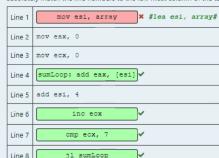
把有7个数的数组array中每一位求和并储存在EAX寄存器中:

Drag-and-drop the correct sequence number of the assembly code to form a program where 7 numbers in an array are added and stored in eax register. Note that your sequence must absolutely match the line numbers to the left-most column of the table. The answers for Lines 2, 3 and 5 have been provided. Complete the rest.



; 将指针ESI指向array**所在地址**

;将累加器设为0

; 将计数器设为0

; sumLoop子程序开始: 将ESI所指地址的值加给EAX

ESI+=4,指针偏移到下一位int(4byte)

; 计数器自增

; 比较ECX与7

如果ECX<7. 则继续子程序(循环体)

jnl sumLoop inc ecx dec ecx lea esi, array jl sumLoop sumLoop: add eax, [esi] cmp ecx, 7 mov esi, array

将一个含有7个int的数组通过冒泡排序排为递增:

Drag-and-drop the correct sequence number of the assembly code to form a program that sorts an array of 7 integers in an ascending order (Bubble Sort). Note that your sequence must absolutely match the line numbers to the left-most column of the table. Complete Lines 4, 7-10. ; 源变址寄存器指向array ; 计数器设置为7 Line 2

outerLoop: mov edx,ecx ; 外循环开始: edx=当前计数器 Line 3 innerLoop: cmp edx, ecx jz noExchange Line 5 Line 6 mov eax, [esi + ecx * 4 - 4] Line 7 cmp ebx, eax Line 9 jnl noExchange x #jl noExchange# mov [esi + ecx * 4], ebx x #mov [esi + ecx * 4 - 4], ebx# ; 不然交换, 即将ebx值赋给eax所在地址 Line 10

; 内循环开始: 比较edx与计数器大小 ;若相等则跳转至noEx子程序

; eax=计算后的地址=array+ecx偏移字数-1

;ebx=计算后的地址=array+edx偏移字数-1 ; 比较ebx与eax的地址的值大小

; 若ebx>eax, 则跳转至noEx子程序

将eax值赋给ebx所在地址

; noEx子程序: edx--

; 如果edx不为0, 继续内循环

; 直到ecx减为0, 停止外循环

mov [esi + ecx * 4 - 4], ebx | cmp ebx, eax | [inl noExchange | [il noExchange | (innerLoop: cmp edx, ecx | (mov ebx, [esi + edx * 4 - 4]) | (mov ebx, [esi + edx * 4]) | (mov ebx, [esi + edx

求数组中所有数之和:

noExchange: dec edx jnz innerLoop

Line 14 | loop outerLoop

Line 12

Line 13

Drag-and-drop the correct arguments and/or instructions to the missing places for the following program that sums all the numbers in an array ;数组大小为5 int arraySize = 5; int intArray[arraySize v] = {12, 3, 7, 23, 9};; 初始化数组, 每次取出为int整数 **v** ; int totalAmt = 0 ;初始化totalAmt归零 ;开始内联汇编 asm{ lea 🗸 edi, intArray 🗸 ; edi指向数组地址 ; 计数器=数组大小 mov ecx, arraySize mov eax, totalAmt ; eax累加器=0 addTotalAmt ✓ : ;addTotalAmt子程序: ✓ eax, [edi] ; eax+=edi所指int add edi, 4 ; 然后edi指向下一个int 100p ✓ addTotalAmt ;循环子程序,直到计数器归零 mov totalAmt ✓ , eax ;将eax累加数值赋给totalAmt

movzx intArray loop arraySize-1 totalAmt arraySize add addTotalAmt 4 [intArray] lea 0 jmp 8

2.将字符串入栈:

