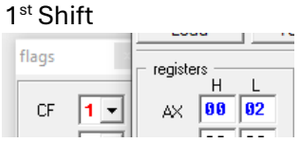
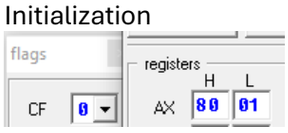
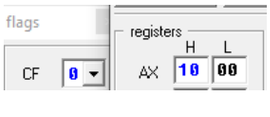
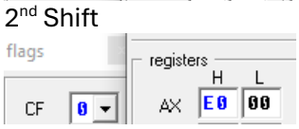
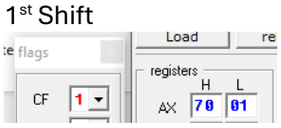
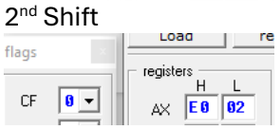
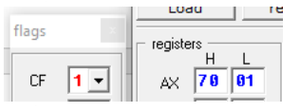
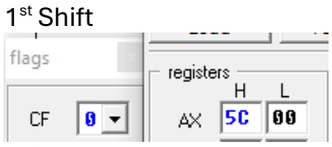
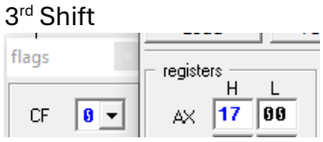
Dave Jhared G. Paduada
BSCpE III - GF

April 5, 2025

SHIFT and ROTATE		
Code	Instruction	Screenshot
MOV AX, 8001H SHL AX,1 SAL AX,1	SHL/SAL	AX and CF(Carry Flag) per shift Initialization  1st Shift  2nd Shift  Shifting in Binary C 0 1000 0000 0000 0001 C 1 0000 0000 0000 0010 C 0 0000 0000 0000 0100
MOV AX,8001H SHR AX,3	SHL	AX and CF(Carry Flag) per shift Initialization  1st Shift  2nd Shift  3rd Shift  Shifting in Binary C 1000 0000 0000 0001 0 C 0100 0000 0000 0000 1 C 0010 0000 0000 0000 0 C 0001 0000 0000 0000 0

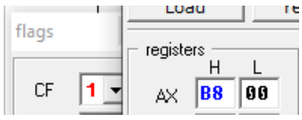
STC MOV AX,8001H SAR AX,3	SAR	AX and CF(Carry Flag) per shift Initialization:  1st Shift  2nd Shift  3rd Shift  Shifting in Binary C 1000 0000 0000 0001 1 C 1100 0000 0000 0000 1 C 1110 0000 0000 0000 0 C 1111 0000 0000 0000 0
MOV AX,0B800H ROL AX,3	ROL	AX and CF(Carry Flag) per rotation Initialization:  1st Shift  2nd Shift  3rd Shift  Rotation in Binary C 0 1011 1000 0000 0000 C 1 0111 0000 0000 0001 C 0 1110 0000 0000 0010 C 1 1100 0000 0000 0101

STC MOV AX, 0B800H RCL AX,3	RCL	AX and CF(Carry Flag) per rotation Initialization:  1st Shift  2nd Shift  3rd Shift  Rotation in Binary C 1 1011 1000 0000 0000 C 1 0111 0000 0000 0001 C 0 1110 0000 0000 0011 C 1 1100 0000 0000 0110
MOV AX,0B800H ROR AX,3	ROR	AX and CF(Carry Flag) per rotation Initialization  1st Shift  2nd Shift  3rd Shift  Rotation in Binary C 1011 1000 0000 0000 0 C 0101 1100 0000 0000 0 C 0010 1110 0000 0000 0 C 0001 0111 0000 0000 0

STC
MOV AX, 0B800H
RCR AX,3

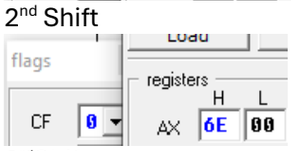
RCR

AX and CF(Carry Flag) per rotation
Initialization:

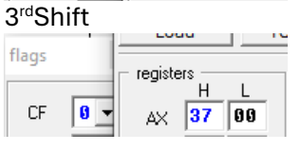


1st Shift

2nd Shift



3rdShift



Rotation in Binary

	C
1011 1000 0000 0000	1
1101 1100 0000 0000	0
0110 1110 0000 0000	0
0011 0111 0000 0000	0

SEARCH AND COMPARE STRING

```
.MODEL SMALL
.DATA
STRING1 DB 'ABCDEFGHI'
STRING2 DB 'FOUND ', '$'
STRING3 DB 'NOT FOUND ', '$'
.CODE
MOV AX, @DATA
MOV ES, AX ;

LEA DI, STRING1
MOV CX,9


ULIT:
MOV AL, 'J' ; TARGET CHARACTER
SCASB
JE HEY
LOOP ULIT

HELLO:
MOV AX,@DATA
MOV DS,AX
MOV AH,09
MOV DX,OFFSET STRING3
INT 21H
JMP EXIT

HEY:
MOV AX,@DATA
MOV DS,AX
MOV AH,09
MOV DX,OFFSET STRING2
INT 21H

EXIT:
MOV AH,04CH
INT 21H
```

SCASB

SS OF FINAL OUTPUT (CHANGE THE VALUE OF TARGET IF NECESSARY)
EXPECTED OUTPUT:

1. FOR FOUND

```
01 .MODEL SMALL
02 .DATA
03 STRING1 DB 'ABCDEFGHI'
04 STRING2 DB 'FOUND ', '$'
05 STRING3 DB 'NOT FOUND ', '$'
06 .CODE
07 MOV AX, @DATA
08 MOV ES, AX
09
10 LEA DI, STRING1
11 MOV CX, 9
12
13 ULIT:
14 MOV AL, 'E' ;TARGET CHARACTER
```

```
SCR emulator screen (80x25 chars)
FOUND
```

2. NOT FOUND

```
01 .MODEL SMALL
02 .DATA
03 STRING1 DB 'ABCDEFGHI'
04 STRING2 DB 'FOUND ', '$'
05 STRING3 DB 'NOT FOUND ', '$'
06 .CODE
07 MOV AX, @DATA
08 MOV ES, AX
09
10 LEA DI, STRING1
11 MOV CX, 9
12
13 ULIT:
14 MOV AL, 'J' ;TARGET CHARACTER
```

```
SCR emulator screen (80x25 chars)
NOT FOUND
```

```
.MODEL SMALL
.DATA
STRING1 DB 'HELL1'
STRING2 DB 'HELLO', '$'
STRING3 DB 'SAME', '$'
STRING4 DB 'NOT THE SAME', '$'
.CODE
MOV AX, @DATA
MOV DS, AX ;
MOV ES, AX ;
LEA SI, STRING1
LEA DI, STRING2
CLD
MOV CX,5
REPE CMPSB
JE HEY

HELLO:
MOV AX,@DATA
MOV DS,AX
MOV AH,09
MOV DX,OFFSET STRING4
INT 21H
JMP EXIT

HEY:
MOV AX,@DATA
MOV DS,AX
MOV AH,09
MOV DX,OFFSET STRING3
INT 21H

EXIT:
MOV AH,04CH
INT 21H
```

CMPSB
norbe

SS OF CONTENT OF STRING1 AND STRING2
AND FINAL OUTPUT
NOTE: CHANGE THE CONTENT OF STRING1 IF NECESSARY TO DISPLAY EITHER “SAME” OR “NOT THE SAME”

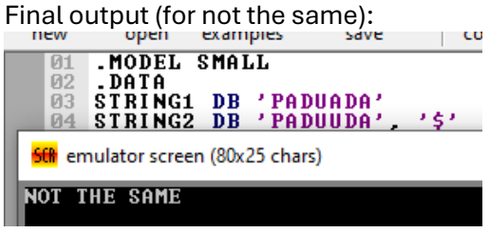
Content of String 1:

```
STRING1 DB 'PADUADA'
```

Content of String 2:

```
STRING2 DB 'PADUUDA', '$'
```

Final output (for not the same):



Final output (for same):