Lab # Chapter 6 — Assessment Worksheet

**Course Name and Number: IAM302**

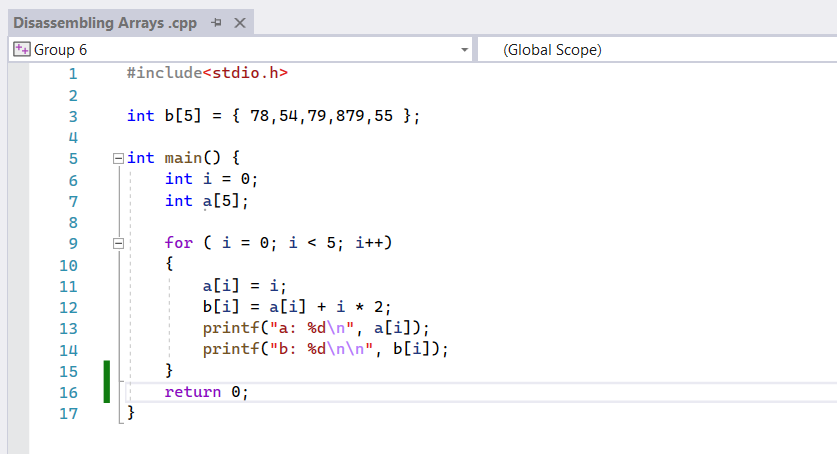
**Student Name and Student Id:**

* **Tran Thanh Tuan – SE161095**
* **Diep Anh Vu – SE160365**
* **Shim Jun Woo – SE161064**

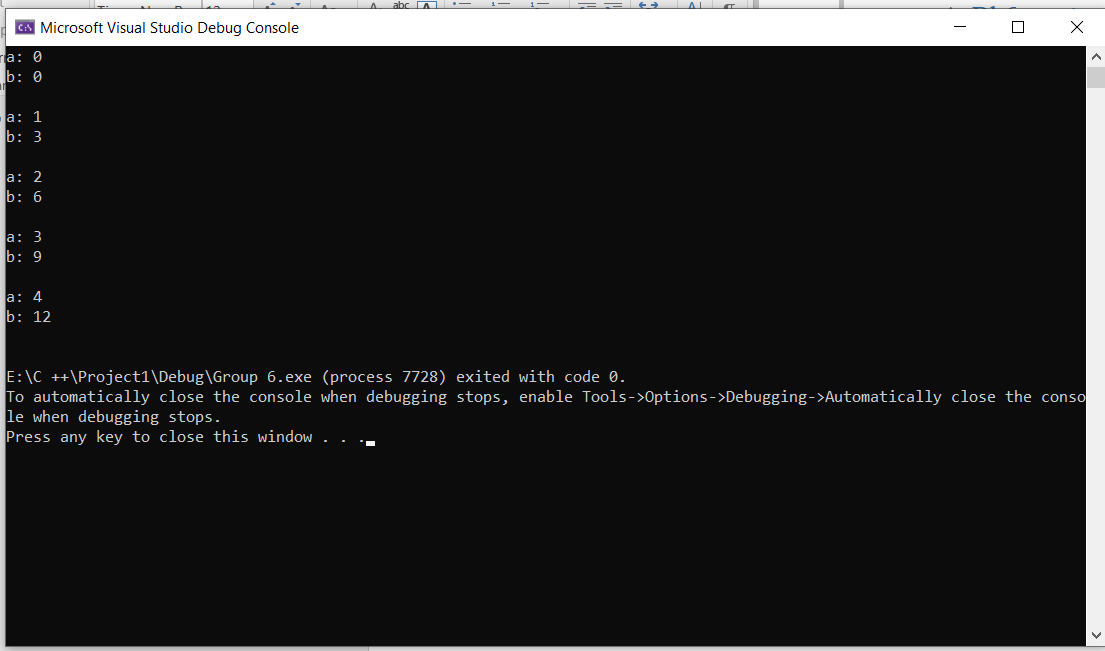
**Instructor Name: Vu Duc Ly**

**Recognizing C Code Constructs In Assembly**

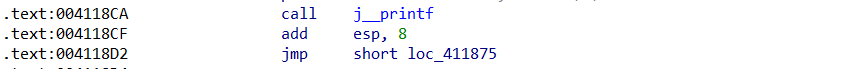
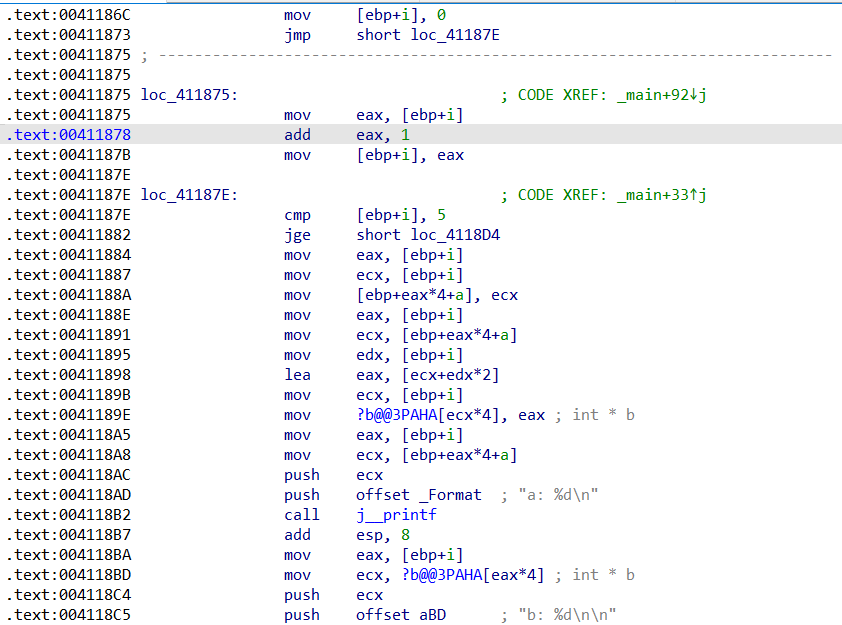
***C code***



***Result***

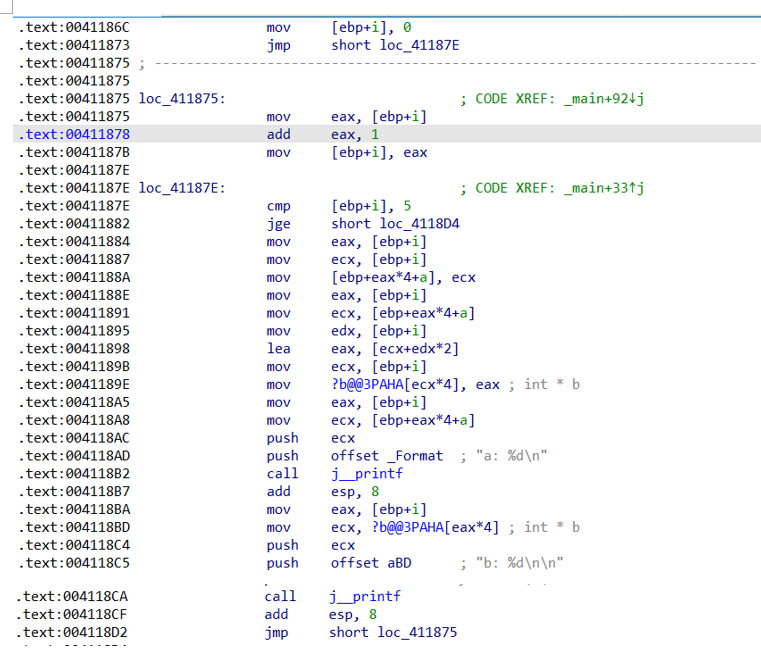


***Assembly Code by IDA***



***Explaining***

There are 6 steps: Initialization, Increment, Comparison, Assign value to Element in a, Assign value to Element in b and Execution.

****

Assign value to Element in b (base is ?b**@@**3PAHA)

Assign value to Element in a (base is a)

Comparison

Initialization

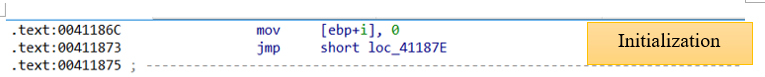
Execution

Increment

1. **Initialization (i starts at 0)**

In the main function, the first step of the for command line will be to assign the value i = 0. Since the variable i is a local variable, it will be [ebp + i] and on stack.

Then it will be a jmp command (Unconditional jump command) to loc\_41187E.



1. **Comparison (Is i smaller than 5)**

At loc\_41187E, the cmp statement compares i ([ebp + i]) with 5.

If i is greater than or equal to 5 execute statement jpe (Jump if greater than or equal) will jump to loc\_4118D4 and terminate.

If i is less than 5, the next command line will be executed after jge.



1. **Assign value to Element in a (base is a)**

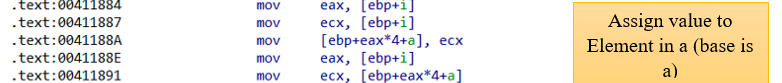
In this listing, the base address of array a corresponds to a.

The statement first copies the value of i ([ebp + i ]) into ecx and eax.

Then it will copy the value of ecx into a ([ebp + eax\*4 + a]).

In this cases, eax is used as the index, which is multiplied by 4 to account for the size of the elements. The resulting value is added to the base address of the array to access the proper array element.

Next the statement will reassign the value of [ebq + i] to eax and will reassign the value of a ([ebp + eax\*4 +a]) to ecx to prepare for the step of assigning the value to b.



1. **Assign value to Element in b (base is ?b@@3PAHA)**

In this listing, the base address of array b corresponds to ?b@@3PAHA.

The statement first copies the value of i ([ebp + i ]) into edx and eax.

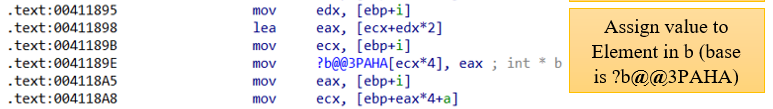
The lea statement then converts the offset address of the operation a[i] + i\*2 ([ecx + edx\*2]) into eax.

The next statement copies the value of i ([ebp + i ]) into ecx.

Then it will copy the value of eax into b (?b@@3PAHA[ecx\*4 ]).

In this cases, ecx is used as the index, which is multiplied by 4 to account for the size of the elements. The resulting value is added to the base address of the array to access the proper array element.

Finally the statement will reassign the value of [ebq + i] to eax and will reassign the value of a ([ebp + eax\*4 +a]) to ecx to prepare for the step of Execution.



1. **Execution (printf command)**

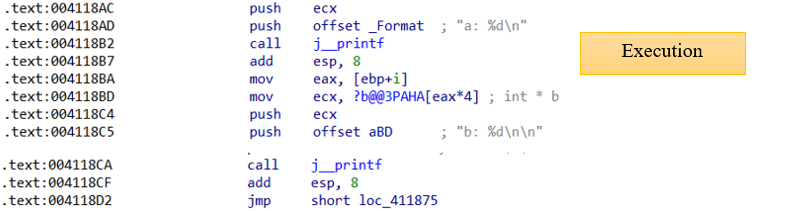
First, the push command will push the value of ecx which contains the value of a onto the stack.

Next push will push the format onto the stack.

Finally is the call to call printf in FILO order.

Similarly, according to b, we also get the printf command in FILO order.

Next jmp command will jump to loc\_411875.



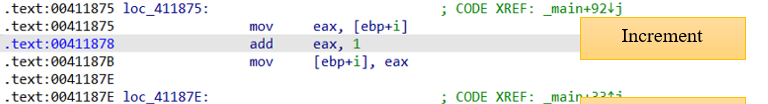
1. **Increment (i++ command)**

At loc\_411875, the statement first assigns the value of i ([ebp +i]) to eax.

Then the add command will add the values of eax and 1 together and save it to eax.

Finally, the statement will copy the value of eax to i ([ebp + i]) and then run loc\_41187E.

In summary, the above statements only terminate when the jge command is fulfilled (when i is greater than or equal to 5).

****