University of Toronto
Department of Electrical and Computer Engineering

Final Examination

CSC467 Compilers and Interpreters
Fall Semester, 2001

Print your name and ID number neatly in the space provided below; print your name at the upper right corner of every page. Place a picture ID on your table for verification.

Name:	
ID Number:	

This booklet should be ten pages long including this page. If not, report this to the instructor or the TA. Do all problems in this booklet. Try not to spend too much time on any one question.

This is an open book exam. You are allowed to refer to the text and your notes. **Do all work in the space provided.** You can use the backs of sheets as scrap paper, if necessary. Ask the proctor if you need more paper. **Nothing in the back of the sheets will be graded.**

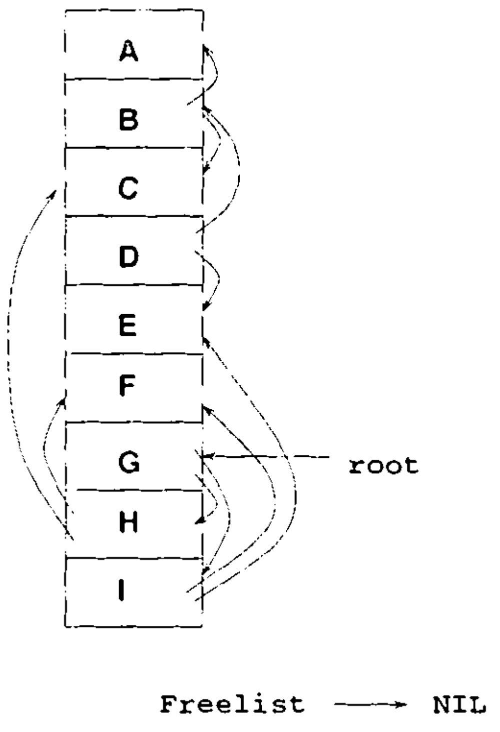
HAPPY HOLIDAY SEASON!

Question	Points	Score	Grader
1	8		
2	8		
3	10		
4	10		
5	12		
6	14		
7	8		
8	15		
9	15		
Total	100		

1. (2+2+2+2 points) In the following program, what does g() print under (a) call-by-value and lexical scope, (b) call-by-value and dynamic scope, (c) call-by-reference and lexical scope, and (d) call-by-reference and dynamic scope?

```
function g() {
  int a = 2;
  void  f(int b) {
     b = b * a;
     a = a - b;
  }
  {
     int a = 10;
     f(a);
     print a;
  }
}
```

2. (4+4 points) Indicate which cells will be marked and which cells will be in the free list after Mark and Sweep garbage collects the following piece of heap memory:



Marked nodes:

Freelist nodes:

3. (10 points) Consider the following fragment of intermediate code:

```
y = w
z = 4
v = y * y
u = z + 2
r = w ** 2  /* exponentiation */
t = r + v
s = u * t
```

Assume that only variable s is live on exit from this fragment. Show the result of applying as much constant propagation, algebraic simplification, common sub-expression elimination, constant folding, copy propagation (see last pages of lecture notes or textbook p. 594) and dead code elimination as possible to this code. To receive full credit you will need to show the type of optimization you perform, the actual result on the intermediate code instruction and also show the optimization steps in the order performed. You need not show the entire code sequence after every optimization, but you should explain clearly the changes at every step.

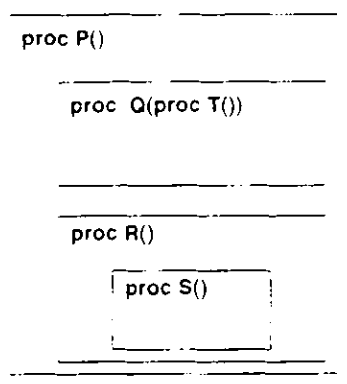
4. (10 points) Grades for students in CSC467 are maintained in a file organized as follows:

- The file contains one or more student records.
- A student record consists of a last name, then a comma, then a first name, then a list of zero or more lab grades separated by commas.
- A grade is an integer followed by zero or more stars. Each star represents one late lab day.

Write a context-free grammar that could be used to parse the CSC467 grade file. Use lower-case words for non terminals and the upper-case words given below for terminals (tokens):

`NAME`	`a sequence of one or more letters`
`COMMA`	`a comma`
`INT`	`an integer`
`STAR`	`a star`

5. (12 points) Consider a program written in a lexically scoped language (such as Pascal) with nested procedures that has the following lexical structure:



P, R and S are parameterless procedures. Q takes a parameterless procedure T as a parameter. Suppose that at run-time the following sequence of calls is made:

P is called from the lexically enclosing `main` program
P calls R
R calls S
S calls P
P calls R
R calls Q with S as a parameter
Q calls T
T calls S

Draw the stack of activation records present after this sequence of calls. You don't need to show the entire contents of an activation record. You just need to indicate the name of the procedure being activated (with the correct index, if necessary), its control link (on the left of the boxes) and its access link (on the right of the boxes).

6. (7+7 points) Consider the following simple model for estimating the cost of garbage collection (GC) for a heap of size n. We divide running time into time spent actually running the user's program (which is useful work) and time spent in GC (which is overhead):

- The number of cycles required for a single GC is equal to the number of different memory words that GC references. That is, even if a GC touches a particular word of memory more than once, we only charge 1 cycle.
- The *survival rate* s is the size of the live data after a GC, expressed as a fraction of the total heap size. For example, if garbage collecting a 10MB heap leaves 1MB of live data then the survival rate is 0.1.
- A user program allocates 1 word of heap for every 10 cycles of time the user program is running (*i.e.*, not in a GC).

Explain and give a formula in terms of s and n for the fraction of total execution time spent in GC for Mark and Sweep. Assume the program runs for a long time (steady state performance).

Explain and give a formula in terms of s and n for the fraction of total execution time spent in GC for Semi-Space. Again, consider steady state performance.

7. (8 points) Given the following three-address code, show the basic blocks and draw the flow graph. You don't need to rewrite every statement: use the numbers to identify the statements in each basic block:

```
(1)     b = c
(2)     i = 0
(3)     t = 4 * i
(4)     u = a[t]
(5)     v = u * u
(6)     w = 4 * i
(7)     a[w] = v
(8)     if v>d goto 13
(9)     i = i + 1
(10)    if I < b goto 3
(11)    x = c + d
(12)    e = x
(13)    y = c + b
(14)    f = y
```

8. (15 points) Consider the following augmented grammar:

$$\begin{aligned} S' &\to S \\ S &\to X\ S\ X \\ S &\to c \\ X &\to a \\ X &\to b \end{aligned}$$

Give the canonical collection of sets of LR(0) items for this grammar.

9. (5+5+5 points) Give state diagrams of DFAs recognizing the following languages. In all cases, the alphabet is $\{0, 1\}$. Briefly explain the main concept(s) behind your DFA construction. The less states your (correct) DFA has the more credit you will get.

(a) $L = \{ w \mid w \text{ does not contain the substring } 001 \}$

(b) $L = \{ w \mid w \text{ contains } 11001 \text{ as a substring} \}$

(c) $L = \{ w \mid$ every block of four consecutive symbols contains substring 10 $\}$
Hint: Think Yahoo!