

```
.MODEL SMALL
.STACK 100H
.DATA
    MSGA DB 'Length of Box: $'
    MSGB DB 'Wight of Box: $'
    MSGC DB 'Hight of Box: $'
    MSGD DB ': The Box is GOOD for Suitcase: $'
    MSGE DB ': The Box is BAD for Suitcase: $'
    MSGF DB 'What time do you want to test: $'
    MSGG DB 13,10,'$'
    MSGH DB 'Case: $'
    COUNT DB ?
    BLength DW ?
    BWIGHT DW ?
    BHIGHT DW ?
    TEMP DW ?
    CASE DW ?
    C_INDEX DW ?

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

TEST_CASE:
    LEA DX,MSGF
    MOV AH,9
    INT 21H

    MOV DX,0
    MOV BX,0
    MOV C_INDEX,1
    MOV AH,1
    INT 21H

    CMP AL,0DH
    JE TEST_INPUTS

CON_TO_NUM_FOR_TEST:
    AND AX,000FH
    MOV TEMP,AX
    MOV AX,10
    MUL BX
    MOV BX,AX
    ADD BX,TEMP

    MOV AH,1
    INT 21H
    CMP AL,0DH
    JNE CON_TO_NUM_FOR_TEST

TEST_INPUTS:
    MOV CASE,BX

RAND_IO:
    CMP CASE,0
    JE END_PRO_BRIDGE

    MOV COUNT,3
    LEA DX,MSGA
    MOV AH,9
    INT 21H

GET_INPUTS:
    MOV DX,0
    MOV BX,0
```

```
MOV AH,1
INT 21H

CMP AL,0DH
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,0DH
JNE CONVERT_TO_NUM

END_INPUTS:
    CMP COUNT,3
    JNE GET_WIGHT

    GET_LENGTH:
        MOV BLength,BX
        LEA DX,MSGB
        MOV AH,9
        INT 21H

        DEC COUNT
        CMP COUNT,1
        JMP GET_INPUTS

    GET_WIGHT:
        CMP COUNT,2
        JNE GET_HIGHT
        MOV BWIGHT,BX
        LEA DX,MSGC
        MOV AH,9
        INT 21H

        DEC COUNT
        CMP COUNT,0
        JNE GET_INPUTS

    GET_HIGHT:
        MOV BHIGHT,BX
        JMP GET_RESULT

END_PRO_BRIDGE:
    JMP END_PRO

GET_RESULT:
    LEA DX,MSGH
    MOV AH,9
    INT 21H

    PRINT_CASE_NUM:
        JMP CASE_NUMBER

BACK_TO_RESULT:
    CMP BLength,20
    JLE CHECK_WIGHT
    JMP PRINT_BAD

    CHECK_WIGHT:
```

```
        CMP BWIGHT,20
        JLE CHECK_HIGHT
        JMP PRINT_BAD

        CHECK_HIGHT:
            CMP BHIGHT,20
            JLE PRINT_GOOD
            JMP PRINT_BAD

PRINT_GOOD:
    LEA DX,MSGD
    MOV AH,9
    INT 21H
    JMP NEW_INPUT

PRINT_BAD:
    LEA DX,MSGE
    MOV AH,9
    INT 21H
    JMP NEW_INPUT

CASE_NUMBER:
    MOV AX,C_INDEX
    MOV CX,0
    MOV BX,10

    STOR_RESULTS:
        XOR DX,DX
        DIV BX
        PUSH DX
        INC CX
        CMP AX,0
        JNE STOR_RESULTS

    PRINT_RESULTS:
        MOV AH,2
        POP DX
        ADD DL,48
        INT 21H
        LOOP PRINT_RESULTS

        INC C_INDEX
        JMP BACK_TO_RESULT

NEW_INPUT:
    LEA DX,MSGG
    MOV AH,9
    INT 21H
    INT 21H
    DEC CASE
    JMP RAND_IO

END_PRO:
    MOV AH,4CH
    INT 21H

MAIN ENDP
END MAIN
```