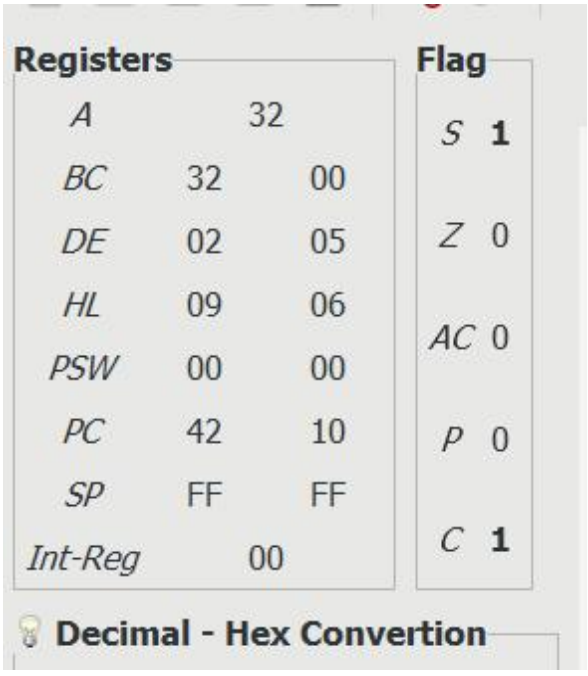
MOV A, B

GREATER: STA 9050

HLD

OUTPUT:





6. Write a program to find a given number is positive or negative and if positive store in location 8050 if negative store in 8051.

LDA 9050H

MOV B, A

RAL

JC NEGATIVE

MVI A,00H

RAR

MOV A, B

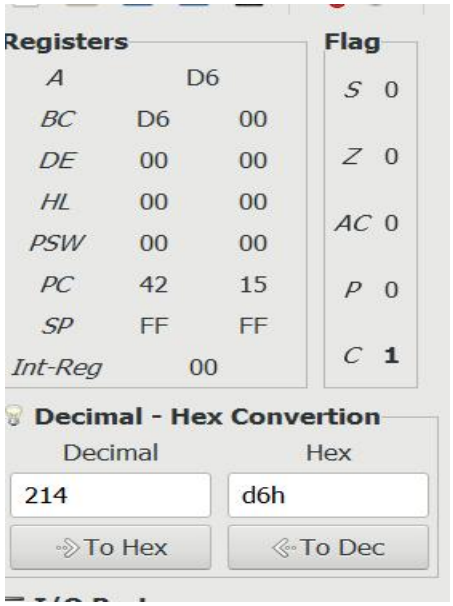
STA 8050H

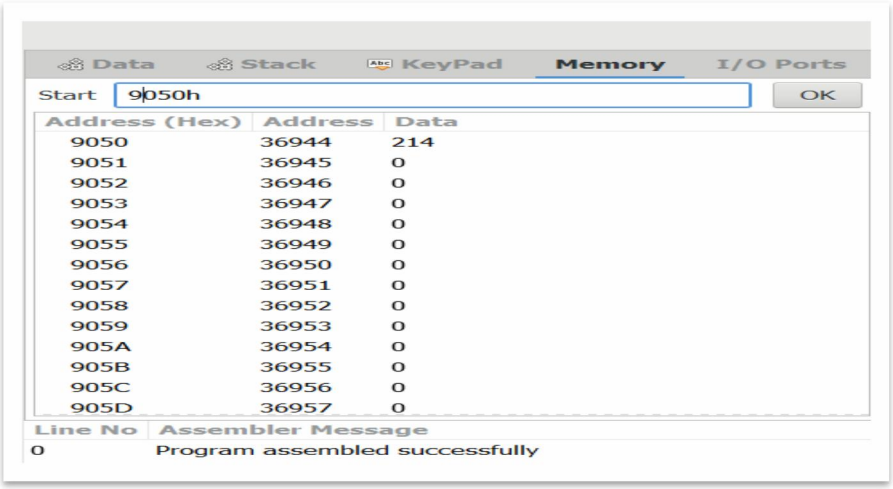
NEGATIVE: MOV A, B

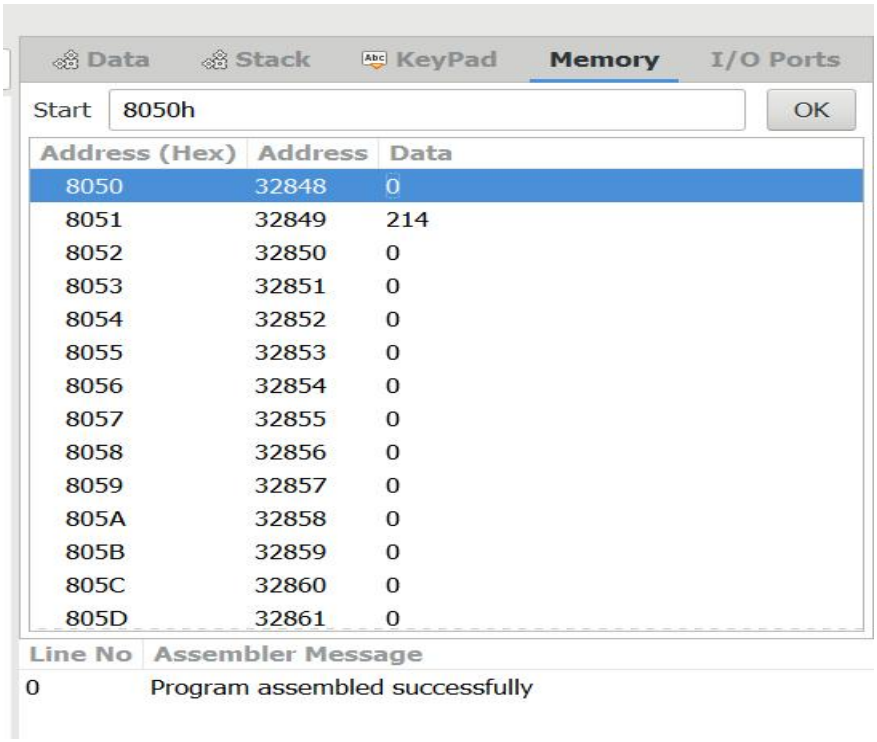
STA 8051H

HLT

OUTPUT:




Data
Stack
KeyPad
Memory
I/O Ports
Start
9050h
OK
Address (Hex)
Address
Data
9050 36944 214
9051 36945 0
9052 36946 0
9053 36947 0
9054 36948 0
9055 36949 0
9056 36950 0
9057 36951 0
9058 36952 0
9059 36953 0
905A 36954 0
905B 36955 0
905C 36956 0
905D 36957 0
Line No
Assembler Message
0
Program assembled successfully


Data
Stack
KeyPad
Memory
I/O Ports
Start
8050h
OK
Address (Hex)
Address
Data
8050 32848 0
8051 32849 214
8052 32850 0
8053 32851 0
8054 32852 0
8055 32853 0
8056 32854 0
8057 32855 0
8058 32856 0
8059 32857 0
805A 32858 0
805B 32859 0
805C 32860 0
805D 32861 0
Line No
Assembler Message
0
Program assembled successfully

7.Write a program to find a given number is even or odd and if even stored in location 9050H and if odd stored in 9051H.

LDA 8050H

ANI 01H

JZ EVEN

MVI A, 01H

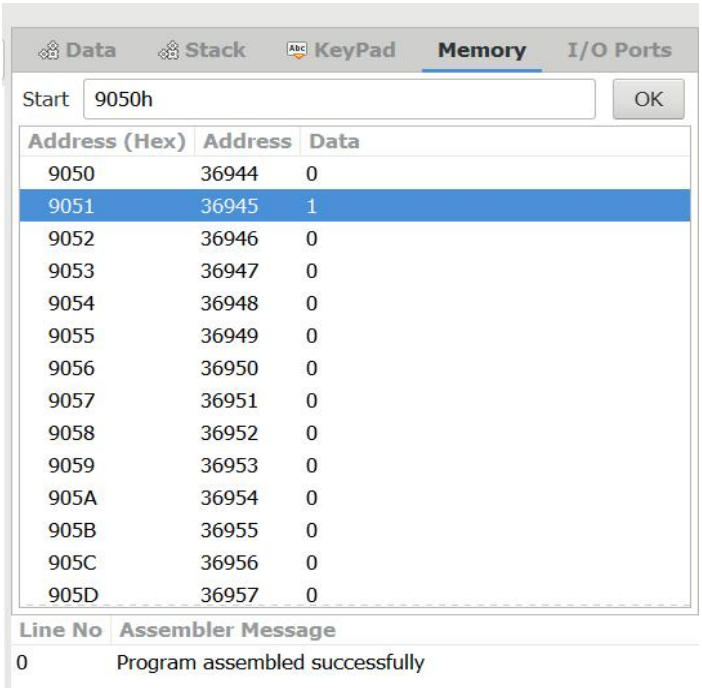
STA 9050H

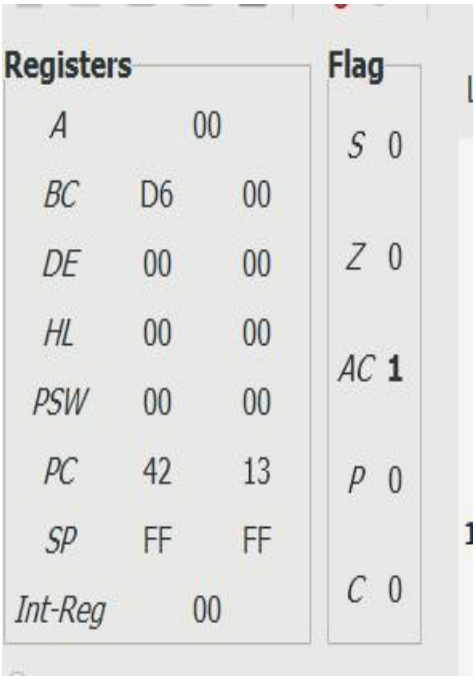
EVEN: MVI A, 00H

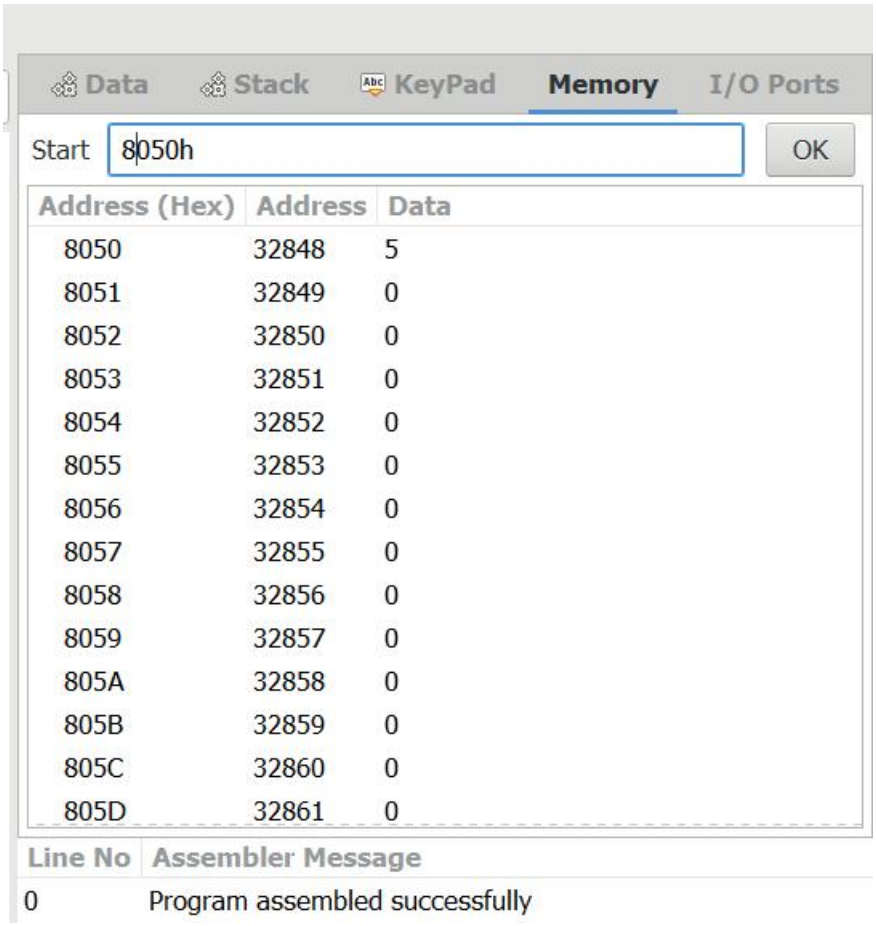
STA 9050H

HLT

OUTPUT:



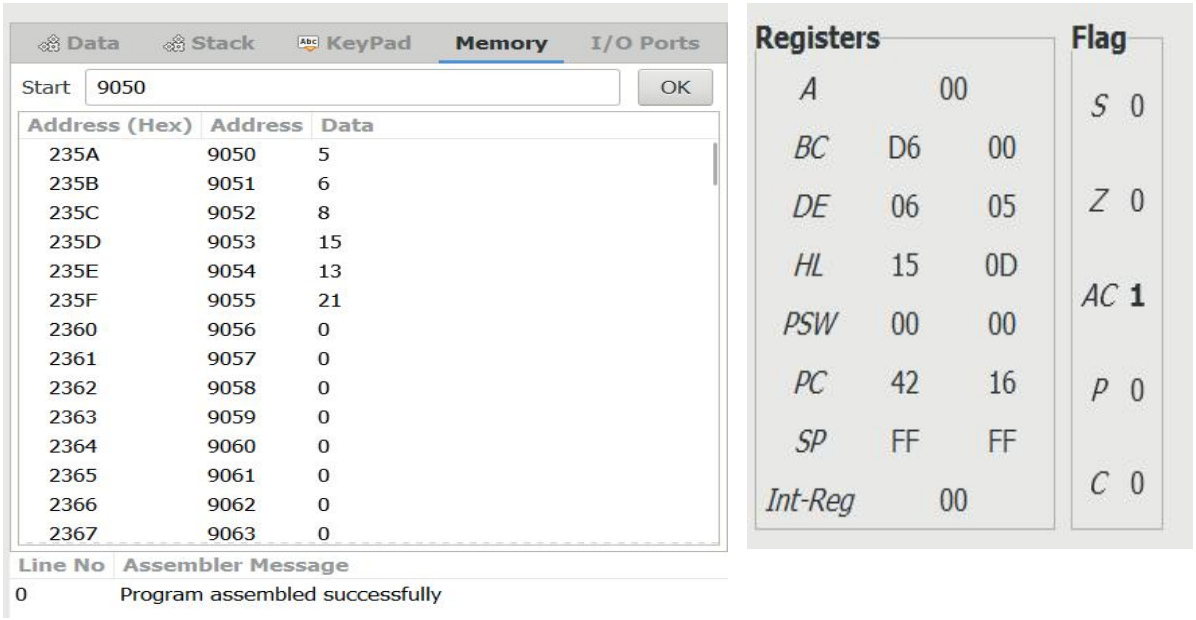



Data
Stack
KeyPad
Memory
I/O Ports
Start
8050h
OK
Address (Hex) Address Data
8050 32848 5
8051 32849 0
8052 32850 0
8053 32851 0
8054 32852 0
8055 32853 0
8056 32854 0
8057 32855 0
8058 32856 0
8059 32857 0
805A 32858 0
805B 32859 0
805C 32860 0
805D 32861 0
Line No Assembler Message
0 Program assembled successfully

8.Write a program to add two 16 bits numbers stored in location 9050,9051 and 9052,9053 and then stored the sum in 9054 and 9055.

```
LHLD 9050

XCHG

LHLD 9052

DAD D

MVI C,00H

JNC LOOP

INR C

LOOP: SHLD 9054

MOV A, C

STA 9056

HLT
```

OUTPUT:



Data | Stack | KeyPad | Memory | I/O Ports

Start 9050 OK

Address (Hex)	Address	Data
235A	9050	5
235B	9051	6
235C	9052	8
235D	9053	15
235E	9054	13
235F	9055	21
2360	9056	0
2361	9057	0
2362	9058	0
2363	9059	0
2364	9060	0
2365	9061	0
2366	9062	0
2367	9063	0

Line No	Assembler Message
0	Program assembled successfully

Registers

A	00	
BC	D6	00
DE	06	05
HL	15	0D
PSW	00	00
PC	42	16
SP	FF	FF
Int-Reg	00	

Flag

S 0

Z 0

AC 1

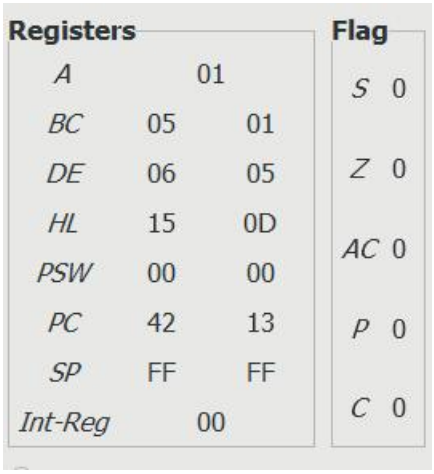
P 0

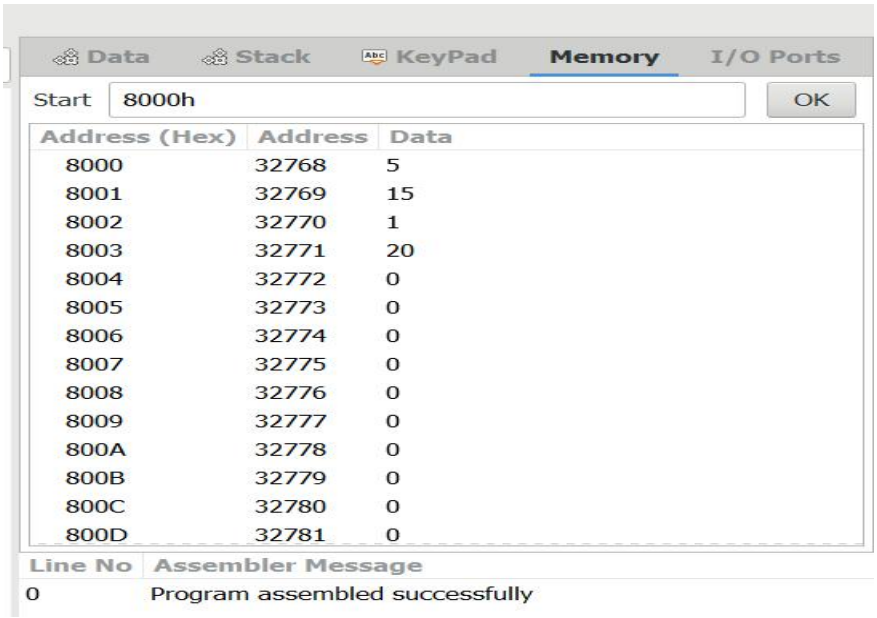
C 0

9.Write a program to add two number stored in location 8000 and 8001 store the sum of addition in 8003 and carry in 8002.

```
MVI C,00
LDA 8000H
MOV B, A
LDA 8001H
ADD B
STA 8003H
INR C
MOV A, C
STA 8002H
HLT
```

OUTPUT:





10. WAP to multiply two number stored in location 8002 and 8003 and stored the product in 8084.

LDA 8000

MOV D, A

LXI H, 8003

MOV C, M

BACK: DCR C

JZ SKIP

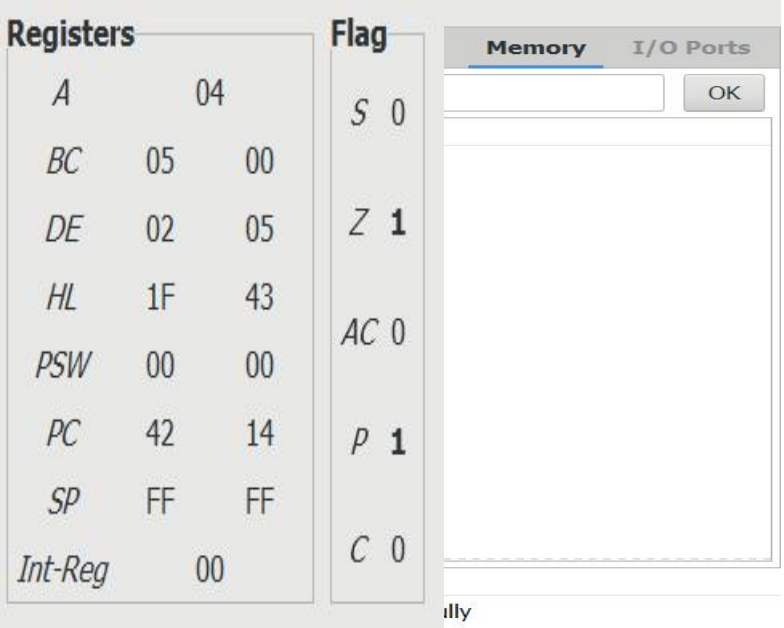
ADD D

JMP BACK

SKIP: STA 8004

HLT

OUTPUT:



11.WAP to find quotient and remainder and store quotient at location 8050 and remainder at location 8051.

LXI H,8000

MOV B, M

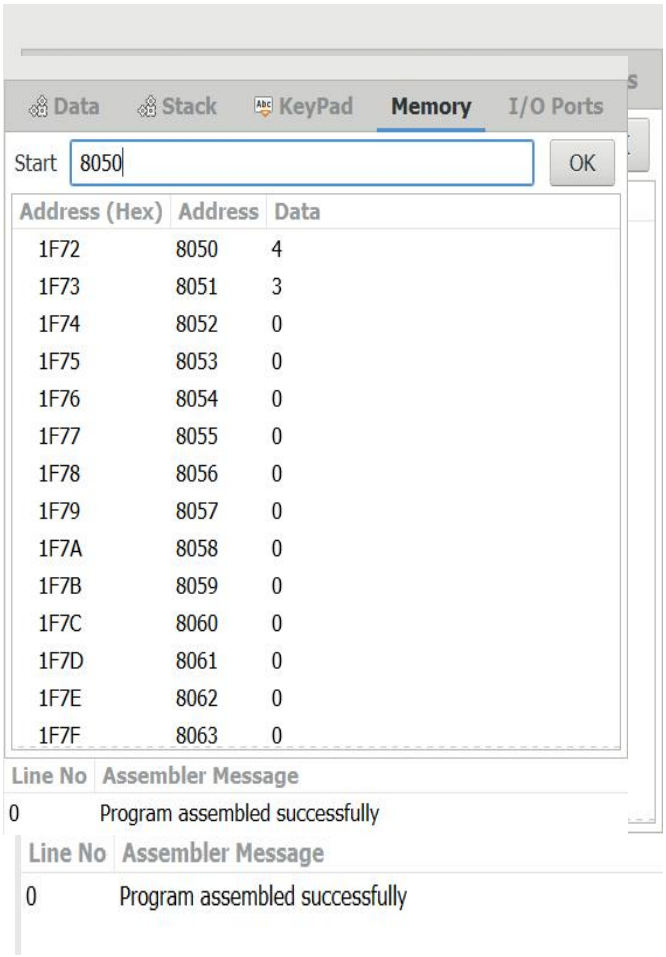
MVI C,00

INX H

```
MOV A, M

NEXT: CMP B

JC LOOP

SUB B

INR C

JMP NEXT

LOOP: STA 8081

MOV A, C

STA 8050

HLT
```

OUTPUT:

Registers			Flag	
A	04		S	1
BC	08	04	Z	0
DE	02	05	AC	0
HL	1F	41	P	0
PSW	00	00	C	1
PC	42	19		
SP	FF	FF		
Int-Reg	00			



Line No	Assembler Message
0	Program assembled successfully

12. Find compliment of a number store in the location 9051and store its compliment in location 9052.

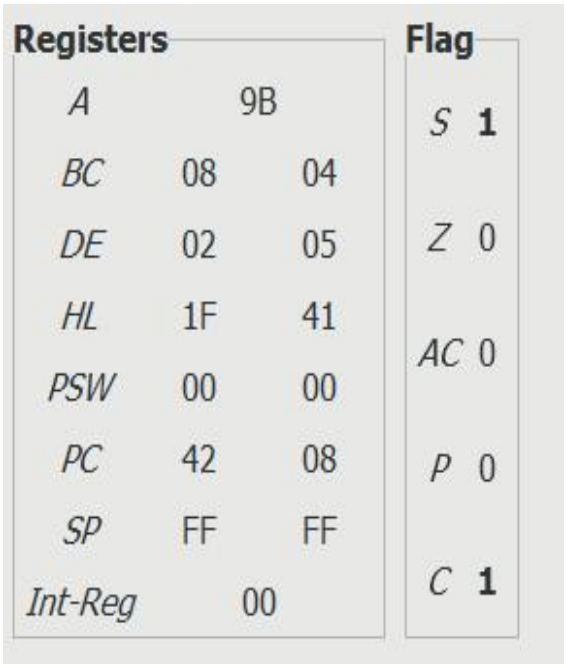
LDA 9051

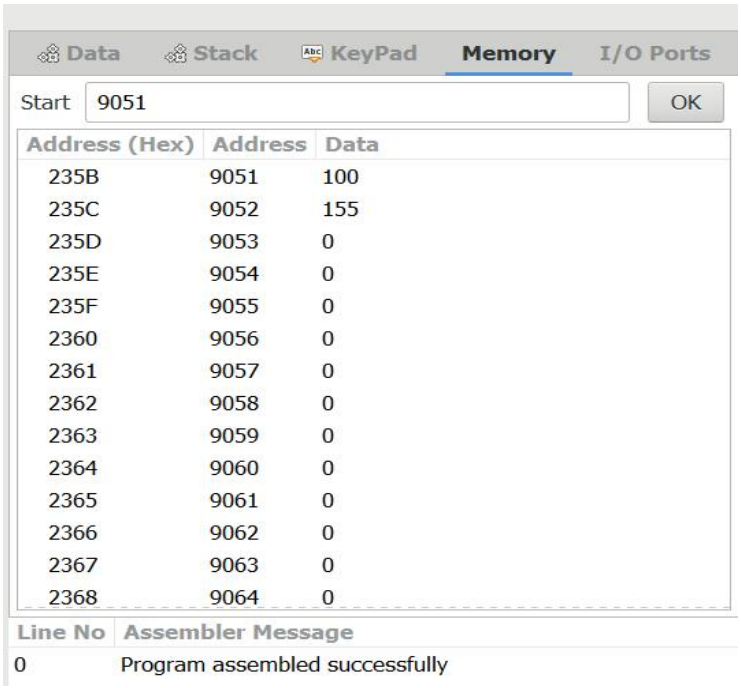
CMA

STA 9052

HLT

OUTPUT:


Registers
A 9B
BC 08 04
DE 02 05
HL 1F 41
PSW 00 00
PC 42 08
SP FF FF
Int-Reg 00
Flag
S 1
Z 0
AC 0
P 0
C 1


Data
Stack
KeyPad
Memory
I/O Ports
Start 9051
OK
Address (Hex) Address Data
235B 9051 100
235C 9052 155
235D 9053 0
235E 9054 0
235F 9055 0
2360 9056 0
2361 9057 0
2362 9058 0
2363 9059 0
2364 9060 0
2365 9061 0
2366 9062 0
2367 9063 0
2368 9064 0
Line No Assembler Message
0 Program assembled successfully

13. Find compliment 2's of a number stored in location 8050 and store its compliment at location.

LDA 8050

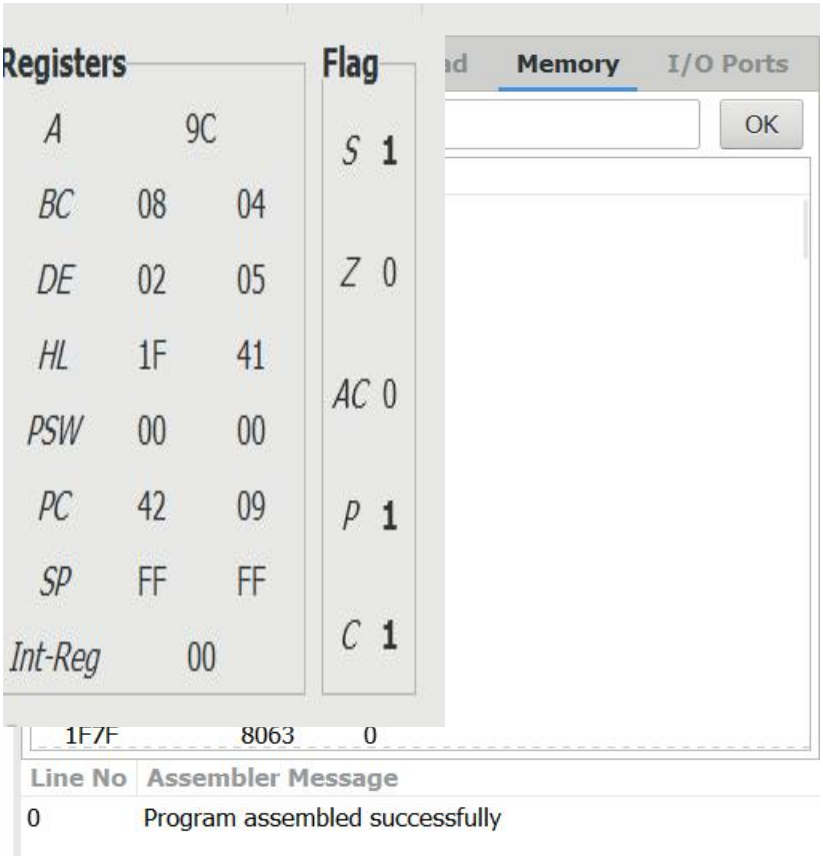
CMA

INR A

STA 8051

HLT

OUTPUT:



14. Find the greatest number out of 10 number stored from location 9010 and number stored the greatest number at location 9060.

LXI H,9010

MOV C, M

INX H

DCR C

MOV A, M

Skip: INX H

CMP M

JNC LOOP

MOV A, M

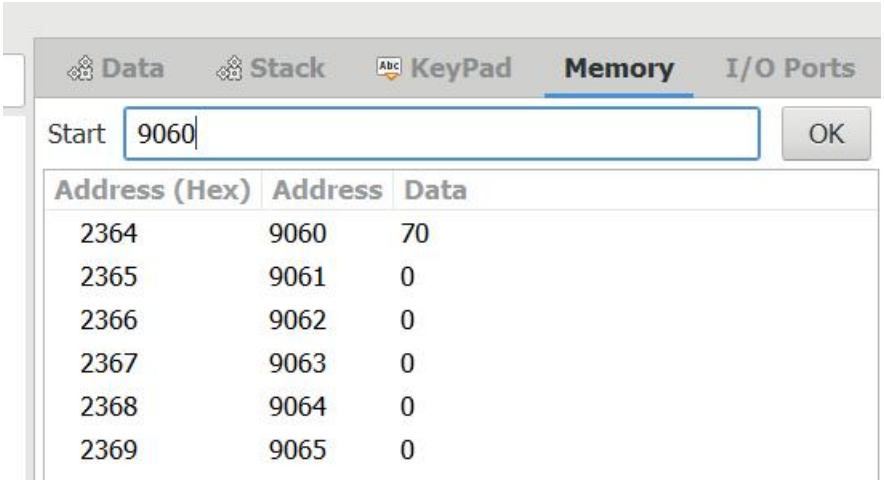
LOOP: DCR C

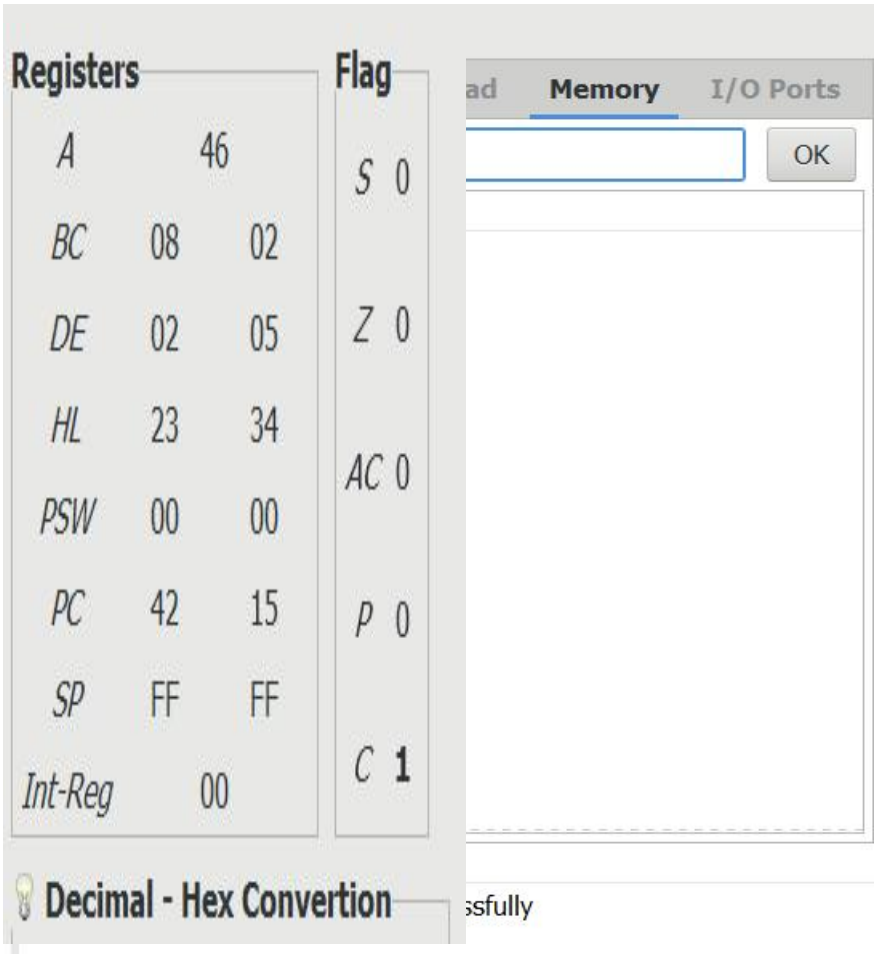
JNC Skip

STA 9060

HLT

OUTPUT:




Registers
Flag
A 46
BC 08 02
DE 02 05
HL 23 34
PSW 00 00
PC 42 15
SP FF FF
Int-Reg 00
S 0
Z 0
AC 0
P 0
C 1
Memory
I/O Ports
OK
Decimal - Hex Convertion

15.Find the smallest number out of 10 number stored from location 9010 and store the smallest at location 9060.

LXI H, 9010

MOV C, M

INX H

MOV B, M

DCR C

LOOP: INX H

MOV A, M

CMP B

JNC Skip

MOV B, A

Skip: DCR C

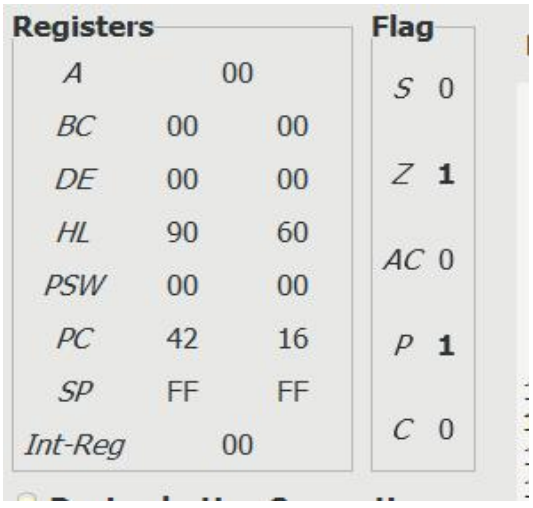
JNZ LOOP

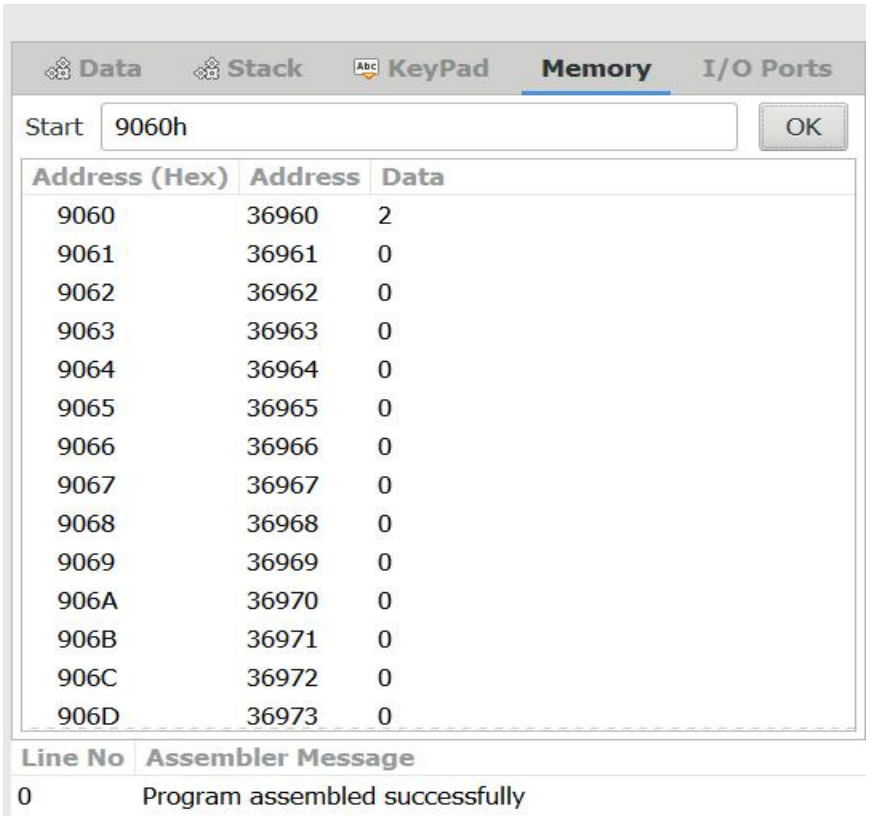
LXI H,9060H

MOV M, B

HLT

OUTPUT:




Data
Stack
KeyPad
Memory
I/O Ports
Start
9060h
OK
Address (Hex) Address Data
9060 36960 2
9061 36961 0
9062 36962 0
9063 36963 0
9064 36964 0
9065 36965 0
9066 36966 0
9067 36967 0
9068 36968 0
9069 36969 0
906A 36970 0
906B 36971 0
906C 36972 0
906D 36973 0
Line No Assembler Message
0 Program assembled successfully

16. WAP to copy 10 data from memory location 8050 to new memory location 9050.

LXI H,8050H

LXI D,9050H

MVI B,10H

LOOP:MOV A, M

STAX D

INX H

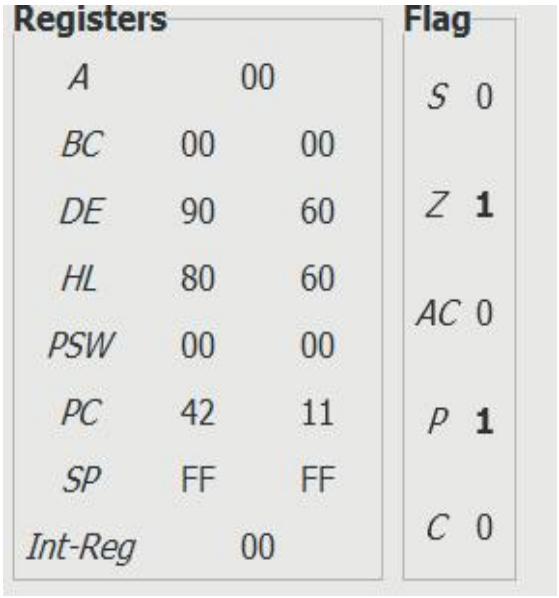
INX D

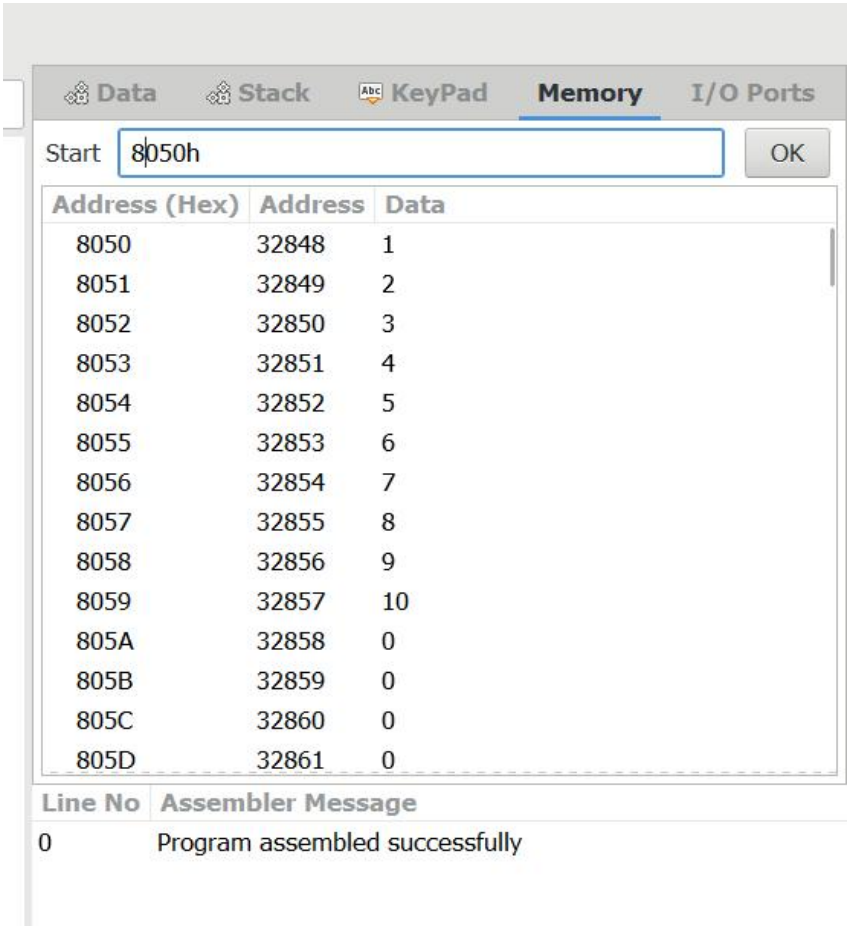
DCR B

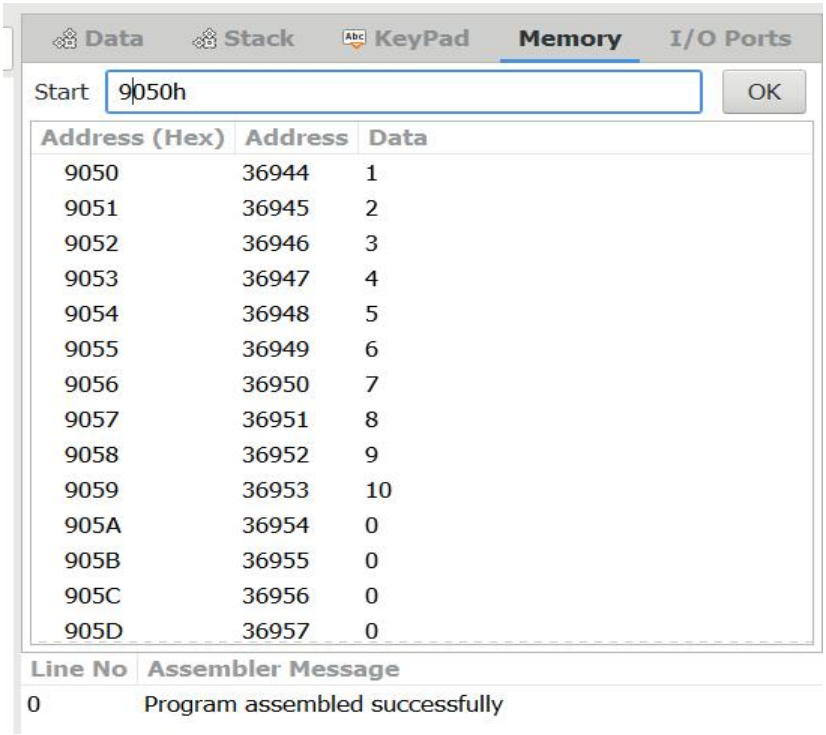
JNZ LOOP

HLT

OUTPUT:



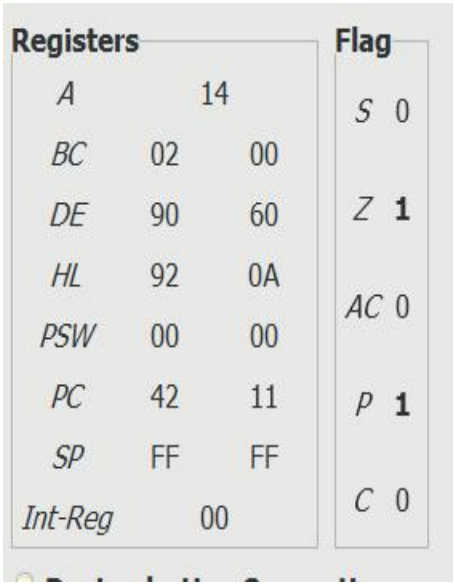

Data
Stack
KeyPad
Memory
I/O Ports
Start
8050h
OK
Address (Hex) Address Data
8050 32848 1
8051 32849 2
8052 32850 3
8053 32851 4
8054 32852 5
8055 32853 6
8056 32854 7
8057 32855 8
8058 32856 9
8059 32857 10
805A 32858 0
805B 32859 0
805C 32860 0
805D 32861 0
Line No Assembler Message
0 Program assembled successfully


Data
Stack
KeyPad
Memory
I/O Ports
Start
9050h
OK
Address (Hex) Address Data
9050 36944 1
9051 36945 2
9052 36946 3
9053 36947 4
9054 36948 5
9055 36949 6
9056 36950 7
9057 36951 8
9058 36952 9
9059 36953 10
905A 36954 0
905B 36955 0
905C 36956 0
905D 36957 0
Line No Assembler Message
0 Program assembled successfully

17. WAP to generate 10 even numbers.

```
MVI A,00H

MVI B,02H

MVI C, 0AH

LXI H,9200H

LOOP: ADDB

MOV M, A

INX H

DCR C

JNZ LOOP

HLT
```

OUTPUT:

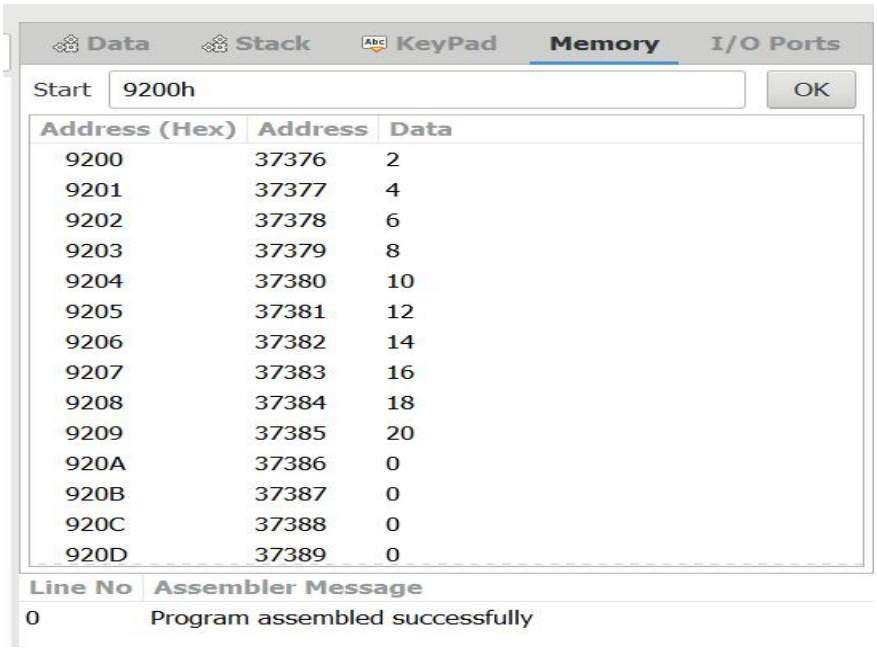


Registers

A	14	
BC	02	00
DE	90	60
HL	92	0A
PSW	00	00
PC	42	11
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 0



Data | Stack | KeyPad | **Memory** | I/O Ports

Start 9200h OK

Address (Hex)	Address	Data
9200	37376	2
9201	37377	4
9202	37378	6
9203	37379	8
9204	37380	10
9205	37381	12
9206	37382	14
9207	37383	16
9208	37384	18
9209	37385	20
920A	37386	0
920B	37387	0
920C	37388	0
920D	37389	0

Line No	Assembler Message
0	Program assembled successfully

18. WAP to generate 10 odd number.

MVI A,01H

MVI B,02H

MVI C,0AH

LXI H, 9050H

LOOP: ADD B

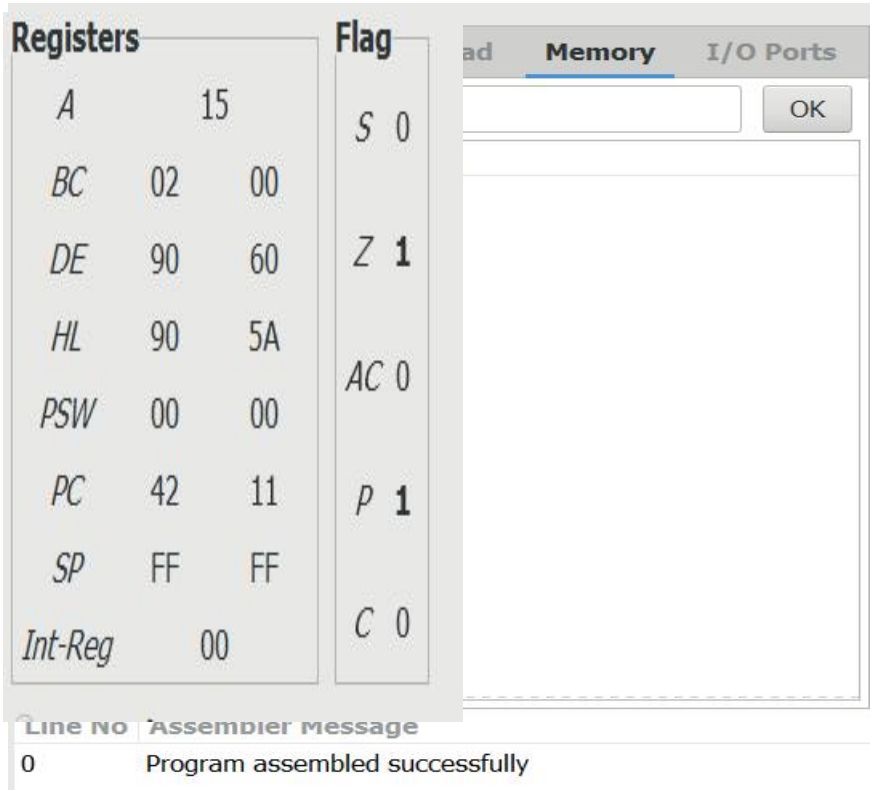
MOV M, A

INX H

DCR C

HLT

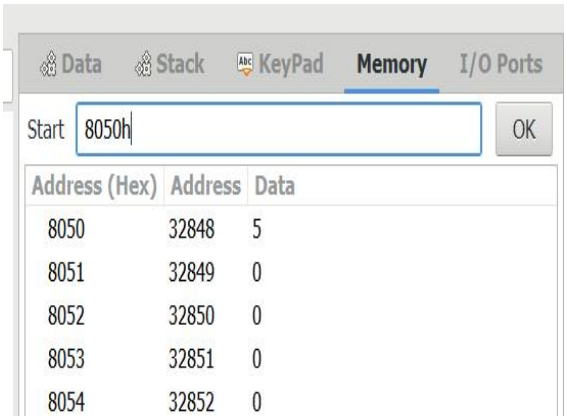
OUTPUT:

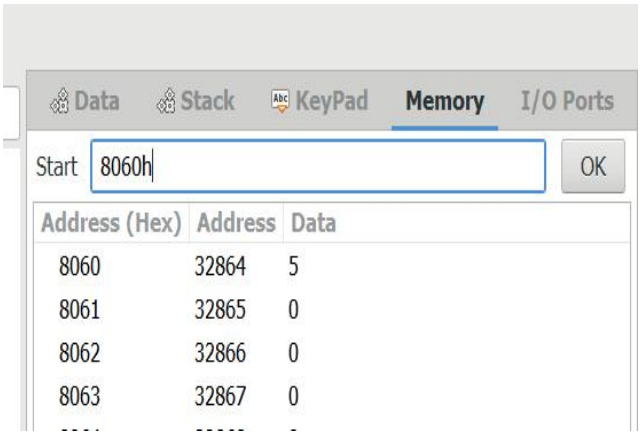


19.WAP to count even number or odd number out of 10 number and store even number in location 8050H and odd number in location 8060H.

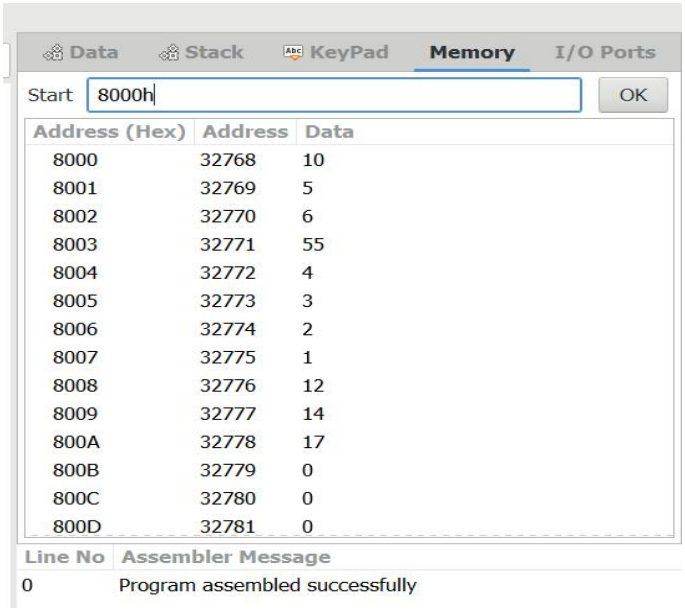
```
LXI H,8000H
MOV C, M
MVI B,00H
MVI D,00H
LOOP2: INX H
MOV A, M
RAR
JC LOOP1
INR B
JMP LOOP3
LOOP1: INR D
LOOP3: DCR C
JNZ LOOP2
MOV A, B
STA 8050h
MOV A, D
STA 8060h
HLT
```

OUTPUT:


Data
Stack
KeyPad
Memory
I/O Ports
Start
8050h
OK
Address (Hex)
Address
Data
8050 32848 5
8051 32849 0
8052 32850 0
8053 32851 0
8054 32852 0


Data
Stack
KeyPad
Memory
I/O Ports
Start
8060h
OK
Address (Hex)
Address
Data
8060 32864 5
8061 32865 0
8062 32866 0
8063 32867 0

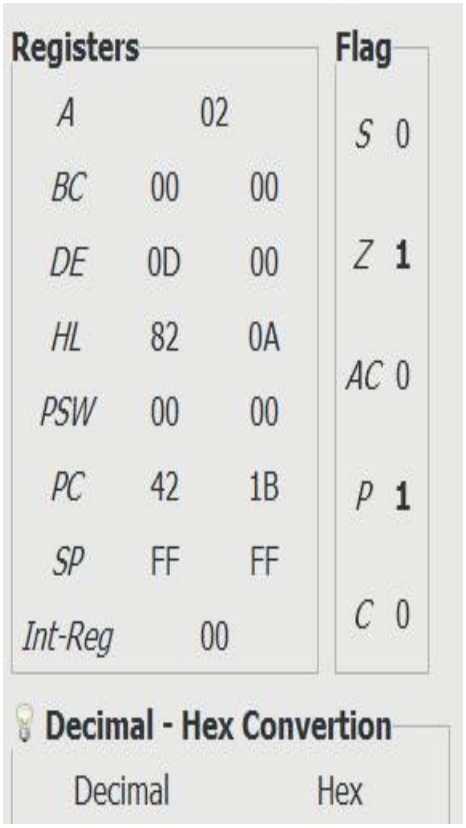

Registers
A 05
BC 05 00
DE 05 00
HL 80 0A
PSW 00 00
PC 42 20
SP FF FF
Int-Reg 00
Flag
S 0
Z 1
AC 0
P 1
C 1

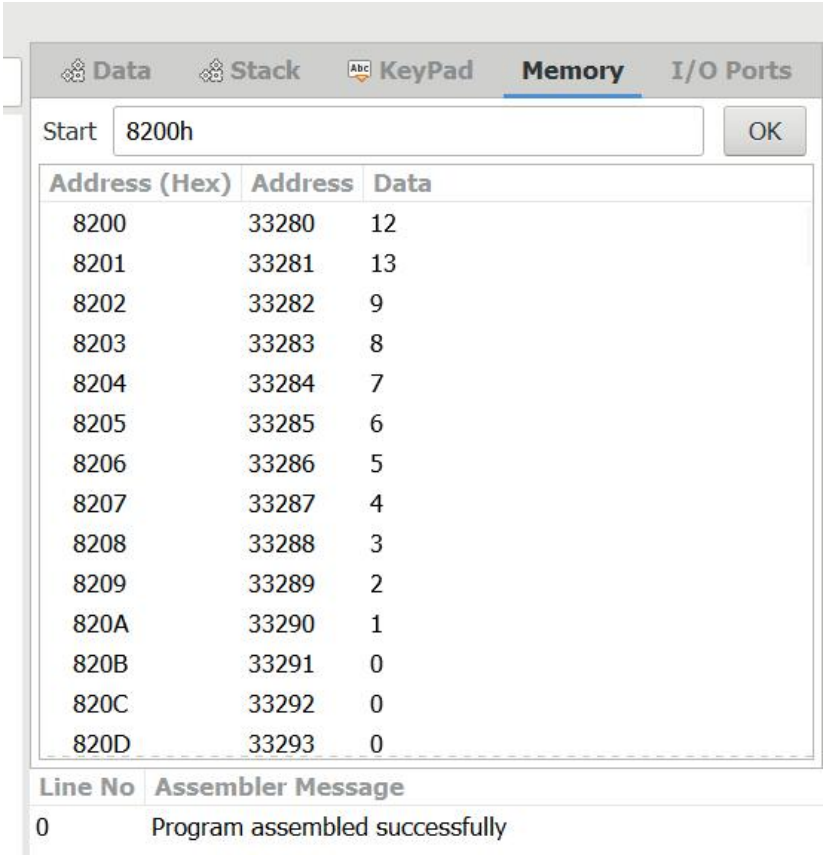

Data
Stack
KeyPad
Memory
I/O Ports
Start
8000h
OK
Address (Hex)
Address
Data
8000 32768 10
8001 32769 5
8002 32770 6
8003 32771 55
8004 32772 4
8005 32773 3
8006 32774 2
8007 32775 1
8008 32776 12
8009 32777 14
800A 32778 17
800B 32779 0
800C 32780 0
800D 32781 0
Line No
Assembler Message
0
Program assembled successfully

20.WAP to sort given 10 numbers from memory location 8200H in ascending order.

```
MVI B, 09
START: LXI H, 8200H
MVI C,0AH
BACK: MOV A, M
INX H
CMP M
JNC SKIP
MOV D, M
MOV M, A
DCX H
MOV M, D
INX H
SKIP: DCR C
JNZ BACK
DCR B
JNZ START
HLT
```

OUTPUT:


Registers
A 02
BC 00 00
DE 0D 00
HL 82 0A
PSW 00 00
PC 42 1B
SP FF FF
Int-Reg 00
Flag
S 0
Z 1
AC 0
P 1
C 0
Decimal - Hex Convertion
Decimal
Hex


Data
Stack
KeyPad
Memory
I/O Ports
Start 8200h
OK
Address (Hex) Address Data
8200 33280 12
8201 33281 13
8202 33282 9
8203 33283 8
8204 33284 7
8205 33285 6
8206 33286 5
8207 33287 4
8208 33288 3
8209 33289 2
820A 33290 1
820B 33291 0
820C 33292 0
820D 33293 0
Line No Assembler Message
0 Program assembled successfully