global main	program oszto
extern read_unsigned, write_unsigned	
extern read_boolean, write_boolean	
segment .bss	
label0: resd 1	integer a;
label1: resd 1	integer i;
label2: resd 1	integer oszto;
label3: resb 1	boolean vanoszto;
segment .text	begin
main:	
call read_unsigned	read(a);
mov [label0], eax	
mov al, 0	vanoszto := false;
mov [label3], al	
mov eax, 2	i := 2;
mov [label1], eax	
label5:	
mov eax, [label0]	while not vanoszto and i < a do
push eax	
mov eax, [label1]	
pop ebx	
cmp eax, ebx	
setl al	
push ax	
mov al, [label3]	
xor al, 1	
pop bx	
and al, bl	
cmp al, 1	
jne near label6	if a mod i = 0 then
mov eax, 0	
push eax	
xor edx, edx	
mov eax, [label1]	
push eax	
mov eax, [label0]	
pop ebx	
div ebx	
mov eax, edx	
pop ebx	
cmp eax, ebx	
sete al	
cmp al, 1	

jne near label4	
mov al, 1	vanoszto := true;
mov [label3], al	
mov eax, [label1]	oszto := i;
mov [label2], eax	
label4:	endif
mov eax, 1	i := i+1;
push eax	
mov eax, [label1]	
pop ebx	
add eax, ebx	
mov [label1], eax	
jmp label5	
label6:	done
mov al, [label3]	if vanoszto then
cmp al, 1	
jne near label7	
xor eax, eax	
mov al, [label3]	
push eax	
call write_boolean	write(vanoszto);
add esp,4	
mov eax, [label2]	
push eax	
call write_unsigned	write(oszto);
add esp,4	else
jmp label8	
label7:	
xor eax, eax	
mov al, [label3]	
push eax	
call write_boolean	write(vanoszto);
add esp,4	endif
label8:	end
ret	