4. [5 pt] Consider a disk with N tracks numbered from 0 to $(N-1)$ and assume that requested sectors are distributed randomly and evenly over the disk. We want to calculate the average number of tracks traversed by a seek. Calculate the probability of a seek of length j when the head is currently positioned over track t. Hint: This is a matter of determining the total number of combinations, recognizing that all the track positions for the destination of the seek are equally likely. A simple function for seek time T_s to traverse n tracks is $T_s = m \times n + s$ where s is the startup time (constant) and m is a constant depending on the disk drive.

5. Data in a system is organized by logical records, with 120 bytes to each record. We have a file of size approximately 523K.

   (a) [6 pt] Calculate how much disk space (in sectors, tracks, and surfaces) will be required to store the logical records if the disk is a fixed sector of 512 bytes per sector, with 96 sectors per track, 110 tracks per surface, and eight usable surfaces. Ignore any file header record(s) and track indexes, and assume that the records cannot span two sectors.

   (b) [6 pt] Assumes that the disk rotates at 360 rpm. A processor reads one sector from the disk using interrupt-driven I/O, with one interrupt per byte. If it takes 2.5 μsec to process each interrupt, what percentage of the time will the processor spend handling I/O (disregard seek time).

CS 376 **Operating Systems** Test 3
Name: Maximum Points: 44

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages.

1. Suppose the page table for the process currently executing on the processor currently looks like the following. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1024 bytes.

Virtual Page No.	Valid Bit	Reference Bit	Modify Bit	Page Frame No.
0	1	1	0	4
1	1	1	1	7
2	0	0	0	1
3	1	0	0	2
4	0	0	0	3
5	1	0	1	0

   (a) [4 pt] Describe exactly how, in general, a virtual address generated by the CPU is translated into a physical main memory address.

   (b) [9 pt] What physical address, if any, would each of the following virtual addresses correspond to? (Do not try to handle any page faults, if any.)

      i. 1052

      ii. 2221

      iii. 5499

2. [8 pt] A process references five pages A, B, C, D, and E, in the following order:

$$A;B;C;D;A;B;E;A;B;C;D;E$$

   Assume that the replacement algorithm is first-in, first-out and find the number of page transfers during this sequence of references starting with an empty main memory with 3 and 4 page frames. How do these cases compare with the optimal algorithm known to you?

3. [6 pt] Assume that we have a computer with a 3-address instruction for adding the contents of two addressed memory locations and depositing the sum in a third addressed memory location. In assembly form, the instruction looks like this:

```
ADD A, B, C                    /* (A) + (B) → (C) */
```

   If the instruction occupies three words and if all addresses are direct (i.e., the effective addresses are part of the instruction itself), what is the minimum number of page frames that this instruction requires to be guaranteed of proper execution. Explain your answer.