1. The program allocates memory at a virtual address of **0xca0000**:
   1. What is the size of this allocation?
      1. **160kb**
   2. What are the permissions?
      1. **Read and Write (RW)**
   3. What is the status (or type)? What does that mean?
      1. **Private: committed - It’s not shared + has physical memory backing it**
   4. What is the value written at the beginning of this allocation?
      1. **Deadcodef0f0f0fe**
   5. What would a call to *VirtualAlloc* look like to make this memory allocation?
      1. **VirtualAlloc( Null | 0xca0000, 160000, MEM\_COMMIT, PAGE\_READWRITE);**
2. The program allocates memory at a virtual address of **0xab0000:**
   1. What are the permissions? How can a program use that differently than the previous allocation?
      1. **It can read write and execute, so it can fill the memory with code and run it**
   2. Why can you not inspect the content of this memory allocation?
      1. **It is only reserved, not committed, meaning that it has not been assigned any physical memory, and that there is nothing to see in that space.**
   3. What would a call to *VirtualAlloc* look like to make this memory allocation?
      1. **VirtualAlloc( Null | 0xab0000, 320000, MEM\_RESERVE, PAGE\_EXECUTE\_READWRITE);**
3. There is another allocation that has RWX permissions:
   1. What does it appear that this allocation is used for?
      1. **0xe000, signature: 4d5a = .exe executable file (also has a .text string and a .data string similar to assembly section headers), probably the instruction set being run.**
      2. **0x76e30000 -> reserved (cant view since it’s not backed)**
      3. **0x76f50000 -> reserved (cant view since it’s not backed)**
4. What handles does this program have open? Describe the importance of each one (or speculate if it’s not clear why the program has that handle).
   * 1. **(**type) Directory
        1. \KnownDlls
           1. **Since all these are DLL’s I’m thinking this might handle the dynamic linking of libraries when ran.**
        2. \KnownDlls32
           1. **Since all these are DLL’s I’m thinking this might handle the dynamic linking of libraries when ran.**
        3. \KnownDlls32 (again?)
           1. **Since all these are DLL’s I’m thinking this might handle the dynamic linking of libraries when ran.**
     2. (type) File
        1. C:\Windows
           1. **It’s probably using a bunch of stuff beneath \Windows**
        2. C:\Windows\SysWOW64\ntdll.dll
           1. **Google Describes it as :  
              “Windows(32bit) On Windows 64”… It’s something to interact with the OS**
        3. C:\Users\User\Desktop
           1. **Home directory of executable**
        4. C:\ Users\User\Desktop\process\_virtual\_memory.exe
           1. **Exact executable path**
     3. (type) Key
        1. HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Versions
           1. **Gander a guess: no idea, control … SYSTEM … HKLM … versions – maybe it has something to do with process startup / system interactions**
        2. HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
           1. **Dealing with** (**HK**)**EY\_**(**L**)**OCAL\_**(**M**)**ACHINE which has configuration information for majority of software on a windows system**
        3. HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options ***(Again?)***
           1. **Dealing with** (**HK**)**EY\_**(**L**)**OCAL\_**(**M**)**ACHINE which has configuration information for majority of software on a windows system**

<https://msdn.microsoft.com/en-us/library/windows/desktop/aa366887(v=vs.85).aspx>  
<https://msdn.microsoft.com/en-us/library/windows/desktop/aa366803(v=vs.85).aspx>  
<https://msdn.microsoft.com/en-us/library/ms810613.aspx>  
<https://msdn.microsoft.com/en-us/library/ms810627.aspx>  
<https://techtalk.intersec.com/2013/07/memory-part-2-understanding-process-memory/>

<https://techtalk.intersec.com/2013/07/memory-part-1-memory-types/>

<https://github.com/goldshtn/Memory>

This one is a pretty concise explanation of different memory types + has a linked github repository (cpp) for examples of different types of allocated memory:

<http://blogs.microsoft.co.il/sasha/2016/01/05/windows-process-memory-usage-demystified/>

Memory Protection Constants:  
 <https://msdn.microsoft.com/en-us/library/windows/desktop/aa366786(v=vs.85).aspx>