Question 1.

Consider the following arrays -

myarray1 DWORD 11h,22h, 33h
myarray2 DWORD 77h, 88h, 99h

Use XCHG instructions to exchange item between these two array. After the exchanges the arrays look like -

myarray1 : 88h, 99h, 77h
myarray2: 22h, 33h, 11h

You can only use EAX register as a temporary storage.

Answer: It can be solved in many different ways. Here is one of the solution.

```
MOV EAX, [myarray1+0]      ; EAX = 00000011
XCHG EAX, [myarray2+8]     ; EAX = 00000099 , myarray2  : 77,88,11
XCHG EAX, [myarray1+4]     ; EAX = 00000022,  myarray1 : 11,99,33
XCHG EAX, [myarray2+0]      ; EAX = 00000077, myarray2 : 22,88,11
XCHG EAX, [myarray1+8]     ; EAX = 00000033, myarray1 : 11,99,77
XCHG EAX, [myarray2+4]     ; EAX = 00000088, myarray2 : 22,33,11
XCHG EAX, [myarray1+0]      ; EAX =  00000011 myarray1  : 88,99,77
```

Things to check :
Temporary register must be EAX
MOV , XCHG can not take two variables as input
The first item in the array can be accessed by `arrayname+0` or `arrayname`
The second item in the array can be accessed by `arrayname+4` or `[arrayname+4]`
The third item in the array can be accessed by `arrayname+8` or `[arrayname+8]`
`Comment is not required.`

Question 2.

Assume that your computer is Little endian. Consider the following code-

```
.data
myarray WORD 1122h, 3344h, 5566h, 7788h
.code
MOV EAX, DWORD PTR myarray+3
```

What is the content of EAX register after executing the code? Explain your answer.

Answer:

1. Take each item and split them into bytes. For example, item 1 is 1122.

11	22

In little endian, store them in reverse order.

22	11

Follow the first step, we can get the offset of each byte in memory:

Data	22	11	44	33	66	55	88	77
offset	0	1	2	3	4	5	6	7

Myarray+3 take you to the offset 3, and DWORD PTR extracts 4 bytes from there (shown in red)

Data	22	11	44	33	66	55	88	77
offset	0	1	2	3	4	5	6	7

In little endian you need to read the bytes in reverse order as well

Now, read these 4 bytes in reverse order from offset 3: 88 55 66 33

EAX contains 88 55 66 33