XOR Swap

Visual Reverse Engineering
Part 1

C code

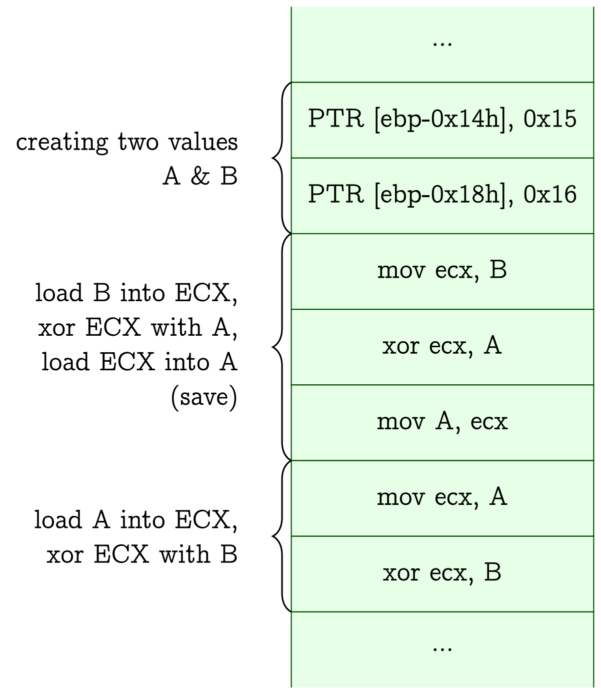
get better syntax highlighting for C & ASM

```c
int main(){

  int a = 5;
  int b = 6;

  a = b ^ a;
  b = a ^ b;
  a = b ^ a;

}
```

GDB dump of code in memory (main function)

```
Dump of assembler code for function main:
   0x100000f80 <+0>:    push   ebp
   0x100000f81 <+1>:    mov    ebp,esp
   0x100000f84 <+4>:    xor    eax,eax
   0x100000f86 <+6>:    mov    DWORD PTR [ebp-0x4],0x5    %set A = 5
   0x100000f8d <+13>:   mov    DWORD PTR [ebp-0x8],0x6    %set B = 6
   0x100000f94 <+20>:   mov    ecx,DWORD PTR [ebp-0x8]    %mov ecx, B
   0x100000f97 <+23>:   xor    ecx,DWORD PTR [ebp-0x4]    %xor ecx, A
   0x100000f9a <+26>:   mov    DWORD PTR [ebp-0x4],ecx    %store A, ecx
   0x100000f9d <+29>:   mov    ecx,DWORD PTR [ebp-0x4]    %mov ecx, A
   0x100000fa0 <+32>:   xor    ecx,DWORD PTR [ebp-0x8]    %xor ecx, B
   0x100000fa3 <+35>:   mov    DWORD PTR [ebp-0x8],ecx    %store B, ecx
   0x100000fa6 <+38>:   mov    ecx,DWORD PTR [ebp-0x8]    %mov ecx, B
   0x100000fa9 <+41>:   xor    ecx,DWORD PTR [ebp-0x4]    %xor ecx, A
   0x100000fac <+44>:   mov    DWORD PTR [ebp-0x4],ecx    %mov A, ecx
   0x100000faf <+47>:   pop    ebp                        %clear stack
   0x100000fb0 <+48>:   ret                               %exit
```

stack


...
creating two values A & B
PTR [ebp-0x14h], 0x15
PTR [ebp-0x18h], 0x16
load B into ECX, xor ECX with A, load ECX into A (save)
mov ecx, B
xor ecx, A
mov A, ecx
load A into ECX, xor ECX with B
mov ecx, A
xor ecx, B
...

explanation

Here we can see that we are loading our values A & B onto the stack at the locations [EBP-4] and [EBP-8], respectively.

When we XOR (symbol $\oplus$) A & B we do the following operations. The cooresponding binary for A is 0101 and B is 0110.

$A \oplus B = 0011$	We take this result and use it
$B \oplus A = 0101$	See here that B is now A
$A \oplus B = 0110$	And A is now B.

We have successfully swapped two values without using any XCHG or MOV instructions. However, will this trick work for all values? Try using the values 13 & 12 for A & B respectively. If you can find a solution to this problem, let me know.

If you need help finding a solution, feel free to ask me. You can find my contact info in the description below. Thank you.

Notes