ASSEMBLLY CODE:

```
.MODEL SMALL
.STACK 100H
.DATA
    MSGA DB 'Enter a number: $'
    MSGB DB 'f91($'
    MSGC DB ') = $'
    MSGD DB ' 91 $'
    NEWL DB 13,10,'$'
    INPUT DW ?
    TEMP DW ?
    FLAG DW ?

.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

RAND_IO:
    LEA DX,MSGA                     ;Print 'Enter a number:'
    MOV AH,9
    INT 21H

    MOV DX,0

    MOV BX,0                        ;Clear bx
    MOV AH,1
    INT 21H

    CMP AL,0DH
    JE END_INPUTS                   ;If enter

    CONVERT_TO_NUM:
        AND AX,000FH                ;Use full 16 bits of AX
        MOV TEMP,AX
        MOV AX,10
        MUL BX
        MOV BX, AX
        ADD BX,TEMP

    MOV AH,1                        ;Input new digit
    INT 21H
    CMP AL,0DH
    JNE CONVERT_TO_NUM

    END_INPUTS:
        MOV INPUT,BX
        MOV TEMP,BX

    FIRST_CON:                      ;If input equal zero then break
        CMP INPUT,0
        JE END_LINK

    SECOND_CON:                     ;If input equal or less then 100 then print 91
        CMP INPUT,100
        JLE FIRST_PRINT

        MOV FLAG,2
        JMP SECOND_PRINT            ;Else print input-10


FIRST_PRINT:
    LEA DX,MSGB                     ;Print 'f91('
    MOV AH,9
    INT 21H
```

```
        PRINT_91:
            MOV AX,INPUT
            MOV CX,0
            MOV BX,10

            STOR_RESULTS:               ;stor each digits in stack
                XOR DX,DX
                DIV BX
                PUSH DX
                INC CX
                CMP AX,0
                JNE STOR_RESULTS

            PRINT_RESULTS:              ;print each digits from stack
                MOV AH,2
                POP DX
                ADD DL,48
                INT 21H
                LOOP PRINT_RESULTS

            LEA DX,MSGC                 ;Print ') = '
            MOV AH,9
            INT 21H

            LEA DX,MSGD                 ;Print '91 '
            MOV AH,9
            INT 21H

            LEA DX,NEWL                 ;Print 'new line'
            MOV AH,9
            INT 21H
            JMP RAND_IO

    END_LINK:
        JMP END_IO

    SECOND_PRINT:
        LEA DX,MSGB                     ;Print 'f91('
        MOV AH,9
        INT 21H

        GET_OUTPUT:
            MOV AX,INPUT
            MOV CX,0
            MOV BX,10

            STOR_OUTPUT:                ;stor each digits in stack
                XOR DX,DX
                DIV BX
                PUSH DX
                INC CX
                CMP AX,0
                JNE STOR_OUTPUT

            PRINT_OUTPUT:               ;print each digits from stack
                MOV AH,2
                POP DX
                ADD DL,48
                INT 21H
                LOOP PRINT_OUTPUT

            DEC FLAG
            CMP FLAG,1
            JE CLOSE_B
            JMP OUTPUT
```

```
        CLOSE_B:
            LEA DX,MSGC             ;Print ') = '
            MOV AH,9
            INT 21H

        OUTPUT:
            SUB INPUT,10
            CMP FLAG,0
            JNE GET_OUTPUT          ;for printing result

            LEA DX,NEWL             ;Print 'new line'
            MOV AH,9
            INT 21H
            JMP RAND_IO

END_IO:
    MOV AH,4CH
    INT 21H

MAIN ENDP
    END MAIN
```