relocation.py

This program allows you to see how address translations are performed in a system with base and bounds registers. As before, there are two steps to running the program to test out your understanding of base and bounds. First, run without the -c flag to generate a set of translations and see if you can correctly perform the address translations yourself. Then, when done, run with the -c flag to check your answers.

In this homework, we will assume a slightly different address space than our canonical one with a heap and stack at opposite ends of the space. Rather, we will assume that the address space has a code section, then a fixed-sized (small) stack, and a heap that grows downward right after, looking something like you see in the Figure below. In this configuration, there is only one direction of growth, towards higher regions of the address space.

```
  -------------- 0KB
  |    Code    |
  -------------- 2KB
  |   Stack    |
  -------------- 4KB
  |    Heap    |
  |     |      |
  |     v      |
  -------------- 7KB
  |   (free)   |
  |     ...    |
```

In the figure, the bounds register would be set to 7~KB, as that represents the end of the address space. References to any address within the bounds would be considered legal; references above this value are out of bounds and thus the hardware would raise an exception.

To run with the default flags, type relocation.py at the command line. The result should be something like this:

```
prompt> ./relocation.py
...
Base-and-Bounds register information:

  Base   : 0x00003082 (decimal 12418)
  Limit  : 472

Virtual Address Trace
  VA  0: 0x01ae (decimal:430) -> PA or violation?
  VA  1: 0x0109 (decimal:265) -> PA or violation?
  VA  2: 0x020b (decimal:523) -> PA or violation?
  VA  3: 0x019e (decimal:414) -> PA or violation?
  VA  4: 0x0322 (decimal:802) -> PA or violation?
```

For each virtual address, either write down the physical address it translates to OR write down that it is an out-of-bounds address (a segmentation violation).
For this problem, assume a simple virtual address space of a given size.

As you can see, the homework simply generates randomized virtual addresses. For each, you should determine whether it is in bounds, and if so, determine to which physical address it translates. Running with -c (the "compute this for me" flag) gives us the results of these translations, i.e., whether they are valid or not, and if valid, the resulting physical addresses.