

CS 540 Language Processing and Runtime Environments

Fall 2003 Final Exam

Name _____

Question	Total Points	Your Score
1	12	
2	10	
3	20	
4	15	
5	14	
6	10	
7	8	
8	11	
TOTAL	100	

1. You will have until 10:30 hours to work on this exam.
2. This exam is to be your own work. Talking or looking at another student's answers is a violation of the honor code and will be treated as such.
3. Write clearly. If I cannot read your answer, it is wrong.

1. **Multiple Choice** (12 pts) For each question, circle **all** choices that apply.

a. How many tokens occur in the following Tiger expression:

```
while a < b do g() end
```

- a. 6
- b. 7
- c. 9
- d. 11

b. Although tokens could be recognized by the context-free parser, a lexical analyzer is normally used because:

- a. A context free automata is more powerful might recognize too large a set of string
- b. Finite state automata are simpler (hence faster)
- c. Overall design of a compiler is simpler that way
- d. LL(1) parsers are simpler than LALR(1) parsers

c. Which of the following are LL(1) grammars equivalent to:

$$W \rightarrow a \mid b \mid W [W] \mid W , W \&$$

- a. $W \rightarrow a \mid b \mid X$
 $X \rightarrow a [W] \mid b , W \&$
- b. $W \rightarrow W X$
 $X \rightarrow a \mid b \mid [W] \mid , W \&$
- c. $W \rightarrow a \mid b \mid W X$
 $X \rightarrow [W] \mid , W \&$
- d. $W \rightarrow a X \mid b X$
 $X \rightarrow [W] \mid , W \& X$
- e. $W \rightarrow a X \mid b X$
 $X \rightarrow \lambda \mid [W] X \mid , W \& X$

d. All of the following Tiger expressions are syntactically correct. Mark the ones that are semantically correct given the type system you implemented for assignment #3.

a. `while 7 do 8 end`

b. `printint(42)`

c. `let var a : int in
if a > 10 then a > 100 else a < 1 end
end`

d. `function f() = ("hello world")`

2. (10 pts) For each item in column one, choose the best match (a – j) from column two.

Match	Column 1	Column 2
	Call by reference	a. Arrays
	Constant propagation	b. Basic block
	Context free grammar	c. Bottom up evaluation
	Control flow graph	d. Dead code elimination
	DFA	e. Language syntax
	Induction variable	f. Parameter passing
	Live variable	g. Regular languages
	Row—major	h. Strength Reduction
	Static (access) link	i. Used before defined
	Synthesized attributes	j. Variable addressing

3. **Array Storage.** (20 pts) Suppose we have a three dimensional array stored at address 100 with dimensions $A[1..5, 2..11, -1..1]$ where the individual elements are 5 bytes in size. Answer the following for row—major storage. Show your equations for complete credit and so that you are not penalized for arithmetic errors.

a. What are the five array elements following element $A[1, 2, -1]$?

b. What is the size of the entire array?

c. What is the size (stride) of each dimension (left to right)?

d. At what memory location is element $A[4, 5, 0]$ stored?

e. What element is stored at location 750?

4. **Parameter passing.** (15 pts) Consider the program below. What 5 values get printed out if the parameters are passed:

- Call by value
- Call by reference
- Call by copy/restore (value/result)

```
A,B: Int;
```

```
new(B,C: Int) : Int is  
begin  
  C ← 2; B ← A + C;  
  putint(A); putint(B); putint(C)  
end;
```

```
main() : Int is begin  
  A ← 6, B ← 1; new(A,B); putint(A); putint(B)  
end;
```

5. **Errors** (14 pts) One important task of a compiler is to find errors in the source code. For each error below, identify the type of the error (i.e. at what point the error can be found) by labeling the error as **(A) syntactic**, **(B) static semantic** or **(C) dynamic semantic (runtime)**. Use A-C to indicate your choice. You can assume Tiger is the language we are considering.

- _____ Wrong number of arguments supplied to a call.
- _____ Division by zero.
- _____ Applying an operator to a variable of the wrong type.
- _____ Missing semicolon after a declaration
- _____ Unwanted infinite loop.
- _____ Unbalanced parentheses
- _____ Use of an uninitialized variable

6. (10 pts) Construct the dag for the following basic block

```
d := b * c
e := a + b
b := b * c
a := e - d
b := b + c
```

7. **Live Variable Analysis** (8 pts) Annotate the intermediate code below with the live set at each point, assuming the initial live set at the end

```
a := 1      {
c := a + 1  {
b := d + c  {
d := a * b  {
a := a + 1  {
c := a + 2  {
e := b + c  {
a := d + 1  {
              { a, e } ← live
```

8. (11 pts) For your tiger assignment, you assumed that parameters were passed by value. Explain how you implemented this style of parameter passing. Explain how you would change this implementation to allow call by value/result parameter passing.