%include “asm\_io.inc”

; Görkem Ertaş / 20070006004

segment .data

msg db “Enter an integer : “, 0

msg1 db “The number of integers divisible by 2 is : “,0

msg2 db “The number of integers divisible by 3 is : “,0

msg3 db “The number of integers divisible by 5 is : “,0

divisblebytwo dd 0

divisiblebythree dd 0

divisiblebyfive dd 0

segment. bss

segment .text

global \_asm.main

\_asm\_main:

enter 0,0

pusha

while:

mov edx, 0

mov eax, msg

call print\_string

call read\_int

mov ebx, eax

cmp eax, 0

jge div2block

jmp elseblock

div2block:

mov edx, 0

mov eax, ebx

mov cx, 2

idiv cx

cmp dx, 0

je increase2block

div3block:

mov edx, 0

mov eax, ebx

mov cx, 3

idiv cx

cmp dx, 0

je increase3block

div5block:

mov edx, 0

mov eax, ebx

mov cx, 5

idiv cx

cmp dx, 0

je increase5block

jmp while

increase2block:

mov eax, [divisiblebytwo]

inc eax

mov [divisiblebytwo], eax

jmp div3block

increase3block:

mov eax, [divisiblebythree]

inc eax

mov [divisiblebythree], eax

jmp div5block

increase5block:

mov eax, [divisiblebyfive]

inc eax

mov [divisiblebyfive], eax

jmp while

elseblock:

jmp end\_while

end\_while:

mov eax, msg1

call print\_string

mov eax, 0

mov eax, [divisiblebytwo]

call print\_int

call print\_nl

mov eax, msg2

call print\_string

mov eax, 0

mov eax, [divisiblebythree]

call print\_int

call print\_nl

mov eax, msg3

call print\_string

mov eax, 0

mov eax, [divisiblebyfive]

call print\_int

popa

mov eax, 0 ; return back to C

leave

ret