5. Consider a 32-bit virtual memory space. Suppose the page size is set to 64KB, and part of the page table entries is given as below.

	Page Table	
Virtual Page Number	Valid	Physical Frame Number
0x0	1	0xE
0x1	0	0x2
0x2	1	0x20B
0x3	1	0xA2
0x4	1	0x6
0x5	0	0x30
0x6	1	0x725

1) Based on the above information, use binary arithmetic to translate the following virtual addresses to their corresponding physical addresses in 32-bit hexadecimal format.
   Virtual Address: 0x00000A96
   Virtual Address: 0x00036813
   You must show the steps of your address translation.

2) If the page size is set to 16KB, using the same page table entries above, translate the following virtual address to the physical address in hexadecimal format (if the translation is not possible, say not possible with the given page table mapping)
   Virtual Address: 0x00008715
   You must show the steps of your address translation.

3) Suppose the physical memory space has 2GB memory, using the inverted page table scheme with the page size as 8KB, and each page table entry has 4 bytes, what would be the total bytes needed for storing the inverted page table, show your steps.

1) Page Size = 64KB = 2^{16} = 4 byte offset

Physical Address = Frame << Pagesize + offset (*)

Address #1 0000|0A96
VPN | Offset

PFN = E | E * 2^{16} = 000E0000 + 0A96 = $\boxed{0x000E0A96}$

Address #2 0003|6813
Vpn | Offset

PFN = A2 | A2 * 2^{16} = 00A20000 + 6813 = $\boxed{0x00A26813}$

2) Pagesize = 16KB = 2^{14}

Address = 0x00008715 = 0000|0000|0000|0000|1000|0111|0001|0101

VPN = 0x2 offset

Frame = 0x208 * 2^{14}

= 0000|0000|1000|0010|1100|0000|0000|0000 + offset

+ 0000|0000|0000|0000|0000 0|0111|0001|0101

= 0000|0000|1000|0010|1100|0111|0001|0101

= 0 0 8 2 C 7 1 5

= $\boxed{0x0082C715}$

3) 2GB of physical memory space = 2^{31}

Page Size = 8kb = 2^{13}

VPN: $2^{31} - 2^{13} = 2^{18}$

Each entry has 4 bytes = 2^2

$2^{18} \cdot 2^2 = 2^{20}$ = $\boxed{\text{1MB of bytes needed}}$