

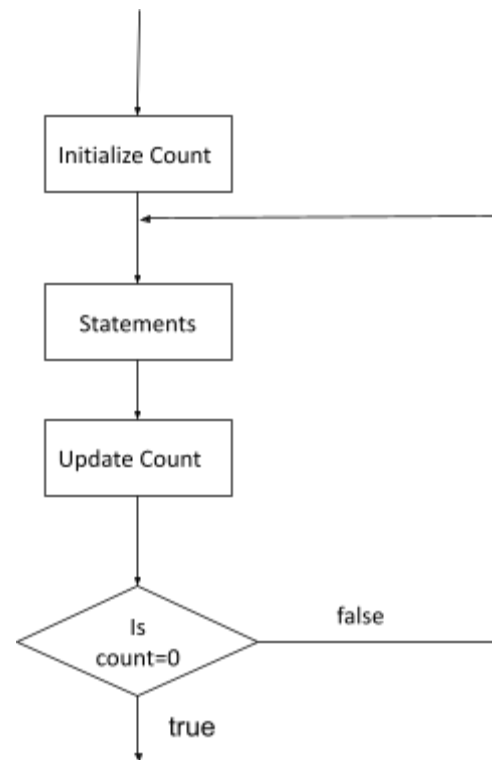


Example 1: Write a count-controlled loop to print a row of 80 '#'s

```
MOV DL, '#'
MOV AH, 2
MOV CX, 80

_LOOP:
INT 21H

LOOP _LOOP
```



Example2: Write a program to take 8 single-key inputs

```
MOV AH, 1
MOV CX, 8

INPUT:
INT 21H

LOOP INPUT
```

Example3: write a program to print the first five digits (0,1,2,3,4)

```
MOV AH, 2
MOV CX, 5
MOV DL, 30H

_LOOP:
INT 21H
INC DL

LOOP _LOOP
```

Example 4: Declare an array of size 10 without any initial data. Prompt the user to enter a line of text and store it into the array.

```
01
02 .MODEL SMALL
03 .STACK 100H
04 .DATA
05     MSG DB "Enter a text$"
06     ARRAY DB 10 DUP(?), '$'
07     NEWL DB 0AH, 0DH, '$'
08 .CODE
09     MOV AX, @DATA
10     MOV DS, AX
11
12     MOV CX, 10
13     MOV AH, 1
14     LEA SI, ARRAY
15
16     _IN:
17     INT 21H
18     MOV [SI], AL
19     INC SI
20
21     LOOP _IN
22
23     MOV AH, 9
24     LEA DX, NEWL
25     INT 21H
26     LEA DX, ARRAY
27     INT 21H
28
29     MOV AH, 4CH
30     INT 21H
31
```

| <u>Unconditional Jump:</u> | <u>Conditional Jump:</u> |
|---|---|
| Example: Write a program to take single-key inputs. | Example: Write a program to take 10 single-key inputs. |
| <pre> 04 05 MOV AH, 1 06 07 INPUT: 08 INT 21H 09 JMP INPUT 10 11 MOV AH, 4CH 12 INT 21H 13 </pre> | <pre> 04 05 MOV AH, 1 06 MOV CX, 10 07 08 INPUT: 09 INT 21H 10 DEC CX 11 JNZ INPUT 12 13 MOV AH, 4CH 14 INT 21H 15 </pre> |

| Signed | Unsigned |
|--|--|
| JE/JZ Jump Equal or Jump Zero JNE/JNZ Jump not Equal or Jump Not Zero JG/JNLE Jump Greater or Jump Not Less/Equal JGE/JNL Jump Greater/Equal or Jump Not Less JL/JNGE Jump Less or Jump Not Greater/Equal JLE/JNG Jump Less/Equal or Jump Not Greater | JE/JZ Jump Equal or Jump Zero JNE/JNZ Jump not Equal or Jump Not Zero JA/JNBE Jump Above or Jump Not Below/Equal JAE/JNB Jump Above/Equal or Jump Not Below JB/JNAE Jump Below or Jump Not Above/Equal JBE/JNA Jump Below/Equal or Jump Not Above |

Example 7: Write a program to take 10 single-key inputs. Terminate the program if the number of user input keys exceed the given size or user inputs a carriage return.

```
04
05 MOV AH, 1
06 MOV CX, 10
07
08 INPUT:
09 INT 21H
10
11 CMP AL, 0DH
12 JE EXIT
13
14 DEC CX
15 JNZ INPUT
16
17 EXIT:
18 MOV AH, 4CH
19 INT 21H
20
21
```

Example 8: Take a user input; if the input is character '1' display 'O', if it is '2', display "E", if it is anything else, do nothing.

```
04
05 MOV AH, 1
06 MOV CX, 10
07 INT 21H
08
09 CMP AL, '1'
10 JE PRINT_O ;if(input==1), goto label(line 18)
11 ;else goto next instruction(line 13)
12
13 CMP AL, 32H
14 JE PRINT_E ;if(input==2), goto label(line 25)
15 ;else goto next instruction(line 16)
16 JMP EXIT ;goto exit & thus skip line 18-28
17
18 PRINT_O:
19 MOV AH, 2
20 MOV DL, 'O'
21 INT 21H ;print 'O'
22
23 JMP EXIT ;goto exit & thus skip line 25-28
24
25 PRINT_E:
26 MOV AH, 2
27 MOV DL, 'E'
28 INT 21H ;print 'E'
29
30 EXIT:
31 MOV AH, 4CH
32 INT 21H
33
```

TEST:

| | |
|--|--|
| <p>Format: TEST destination, source</p> <p>Example: TEST AL, 1</p> <p>**TEST and AND are similar, the only difference is TEST doesn't write the result of the operation on destination.</p> | <p>AL: 1001 1011 (155) 0000 0001 ----- 0000 0001 1 => not zero => odd</p> <p>AL: 1001 1010 (154) 0000 0001 ----- 0000 0000 0 => zero => even</p> |
|--|--|

Example: Read a character and check if the input contains an even number. If it is even, print 'e' otherwise do nothing.

```
MOV AH, 1
INT 21H
XOR AH, AH ;set ah to 0
MOV BL, 2

DIV BL ;AX/BL: quotient in al,
;remainder in ah

CMP AH, 0 ;check if remainder is 0
;means even number
JE PRINT_E ;goto label

JMP EXIT ;if not even, goto exit

PRINT_E:
MOV AH, 2
MOV DL, 'E'
INT 21H ;print 'E'

EXIT:
MOV AH, 4CH
INT 21H
```

```
MOV AH, 1
INT 21H

TEST AL, 1
;checks if LSB is zero
JZ PRINT_E

JMP EXIT
;if not even, goto exit

PRINT_E:
MOV AH, 2
MOV DL, 'E'
INT 21H

EXIT:
MOV AH, 4CH
INT 21H
```