University of Toronto
CSC228F— File Structures and Data Management, Fall 1994

Midterm examination

Duration: 50 minutes

Aids allowed: One $8\frac{1}{2} \times 11$ inch page.

Make sure that your examination booklet has 5 pages (including this one).

Write your answers in the spaces provided. You will be rewarded for concise, well-thought-out answers, rather than long rambling ones. Write legibly.

Name: ______________________
(Circle your surname)

Student #: ______________________

Tutor: ______________________

1.	________	/ 8
2.	________	/ 4
3.	________	/ 4
4.	________	/ 8
5.	________	/ 6
Total	________	/ 30

1. (8 marks)
   Fill in the blanks appropriately in the following C program segments, so that in each case, the function will reset the variable as indicated in the comment.

   (a)
```
#include <stdio.h>
void resetInt (         n)
{
              = 0;
}

int main (void)
{
    int num = 938;
    /* Reset num to zero. */
    resetInt (         );
    return 0;
}
```

   (b)
```
#include <stdio.h>
typedef struct nodetyp {
    int             data;
    struct nodetyp *next;
    } Node;

void resetLinkedList (         p)
{
              = NULL;
}

int main (void)
{
    Node *front = (Node *) malloc(sizeof(Node));
    front -> data = 149;
    /* Reset the linked list to be empty. */
    resetLinkedList (         );
    return 0;
}
```

   (c) If you have a pointer to some data structure (for example, a linked list, as in (b) above) and you want to pass it to a function, how do you decide whether to pass the pointer itself, or a pointer to the pointer?

2. (4 marks)
   There are many ways arrange the storage of files on disk. If files are to be stored in equal-sized pieces, linked together, a size must be chosen for the pieces.

   (a) Describe a disadvantage that arises if the pieces are very small.

   (b) Describe a disadvantage that arises if the pieces are very large.

3. (4 marks)

   (a) Why would an operating system ever remove something from the cache?

   (b) There are many possible criteria that the operating system might use to decide *which* information in the cache to remove. Describe the overall goal that these criteria attempt to satisfy.

4. (8 marks)
Consider your program for assignment 2. The initial transaction file, the "calls log file", is sorted according to when the calls were made. However, the master-file update algorithm needs to access the calls in order by phone number (and secondarily, by when the calls were made). Below I describe four ways to deal with this problem. For each one, say whether or not it is a good approach, and explain why or why not.

(a) Create a linked list in memory that contains the information from the calls log file but is sorted in the order that the master-file update algorithm requires.

(b) Create a data structure in memory that contains not the call information itself, but file position indicators that point into the calls log file in such a way that if you traverse the data structure, the file pointers will direct you to the calls in the order required by the master-file update algorithm.

(c) Use a "sort-merge" method, as described in assignment 1, to sort the calls log file into the order required by the master-file update algorithm, and put the resulting sorted information into a new file.

(d) Do not reorganize the calls log file or create a reorganized copy. Instead, update the master file using a different algorithm; one that doesn't need the call information to be in any particular order.

5. (6 marks)
   With some file structures, we *compute* a record's location in a file from its key. There are many methods for doing this.

   (a) With direct mapping, we simply use the key itself as the record's location. This method can waste a great deal of space. There is one circumstance under which it does not. Explain.

   (b) Hashing is another method. With hashing, collisions can occur. What is a collision?

   (c) Imagine that you are designing an information system for student records at the University of Toronto, and 9 digit student numbers are the key by which operations are to be performed. Is it possible to design a hashing scheme in which collisions never occur? If it is possible, describe such a scheme; if it is not possible, explain why not.

END OF EXAM