Assumptions:

- The page size is an unrealistically-small 32 bytes
- The virtual address space for the process in question is 1024 pages, or 32 KB
- Physical memory consists of 128 32-byte pages

Thus, a virtual address needs 15 bits (5 for the offset, 10 for the VPN).
A physical address requires 12 bits (5 offset, 7 for the PFN).

The system assumes a multi-level page table. Thus, the upper five bits of a virtual address are used to index into a page directory; the page directory entry (PDE), if valid, points to a page of the page table. Each page table page holds 32 page-table entries (PTEs). Each PTE, if valid, holds the desired translation (physical frame number, or PFN) of the virtual page in question.

The format of a PTE is thus:

```
VALID  |  PFN6  ...  PFN0
```

and is thus 8 bits or 1 byte.

The format of a PDE is essentially identical:

```
VALID  |  PT6  ...  PT0
```

The **Page Directory Base Register (PDBR)** is set to decimal **99**. (this means the page directory is held in this page)

The problem: **Translate the virtual address 0x60B8**. What value do you get back from memory when you fetch this virtual address?