paging-linear-translate.py

In this homework, you will use a simple program, which is known as paging-linear-translate.py, to see if you understand how simple virtual-to-physical address translation works with linear page tables.

First, run the program without any arguments:

```
prompt> ./paging-linear-translate.py

ARG address space size 16k
ARG phys mem size 64k
ARG page size 4k
```

The above states the address space size (16KB), physical memory size (64KB), and page size (4KB).

```
It then shows a page table. The format of the page table is simple:
The high-order (left-most) bit is the VALID bit.
  If the bit is 1, the rest of the entry is the PFN.
  If the bit is 0, the page is not valid.
Use verbose mode (-v) if you want to print the VPN # by
each entry of the page table.

Page Table (from entry 0 down to the max size)
   0x8000000c
   0x00000000
   0x00000000
   0x80000006

Virtual Address Trace
  VA  0: 0x00003229 (decimal:    12841) --> PA or invalid?
  VA  1: 0x00001369 (decimal:     4969) --> PA or invalid?
  VA  2: 0x00001e80 (decimal:     7808) --> PA or invalid?
  VA  3: 0x00002556 (decimal:     9558) --> PA or invalid?
  VA  4: 0x00003a1e (decimal:    14878) --> PA or invalid?

For each virtual address, write down the physical address it
translates to OR write down that it is an out-of-bounds
address (e.g., a segmentation fault).
```