Exercise 1

A computer with a 32-bit address uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-level page table field, and an offset. How large are the pages and how many are there in the address space?

Exercise 2

A computer has 32-bit virtual addresses and 4 KiB pages. The program and data together fit in the lowest page (0–4095). The stack fits in the highest page. How many entries are needed in the page table if traditional (one-level) paging is used? How many page table entries are needed for two-level paging, with 10 bits in each part?

Exercise 3

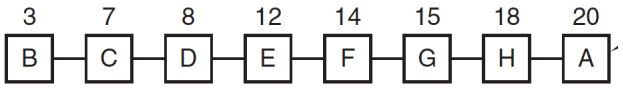
A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 5 nsec. To reduce this overhead, the computer has a TLB, which holds 32 (virtual page, physical page frame) pairs, and can do a lookup in 1 nsec. What hit rate is needed to reduce the mean overhead to 2 nsec?

Exercise 4

If FIFO page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string 0172327103 if the four frames are initially empty? Now repeat this problem for LRU.

Exercise 5

Consider the following page sequence:



(The numbers above the pages are their load times.) Suppose that the R bits for the pages B through A are 11011011, respectively. Which page will second chance remove?

Exercise 6

A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(a) Which page will NRU replace?

(b) Which page will FIFO replace?

(c) Which page will LRU replace?

(d) Which page will second chance replace?

Exercise 7

Consider the following two-dimensional array:

```
int X[64][64];
```

Suppose that a system has four page frames and each frame is 128 words (an integer occupies one word). Programs that manipulate the X array fit into exactly one page and always occupy page 0. The data are swapped in and out of the other three frames. The X array is stored in row-major order (i.e., X[0][1] follows X[0][0] in memory). Which of the two code fragments shown below will generate the lowest number of page faults? Explain and compute the total number of page faults.

Fragment A:

```
for (int j = 0; j < 64; j++)
    for (int i = 0; i < 64; i++)
        X[i][j] = 0;
```

Fragment B:

```
for (int i = 0; i < 64; i++)
    for (int j = 0; j < 64; j++)
        X[i][j] = 0;
```