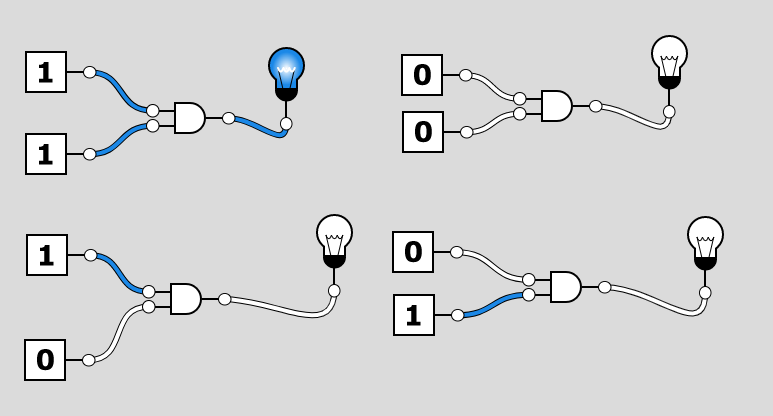
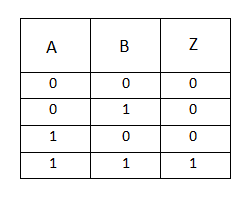
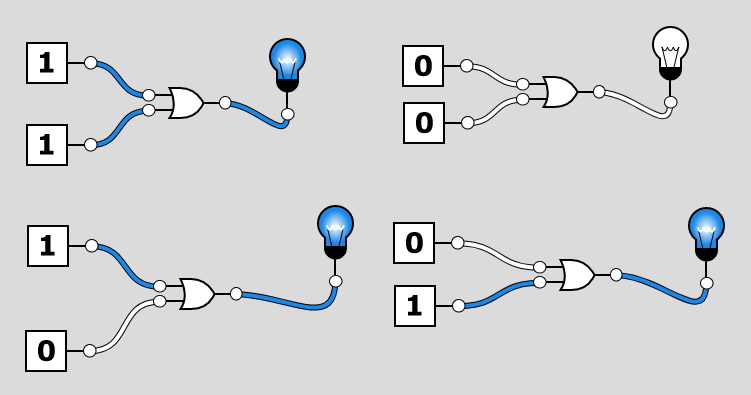
**EXPT 2**

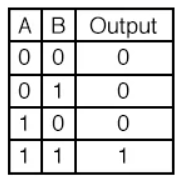
**AND GATE:**

****

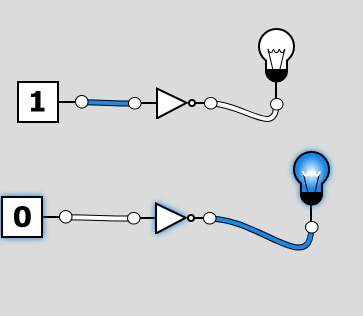
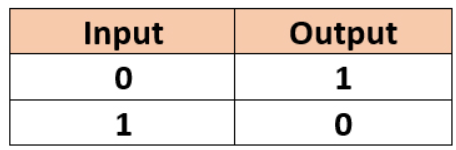
****

**OR GATE:**

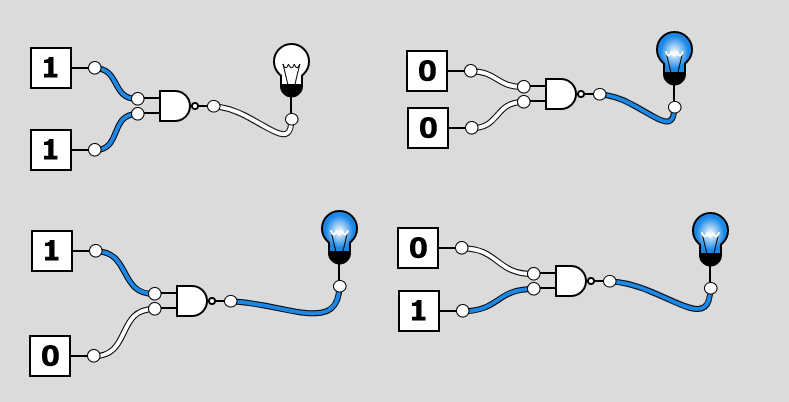
****

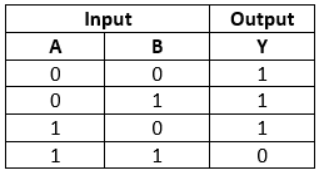


**NOT GATE:**

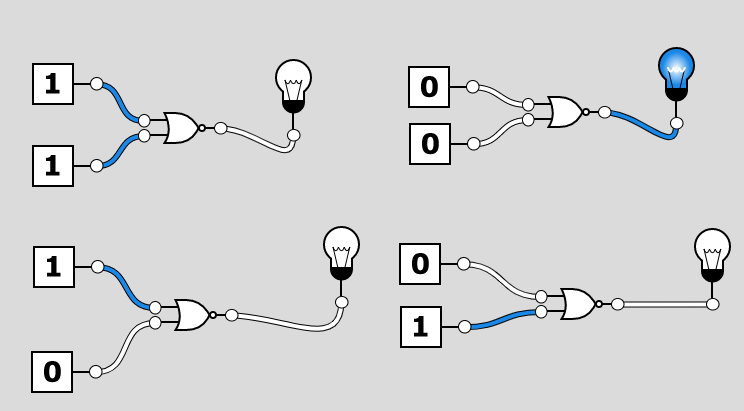
**** 

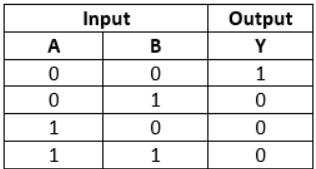
**NAND GATE:**

****

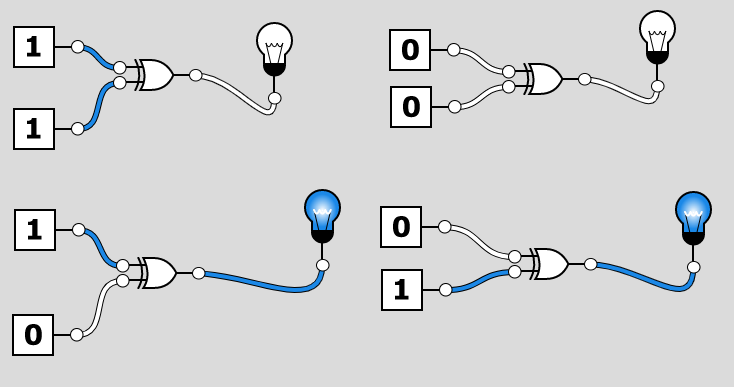


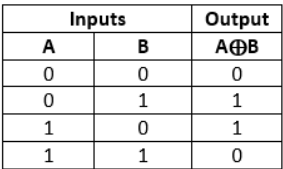
**NOR GATE:**

****

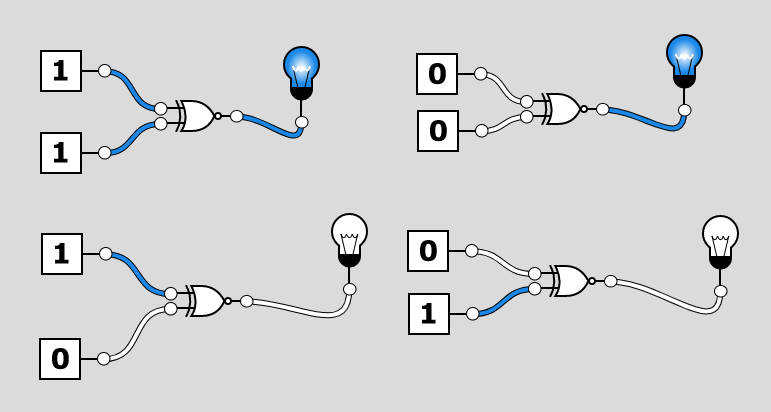


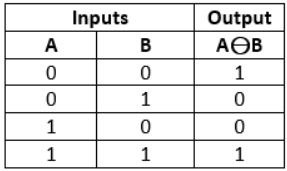
**XOR GATE:**

****

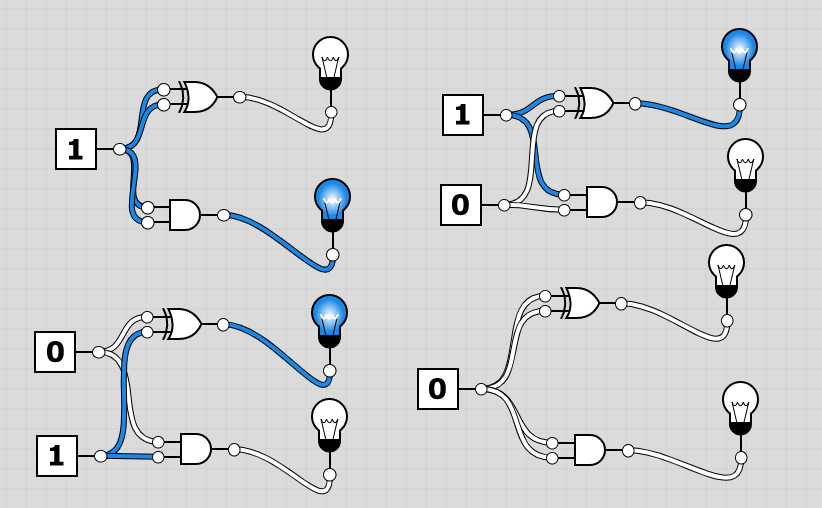


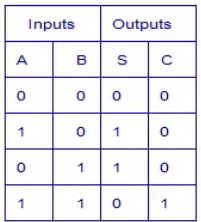
**XNOR GATE:**

****

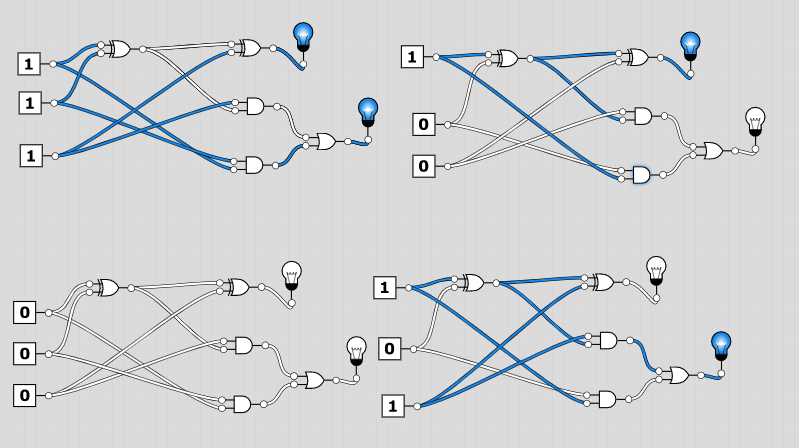


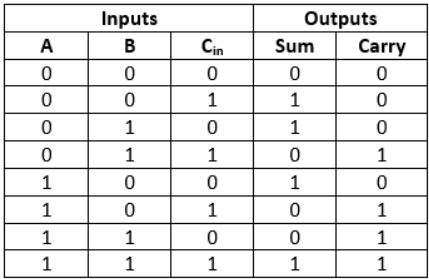
**HALF ADDER:**

****

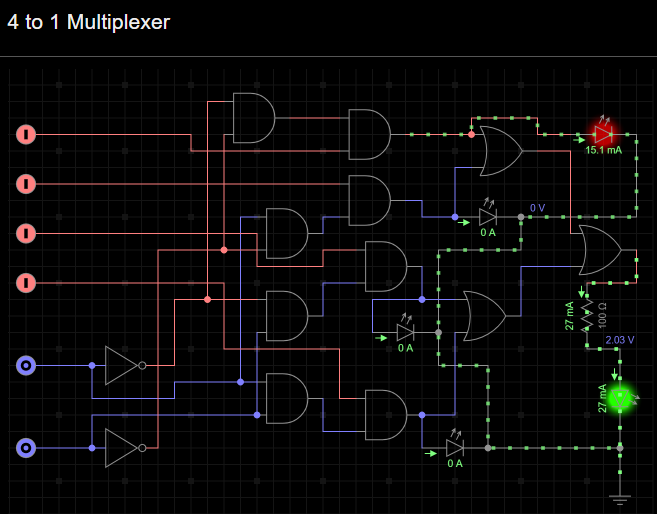


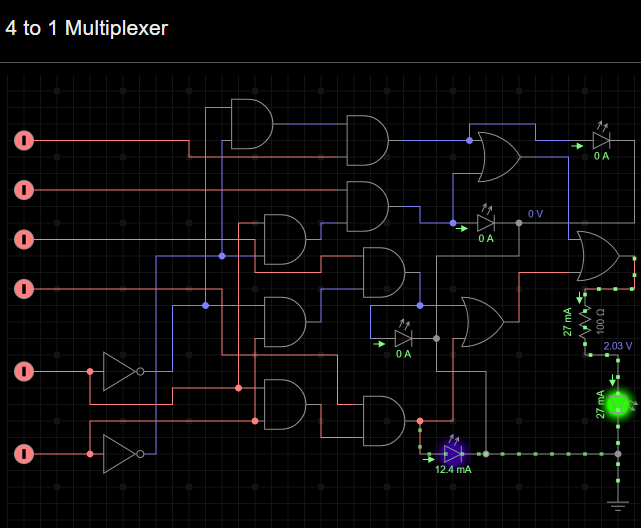
**FULL ADDER:**

****

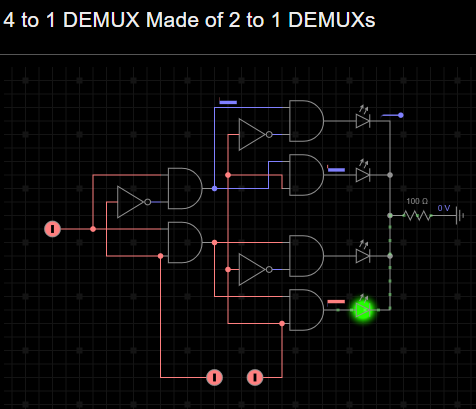


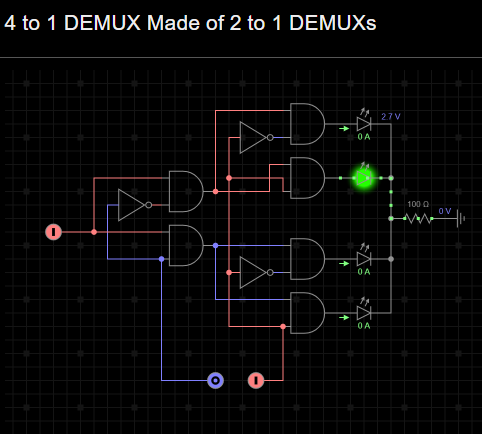
**MULTIPLEXER**

****

****

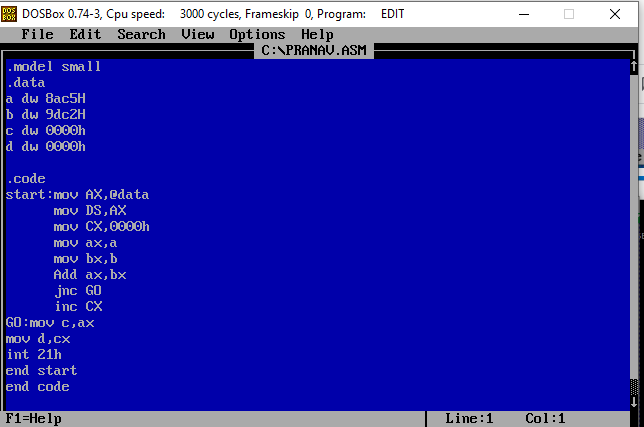
**DEMULTIPLEXER**

****

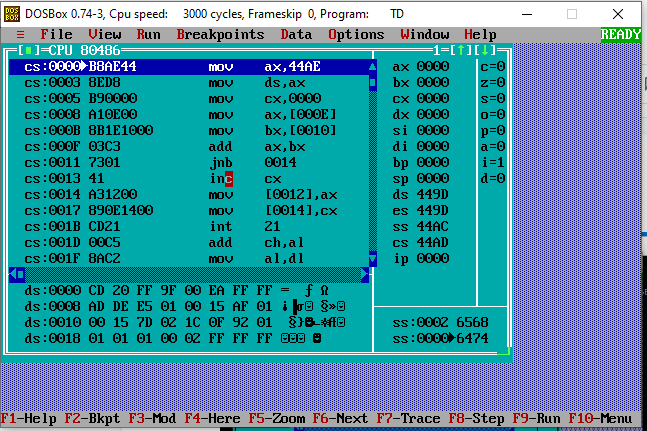
****

**EXPT 3**

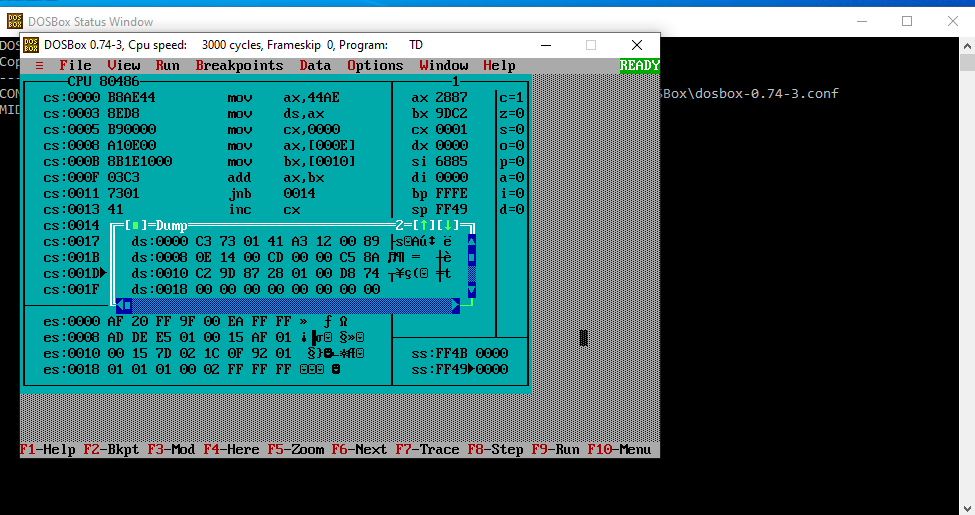
**Input –**



**Output –**



**Dump -**



**EXPT 4**

**1.Program to count number of 1’s and 0’s in**

**a given 8 bit number**

**Input –**

DATA SEGMENT

NO DW 5648H

Z DW ?

O DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, NO

MOV BX, 00H

MOV CX, 10H

MOV DX, 00H

UP:

ROL AX,1

JC ONE

INC BX

JMP NXT

ONE:

INC DX

NXT:

DEC CX

JNZ UP

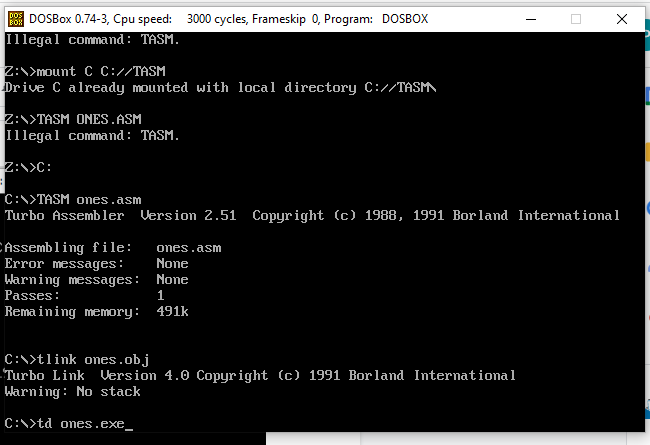
MOV Z, BX

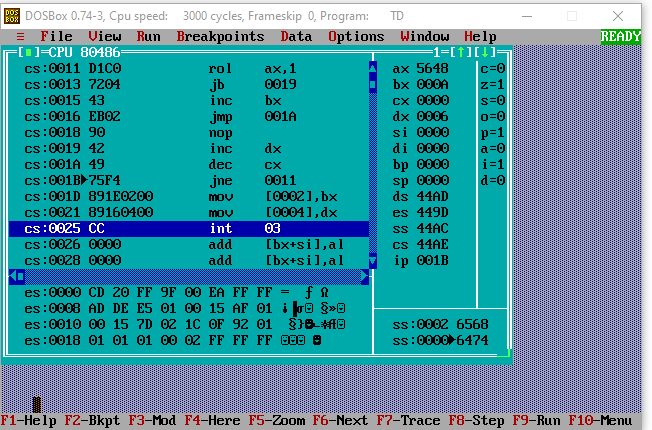
MOV O, DX

INT 3

CODE ENDS

END START





**2.Program to find even and odd numbers**

**from a given list:**

**CODE :**

.model small

.stack 100h

.data

ev db 'Even$'

od db 'Odd$'

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

mov bl,2

div bl

cmp ah,0

je IsEven

mov dx,10

mov ah,2

int 21h

mov dx,13

mov ah,2

int 21h

mov dx,offset od

mov ah,9

int 21h

mov ah,4ch

int 21h

IsEven:

mov dx,10

mov ah,2

int 21h

mov dx,13

mov ah,2

int 21h

mov dx,offset ev

mov ah,9

int 21h

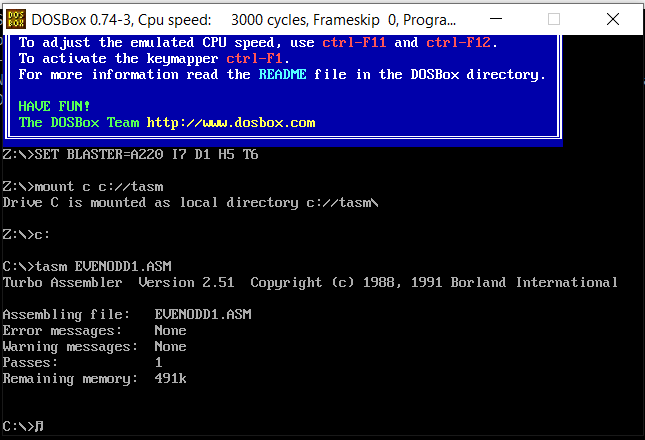
mov ah,4ch

int 21h

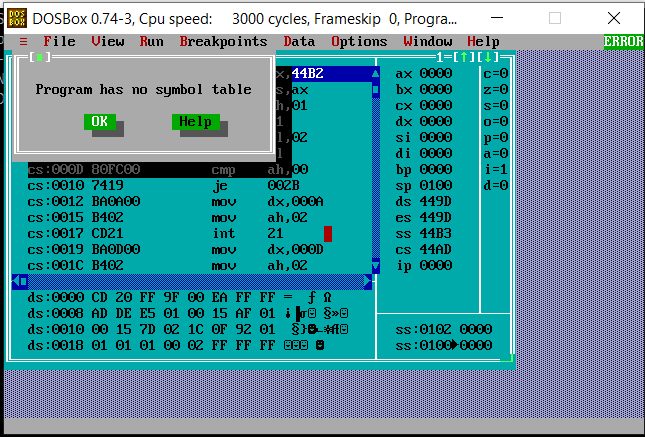
main endp

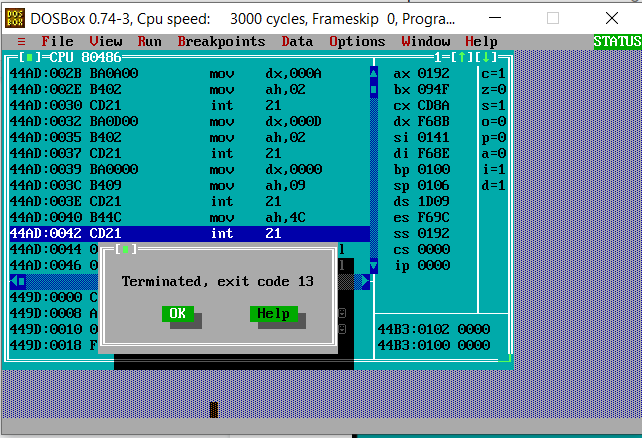
end main

**Execution Of ASM File:**

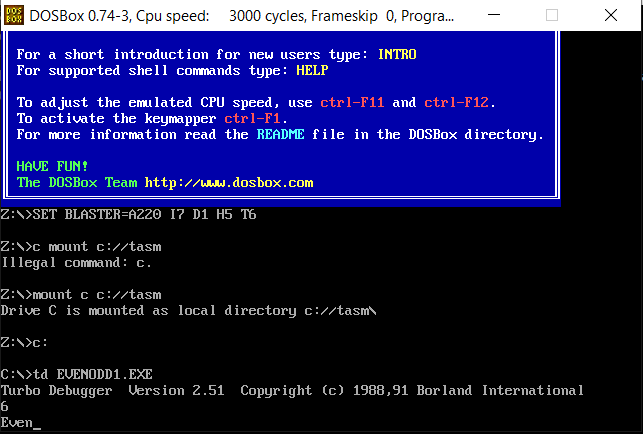
****

**Initial Output Window:**

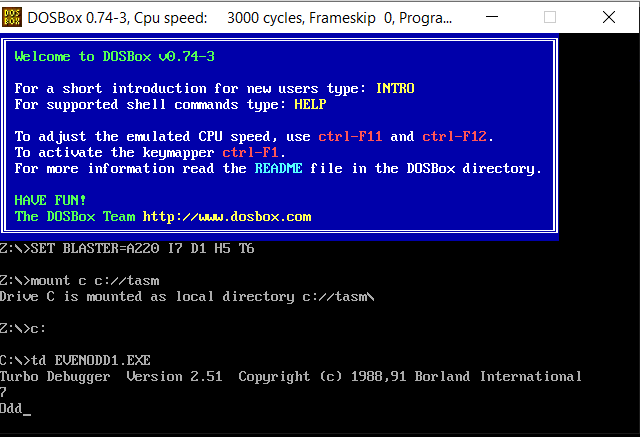
****

****

**FOR EVEN**



**FOR ODD**

****

**EXPT 5**

**CODE:**

DATA SEGMENT

BLOCK1 DB 'MALAYALAM'

MSG1 DB "IT IS PALINDROME $"

MSG2 DB "IT IS NOT PALINDROME $"

PAL DB 00H

DATA ENDS

PRINT MACRO MSG

MOV AH,09H

LEA DX,MSG

INT 21H

INT 3H

ENDM

EXTRA SEGMENT

BLOCK2 DB 9 DUP(?)

EXTRA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,ES:EXTRA

START: MOV AX,DATA

MOV DS,AX

MOV AX,EXTRA

MOV ES,AX

LEA SI,BLOCK1

LEA DI,BLOCK2+8

MOV CX,00009H

BACK: CLD

LODSB

STD

STOSB

LOOP BACK

LEA SI,BLOCK1

LEA DI,BLOCK2

MOV CX,0009H

CLD

REPZ CMPSB

JNZ SKIP

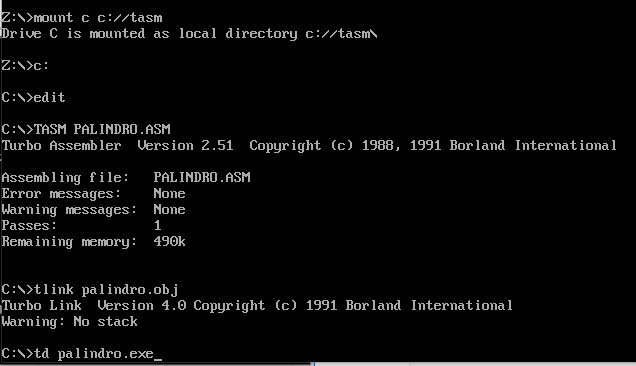
PRINT MSG1

SKIP: PRINT MSG2

CODE ENDS

END START

**Execution Of ASM File:**



**Initial Output Window:**

