%include "along32.inc"  
  
global main  
  
section .data   
 hex dd 0  
 lsb dd 0  
 msb dd 0  
 total dd 0  
   
section .rodata   
  
 startMsg db "Enter a hexadecimal: ", 0x0  
 LSBMsg db "The least significant bit set is ", 0x0  
 MSBMsg db "The most significant bit set is ", 0x0  
 TotMsg db "The total number of 1 bit sets is ", 0x0  
 nl db 0xa, 0x0   
   
section .text  
  
main:   
 mov edx, startMsg   
 call WriteString   
 call ReadHex   
 mov [hex], eax   
   
 bsf ebx, eax   
 mov [lsb], ebx   
  
 mov eax, [hex]   
 bsr ebx, eax   
 mov [msb], ebx   
   
 mov eax, [hex]   
 mov ecx, 0   
   
loop:   
 shr eax, 1   
 jc tot   
 jmp check   
   
tot:   
 mov ebx, [total]   
 inc ebx   
 mov [total], ebx   
   
check:   
 inc ecx   
 cmp ecx, 32  
 jge write   
 jmp loop  
  
write:   
 mov edx, LSBMsg   
 call WriteString  
 mov eax, [lsb]  
 call WriteInt   
 mov edx, nl   
 call WriteString  
   
 mov edx, MSBMsg   
 call WriteString  
 mov eax, [msb]  
 call WriteInt   
 mov edx, nl  
 call WriteString  
   
 mov edx, TotMsg   
 call WriteString  
 mov eax, [total]  
 call WriteInt   
 mov edx, nl  
 call WriteString  
 jmp exit   
   
  
exit:   
 mov eax, 01h   
 mov ebx, 0h   
  
 int 80h