

APPLICATION

```
.data
myArray BYTE 1,2,5,7,10,11,15

.code
main PROC
    mov ecx, LENGTHOF myArray ; Set the number for LOOP executions
    xor ebx, ebx               ; ebx =0
    xor esi, esi               ; esi =0
L1:
    movzx eax, myArray[esi]
    and eax, 1                  ; if eax is 0, myArrayp[esi] is an even number
                                ; if eax is 1, myArrayp[esi] is an odd number
    add ebx, eax
L2:
    inc esi
    loop L1

    exit
main ENDP
END main
```



YOUR TURN

- Assume that the 0th row shows the value of registers when the program executes to L1, and fills the value of the registers in the i-th row of the table when the “loop L1” is executed to the L2.

	EAX	EBX
0	00000000h	00000000h
1		
2		
3		
4		
5		
6		
7		



```

.data
myArray BYTE 13,2,4,11,5,8
.code
Main PROC
    mov ecx, LENGTHOF myArray
    xor eax, eax
    xor ebx, ebx
    xor edx, edx
    xor esi, esi
L1:
    movzx eax, myArray[esi]
    test eax, 1                                ;Check if the last bit of eax is 1

    _____                                ;If the last bit of eax is equals to 1,
                                                ;jump to L2

    add ebx, eax                                ;Calculate the sum of even number

    _____                                ; jump to L3
                                                ; Avoid Calculate the sum of odd numbers

L2: add edx, eax                                ;Calculate the sum of odd number
L3: inc esi
    loop L1

exit
Main ENDP
END main

```



YOUR TURN

- Assume that the 0th row shows the value of registers when the program executes to L1, and fills the value of the registers in the i-th row of the table when the “loop L1” is executed to the L3.

	EAX	ZF	EBX	EDX
0	00000000h	0	00000000h	00000000h
1				
2				
3				
4				

