UNIVERSITY OF TORONTO
Faculty of Arts and Science

April/May Examinations 1998

CSC228S

Duration — 3 hours

Aids allowed: None.

- **Make sure your examination booklet has 13 pages (including this one).**
- If you run out of space on any question, use the back of a page, and draw an arrow to point this out.
- You will be rewarded for concise, well-thought-out answers, rather than long rambling ones.
- Write legibly. Unreadable answers will be given 0.
- When asked to write code, comments are not necessary.
- If you need to call a standard C function but can't remember the correct order of arguments, just indicate the meaning of each argument.

Family Name: ______________________ Student #: ______________________

Given Names: ______________________

Tutor (circle one): H.S. Teoh Joe Rainsberger Darrell Grainger

Lidia Widjojo Brian Wherrett Oliver Fisher

1.	________	/ 7
2.	________	/ 12
3.	________	/ 15
4.	________	/ 6
5.	________	/ 12
6.	________	/ 16
7.	________	/ 15
8.	________	/ 17
Total	________	/ 100

Question 1

[7 marks in total]

a. [3 marks]
What would be the output from the program below?

```
#include <stdio.h>
int main (void) {
      int i;
      int a[5];
      int *p;
      for (i=0; i<5; i++)
            a[i] = i*i;
      p = a;
      printf ("P: %p %p %d\n", &p, p, *p);
      printf ("2: %p %p %d\n", &a[2], (a+2), *(a+2));
      return 0;
}
```

Assume that four consecutive memory locations are used to store a single integer, and that memory is allocated so that the variables are stored at the following memory locations:

Variable	Location in Memory
i	effffb2f
a	effffc02
p	effffb04

You may use this space to trace the code:

Output: (only this will be marked)

b. [4 marks]
The program below runs, but does not work. It fails to create a linked list with two nodes: one containing the integer 59, followed by one containing 11. Fix the bug by writing on the code below.

```
#include <malloc.h>


typedef struct Node {

     int data;

     struct Node *link;

} Node;


void insert (int value, Node *front) {

     Node *newOne;

     newOne = (Node *) malloc(sizeof(Node));

     newOne->data = value;

     newOne->link = front;

     front = newOne;

}


int main (void) {

     Node *myList = NULL;

     insert (11, myList);

     insert (59, myList);

     return 0;

}
```

Question 2

[12 marks in total]

Suppose we have a data file, in binary format, and that it was created using `fwrite` on an array of `StudentRec`s. `StudentRec` is some type of `struct`, whose details don't matter for this question. If we need to extract a student number out of a `StudentRec`, we can use function `studentNum`, whose prototype is:

```
int studentNum (StudentRec *s);
```

You are going to write a function that will read the data file and create, in memory, an index on that file. Although the data file won't fit in memory, the index will. The index should be by student number, and should *not* be sorted.

a. [2 marks]
First, give any `typedef`s that you'll need for declaring the index.

b. [10 marks]
Now write the function. Assume that the data file has already been opened successfully. Do not bother doing any sort of error checking. All information required by your function should be passed through arguments.

Question 3

[15 marks in total]

a. Linear hashing and linear probing solved two different problems. What are the problems?

Linear hashing:

Linear probing:

b. When does splitting occur with linear hashing? With extendible hashing?

Linear hashing:

Extensible hashing:

c. What problem is open addressing meant to solve?

d. Name three different schemes for open addressing.

1.

2.

3.

e. What is the difference between a collision and overflow?

f. What is seek time?

g. Databases have a "data dictionary". What does it define?

h. If we think of a database relation as a table, what do the project and select operations do in terms of the rows and columns of the table?

Project:

Select:

Question 4

[6 marks in total]

Consider a linear hashing scheme, where the initial hash function is $h(k) = k \bmod 17$.

a. How many buckets are there initially?

b. What is the first bucket to be split?

c. If we are splitting bucket p, to which buckets will its records go?

d. Once we have split all of the original buckets, how many buckets do we have in total?

e. If we need to split another bucket, which one will be split next?

f. What mod factor will be used when splitting that bucket?

Question 5

[12 marks in total]

a. [4 marks]
Consider insertion in a B-tree of order M. If insertion into a particular node causes that node to split, how many records will the two resulting nodes have, in terms of M?
ANSWER:

EXPLAIN EXACTLY HOW YOU ARRIVED AT YOUR ANSWER:

b. [4 marks]
Consider deletion of key k from a non-leaf node in a tree. The first step is to find k's "successor" — the smallest key in the tree that is greater than k. In an ordinary binary search tree, can the successor occur in a non-leaf node?

Yes No

In a B-tree, can the successor occur in a non-leaf node?

Yes No

Explain why or why not:

d. [4 marks]
Consider a B-tree of order M and height h. Fill in the following table to show the minimum and maximum number of nodes that can occur on a given level in the tree.

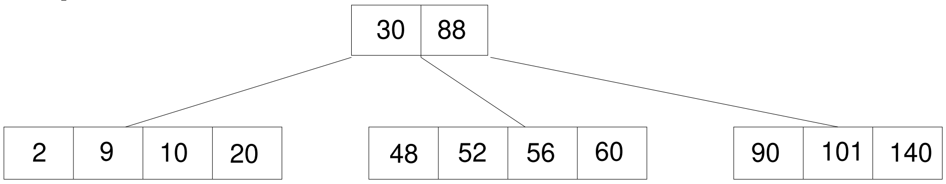
Level	Minimum number of nodes	Maximum number of nodes
1 (where the root is)		
2		
3		
k		

Question 6

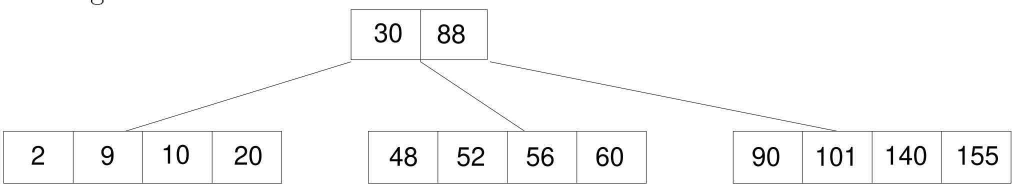
[16 marks in total]

For each part below, you will show the result of applying an operation to a B-tree. Each part is a separate question; they do not represent a sequence of operations applied to the same tree.

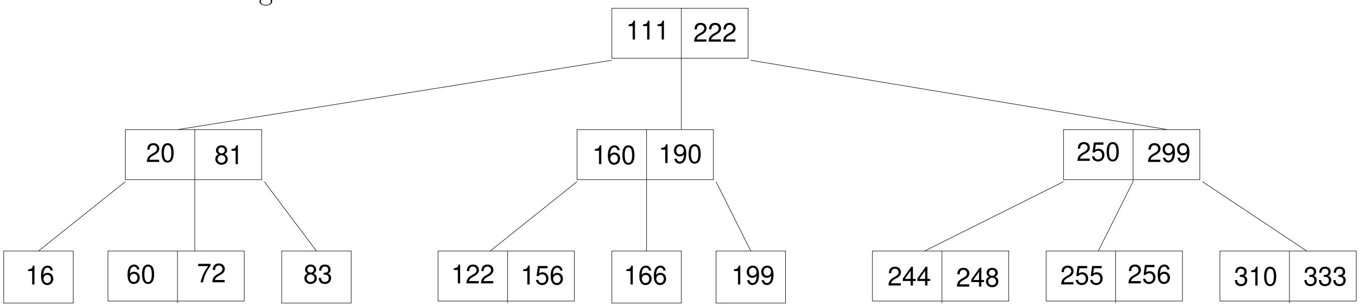
a. The following B-tree has order 5. Show what the tree would look like if 49 were inserted.



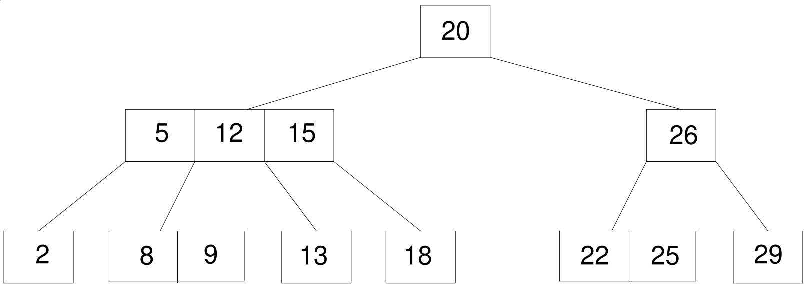
b. The following B-tree has order 5. Show what the tree would look like if 55 were inserted.



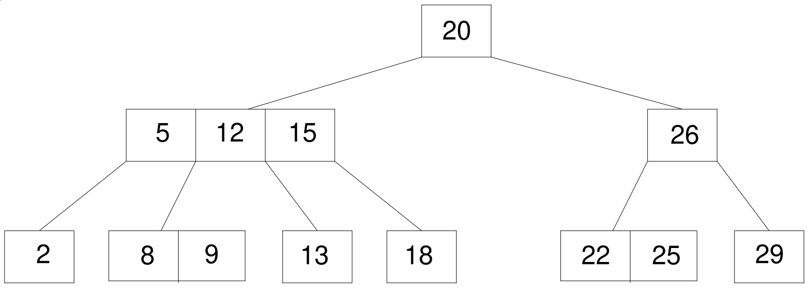
c. The following B-tree has order 3. Show what the tree would look like if 233 were inserted.



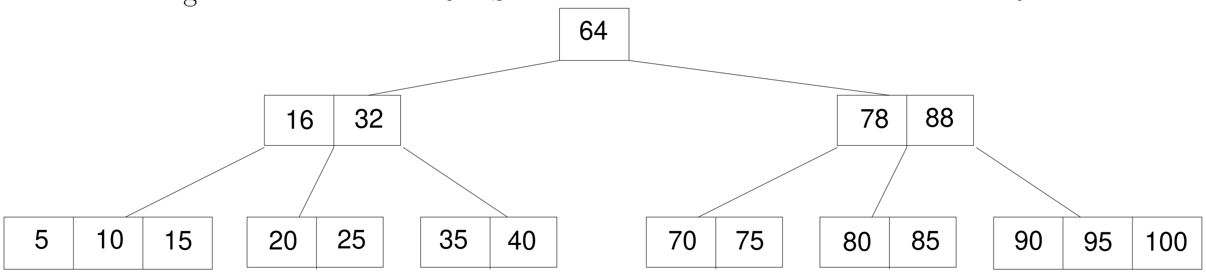
d. The following B-tree has order 4. Show what the tree would look like if 18 were deleted.



e. The following B-tree has order 4. Show what the tree would look like if 29 were deleted.



f. The following B-tree has order 5. Show what the tree would look like if 64 were deleted.



Question 7

[15 marks in total]

For each design idea below, indicate whether it is impossible to implement the idea, it is possible but inefficient, or it is a perfectly fine idea. Then explain why. Do not bother guessing; answers without an explanations will receive no marks

a. Using chaining for overflow, keeping the overflow buckets at the end of the file, and using linear hashing. IMPOSSIBLE POSSIBLE BUT INEFFICIENT A FINE IDEA

EXPLAIN:

b. Performing binary search on a file of variable length records, sorted by key.

IMPOSSIBLE POSSIBLE BUT INEFFICIENT A FINE IDEA

EXPLAIN:

c. Preventing overflow simply by making bigger any bucket that is going to overflow.

IMPOSSIBLE POSSIBLE BUT INEFFICIENT A FINE IDEA

EXPLAIN:

d. Using a B-tree to access data by one unique key, with the actual data records stored in the B-tree nodes, and using hashing to gain access by a second unique key.

IMPOSSIBLE POSSIBLE BUT INEFFICIENT A FINE IDEA

EXPLAIN:

e. Using a hash function that takes a student's name (the key) and produces a 6-digit integer by concatenating 3 pairs of digits, each representing one of the last 3 letters in the name. For example, "Horton" would be hashed to the integer 201514 ('t' is the 20th letter of the alphabet, 'o' is the 15th, and 'n' is the 14th).

IMPOSSIBLE POSSIBLE BUT INEFFICIENT A FINE IDEA

EXPLAIN:

Question 8

[17 marks in total]

Suppose the university has a relational database to keep track of students, courses, etc. The database includes the relations below.

STUDENTS:

StudentNumber	Name	Address	PhoneNumber
651123	Fred Flintstone	59 Bedrock Blvd	231-2431
669124	Barney Rubble	57 Bedrock Blvd	231-7444

... *etc.*

COURSES:

Course	Title
csc434	Database Systems
csc228	File Structures

... *etc.*

OFFERINGS:

ID	Course	Term	Location	Time	Days
10612	csc228	97f	WB100	2	TR
10554	csc364	98s	SS2112	9	MF

... *etc.*

CREDITS:

StudentNumber	Offering	Mark
669124	10554	74
651123	10612	91

... *etc.*

For this question, you will write a series of queries on this database. You must use relational algebra.

a. Write a query to extract the location of all classes ever offered at 9:00.

b. Write a query to extract the csc228 mark of all students who ever took that course.

c. Write a query to extract the name of every student who took csc228 in the term 97f and got more than 90 in it.

d. Write a query to extract the phone number of all students who have taken csc209 and have a mark over 80 in csc378.

e. Write a query to extract the phone number of all students who ever got 100 in any course.

f. Write a query to extract the course title and mark Fred Flintstone earned for every course he has taken.

END OF EXAM; Total pages = (13)