26 Drag-and-drop (for online test) or write the sequence number (for on-site test) of the assembly code to form a program that finds the **sum of even** numbers in an array of **6 integers** and stores this value in the eax register. Note that your sequence must absolutely match the line numbers to the left-most column of the table. The answers for Lines 1,2, and 4 have been provided. Complete the rest.

	Correct Sequence	**Pick From Here**	
Line 1	1	1	`lea esi, myArray`
Line 2	2	2	`mov eax, 0`
Line 3		3	`add eax, [esi]`
Line 4	4	4	`L1: test [esi], 01H`
Line 5		5	`inc eax`
Line 6		6	`L2: inc, esi`
Line 7		7	`add eax, [esi +4]`
Line 8		8	`L2: add esi, 4`
		9	`jnz L2`
		10	`jz L2`
		11	`loop L1`
		12	`loop L2`
		13	`mov ecx, 5`
		14	`mov ecx, 6`

Part II. (40 marks)

26. Drag-and-drop (for online test) or write the sequence number (for on-site test) of the assembly code to form a program that finds the **sum of odd** numbers between **1 and 100** and stores this value in the 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. **(30 marks)**

	Correct Sequence	**Pick From Here**	
Line 1		1	mov eax, 0
Line 2	2	2	mov ebx, 1
Line 3	3	3	mov ecx, 100
Line 4		4	mov eax, 1
Line 5	5	5	jz L2
Line 6		6	L1: and ebx, 00H
Line 7		7	L1: and ebx, 01H
Line 8		8	L2 : inc eax
		9	L2 : inc ebx
		10	L1: test ebx, 00H
		11	L1: test ebx, 01H
		12	loop L1
		13	add eax, ebx
		14	loop L2

28. Supply correct arguments and/or instructions to the missing places **(i)** to **(x)** in the following bubble sort program segment that compares two integers and swap them, resulting in a sorted list of integers in ascending order. **(Total 10 marks: each worth 1 mark)**

```
1    int MAX_SZ = 4
2    int intArray[(i)] = {6, 1, 8, 4};
3    (ii){
4      (iii) esi, (iv)     ; load first item's memory address
5      mov   eax, (v)      ; move content of memory address
6      (vi)    [esi+(vii)], eax ; compare adjacent items
7      (viii)  L3          ; jump out to L3
8      (ix) eax, [esi+4]   ; otherwise swap contents
9      (x) [esi], eax  ; continue swapping
10   }
```

28. (20 marks) Supply correct arguments and instructions to the missing places **(i)** to **(x)** in the following program segment that ensures the user enter a number greater than 10 otherwise the program requests the user to input again.

```
1    int number;
2    (i) format[] = "%d";
3    _asm{
4    (ii):
5      (iii) eax, number ; load number's memory address
6      push eax          ; push onto stack
7      lea   ebx, (iv)
8      push  (v)         ; push onto stack
9      call  (vi)        ; read user input
10     add   esp, (vii)    ; return stack pointer
11     mov   eax, (viii)   ; move user input into eax
12     cmp   eax, (ix)     ; compare with set condition
13     (x) readNumber  ; jump if condition not met
14   }
```