.model small .stack 100h

These directives set up the memory model and stack size for the program.

.data ev db 'Even$' od db 'Odd$'

These lines declare two character strings: "Even" and "Odd". The $ character marks the end of the string.

.code main proc mov ax,@data mov ds,ax

This section declares the code segment and defines the entry point of the program. It also sets up the data segment register (DS) with the address of the data segment.

mov ah,1 int 21h

These instructions use the DOS interrupt 21h to read a single character from the keyboard. The character is returned in the AL register.

mov bl,2 div bl cmp ah,0 je IsEven

These instructions divide the value in AL by 2 using the DIV instruction. If the remainder (AH) is zero, the number is even and the program jumps to the IsEven label. If the remainder is non-zero, the number is odd and the program continues to the next instruction.

mov dx,10 mov ah,2 int 21h mov dx,offset od mov ah,9 int 21h mov ah,4ch int 21h

If the number is odd, these instructions display the "Odd" message on the screen using the DOS interrupt 21h. The program then terminates using the INT 21h, function 4Ch instruction.

IsEven: mov dx,10 mov ah,2 int 21h mov dx,13 mov ah,2 int 21h

If the number is even, these instructions move the cursor to the beginning of the next line using the carriage return and line feed characters. This is done using the MOV and INT 21h, function 2 instructions.

mov dx,offset ev mov ah,9 int 21h mov ah,4ch int 21h

These instructions display the "Even" message on the screen using the DOS interrupt 21h. The program then terminates using the INT 21h, function 4Ch instruction.

main endp end main

These instructions mark the end of the main procedure and the end of the program, respectively.