Suppose we have a system with a page size of 1K and a 4K virtual address space. The physical memory size is 16K.

A single load instruction is stored at virtual address 1024 (1K):

```
ld 2048, r1
```

This instruction generates two memory accesses: an instruction fetch from virtual address 1024 and a data load from virtual address 2048.

1. Draw the virtual address space. Include overall size (starting and ending addresses), load instruction and data (you can make up a value if you want to).
2. How many pages are in the virtual address space? Add those to your previous drawing.
3. How many bits are needed to encode each virtual address?
4. Show how to divide the virtual address into a virtual page number and offset.
5. How many bits are needed to encode each physical address?
6. Now suppose the system has a (hardware managed) linear page table. Each page table entry (PTE) consists of 8 bits:

```
| valid | Read  | Write | eXecute |  PFN  |
| 1 bit | 1 bit | 1 bit | 1 bit   | 4 bits|
```

The RWX bits are set to 1 if reading, writing, and execution respectively are allowed on a page.
The 4 entries in the page table are:
0xF5
0xCA
0xFF
0x00

(a) Find the physical address of the INSTRUCTION.

(b) Find the physical address of the DATA.