

Auxiliar 4
Rodrigo Cánovas
10 de Octubre del 2008

1. Problema 1

El siguiente es un programa en assembler x86. Escriba el programa equivalente en C.

<pre>.globl Q Q: pushl %ebp movl %esp, %ebp pushl %esi pushl %ebx movl 8(%ebp), %ecx # 1er. parámetro movl 12(%ebp), %ecx # 2do. parámetro movl 16(%ebp), %ebx # 3er. parámetro movl (%ecx,%ecx,4), %esi movl (%ecx,%ebx,4), %eax movl %eax, (%ecx,%ecx,4) movl %esi, (%ecx,%ebx,4) popl %ebx popl %esi popl %ebp ret .globl P P: pushl %ebp movl %esp, %ebp pushl %edi pushl %esi pushl %ebx subl \$12, %esp movl 8(%ebp), %edi movl 12(%ebp), %ebx movl 16(%ebp), %esi</pre>	<pre># si %ebx>=%esi goto .L10 cmpl %esi, %ebx jge .L10 .L8: # si (%edi,%ebx,4)>=0 goto .L6 cmpl \$0, (%edi,%ebx,4) jns .L6 addl \$1, %ebx jmp .L3 .L6: subl \$4, %esp pushl %esi # 3er. arg pushl %ebx # 2do. arg pushl %edi # 1er. arg call Q subl \$1, %esi addl \$16, %esp .L3: # si %ebx<%esi goto .L8 cmpl %esi, %ebx jl .L8 .L10: movl %ebx, %eax # valor de ret. leal -12(%ebp), %esp # %esp= %ebp-12 popl %ebx popl %esi popl %edi popl %ebp ret</pre>
---	--