## Exercise 1

Consider the page table shown in Figure 3.1 for a system with 12-bit virtual and physical addresses and with 256-byte pages. The list of free page frames is D, E, F (that is, D is at the head of the list, E is second, and F is last). Convert the following virtual addresses to

| Page | Page Frame |
|---|---|
| 0 | – |
| 1 | 2 |
| 2 | C |
| 3 | A |
| 4 | – |
| 5 | 4 |
| 6 | 3 |
| 7 | – |
| 8 | B |
| 9 | 0 |

their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory.)

- 9EF
- 111
- 700
- 0FF

## Answer

There are 2^12 addresses in the virtual addresses space.

Page size is given to be 256 bytes (2^8 addressed in a page).

So, the number of pages will be 2^12 / 2^8 = 2^4 = 16 pages

The number of needed bits to act as offset inside page:

log2(page size) = log2(256) = 8 bits for offset

Get the frame address in main memory just use the first 4 bits

| Virtual address | Physical address |
|---|---|
| 0x9EF | 0x0EF |
| 0x111 | 0x211 |
| 0x700 | 0xD00 |
| 0x0FF | 0xEFF |