```
.MODEL SMALL
.STACK 100H
.DATA
   MSGA DB 'First Number: $'
    MSGB DB 'Second Number: $'
    MSGC DB 'Relation: $'
    MSGD DB 13,10,'$'
    MSGE DB 'What time do you want to test: $'
    LESS DB '< $'
    GRET DB '> $'
    EQUA DB '= $'
    COUNT DB ?
    FIRST DW ?
    SECOND DW ?
    TEMP DW ?
    CASE DW ?
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX
TEST_CASE:
                             ;Test case
    LEA DX, MSGE
    MOV AH, 9
    INT 21H
    MOV DX, 0
    MOV BX, 0
    MOV AH, 1
    INT 21H
    CMP AL, ODH
    JE END_CASE_INPUT
        CONVERT_TO_NUM_CASE:
            AND AX,000FH
            MOV TEMP, AX
            MOV AX, 10
            MUL BX
            MOV BX, AX
            ADD BX, TEMP
        MOV AH, 1
                                 ;
        INT 21H
        CMP AL, ODH
        JNE CONVERT_TO_NUM_CASE
        END_CASE_INPUT:
            MOV CASE, BX
RAND_IO:
    MOV COUNT, 2
    LEA DX, MSGA
    MOV AH, 9
    INT 21H
    GET_INPUTS:
                                ;Get Input
        MOV DX, 0
        MOV BX, 0
        MOV AH, 1
        INT 21H
        CMP AL, ODH
        JE END_INPUTS
```

```
CONVERT_TO_NUM:
        AND AX,000FH
        MOV TEMP, AX
        MOV AX, 10
        MUL BX
        MOV BX, AX
        ADD BX, TEMP
    MOV AH, 1
    INT 21H
    CMP AL, ODH
    JNE CONVERT_TO_NUM
    END_INPUTS:
        CMP COUNT, 1
        JNE GET_FIRST
        MOV SECOND, BX
        JMP GET_SECOND
        GET_FIRST:
            MOV FIRST, BX
            LEA DX, MSGB
            MOV AH, 9
            INT 21H
        GET_SECOND:
            DEC COUNT
            CMP COUNT, 0
            JNE GET_INPUTS
            JMP GET_RESULT
NEW_INPUT_BRIDGE:
                                           ; This for next case
    JMP RAND_IO
GET_RESULT:
    LEA DX, MSGC
    MOV AH, 9
    INT 21H
    MOV BX, FIRST
    MOV AX, SECOND
    CMP BX, SECOND
    JGE FIRST_CON
                                           ;Gretter then or equal
    JMP THIRD_CON
    FIRST_CON:
        CMP BX, SECOND
        JE SECOND_CON
                                           ; Equal or not
        LEA DX, GRET
        MOV AH, 9
        INT 21H
        JMP NEW_INPUT
        SECOND_CON:
            LEA DX, EQUA
            MOV AH, 9
            INT 21H
            JMP NEW_INPUT
    THIRD_CON:
                                           ;Less
        LEA DX, LESS
        MOV AH, 9
```

```
INT 21H
JMP NEW_INPUT:

LEA DX, MSGD

MOV AH, 9

INT 21H

INT 21H

DEC CASE

CMP CASE, 0

JNE NEW_INPUT_BRIDGE

MAIN ENDP

END MAIN
```