```
ASSEMBLLY CODE:
    .MODEL SMALL
    .STACK 100H
    .DATA
        MSGA DB 'Enter Vertical size(M): $'
        MSGB DB 'Enter Horizontal size(N): $'
        MSGC DB 'Minimum Cuts: $'
        MSGD DB 13,10,'$'
        COUNT DB ?
        M DW ?
        N DW ?
        CUT DW ?
        TEMP DW ?

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

    RAND_IO:
        MOV COUNT,2
        LEA DX,MSGA                     ;Print 'Enter Vertical size:'
        MOV AH,9
        INT 21H

        GET_INPUTS:
            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:
                CMP COUNT,1
                JNE GET_M

                MOV N,BX                ;Get Time
                JMP GET_N

                GET_M:
                    MOV M,BX            ;Get M
                    LEA DX,MSGB         ;Print 'Enter Horizontal size(N): '
                    MOV AH,9
                    INT 21H

                GET_N:
                    DEC COUNT
                    CMP COUNT,0
                    JNE GET_INPUTS
        GET_MIN_CUTS:
            MOV BX,M
            MOV AX,N
            MUL BX                      ;AX = AX*BX
            MOV CUT,AX                  ;Get result in CUT
            DEC CUT

        START_PRINT:
            LEA DX,MSGC                 ;Print Cuts
            MOV AH,9
            INT 21H

            MOV AX,CUT
            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

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

    MAIN ENDP
        END MAIN
```