$Id: cmps104a-2012q4-exam3.mm,v 1.50 2012-11-30 13:45:38-08 - - $

page 1	page 2	page 3	page 4	page 5	Total / 52

Please print clearly :

Name: SOLUTION

Login: @ucsc.edu

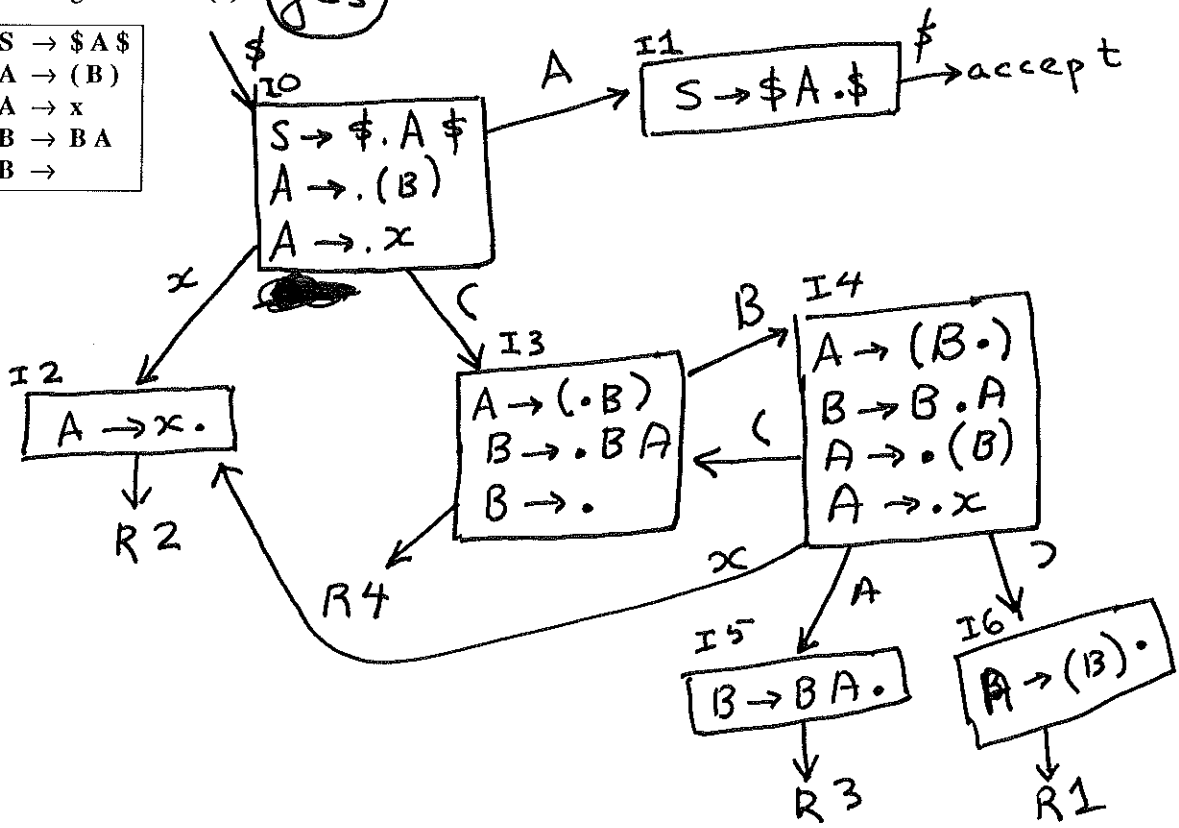
No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Write code for the function `malloc`. Assume that there is a single contiguous free area bounded by `firstfree` and `end`. Call a garbage collector if space is unavailable. Do not code the gc. Always return a chunk of memory which is a multiple of 16 bytes aligned on a 16-byte boundary. [4✓]

```
void *malloc (size_t n) {
    n = (n + 15) & ~15;
    if (firstfree + n > end) gc();
    void *ret = firstfree;
    firstfree += n;
    return ret;
}
```

2. Given the grammar presented here, and using the style from the LALR(1) handout : [6✓]
   (α) Construct the characteristic finite state machine (CFSM), sets of items and transition diagram, showing shifts, reductions, and acceptance.
   (β) Is the grammar LR(0) ? yes

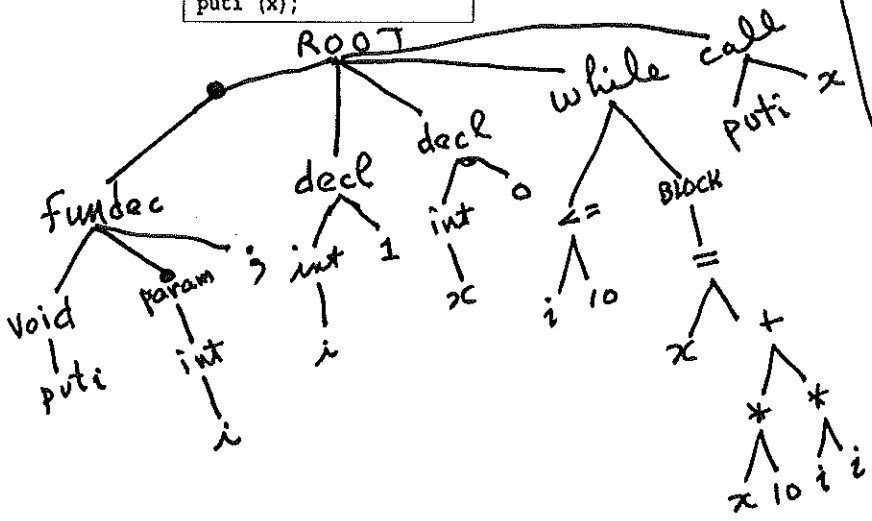
0.	S	→ $ A $
1.	A	→ (B)
2.	A	→ x
3.	B	→ B A
4.	B	→



3. Given the oc program presented here:
   (α) Draw the abstract syntax tree. [3✓]
   (β) Show the contents of the symbol table, with attributes. [3✓]
   (γ) Show the generated oil program. [4✓]

Note that there is no header file included in this program.

```
void puti (int i);
int i = 1;
int x = 0;
while (i <= 10) {
   x = x * 10 + i * i;
}
puti (x);
```



puti {0}. void fn {1}
i {1} param var lval int
i {0} var lval int
x {0} var lval int

```
void --puti (int _1_i);
int --i;
int --x;
void --ocmain () {
   --i = 1;
   --x = 0;
while_10_20: bool b1 = i <= 10;
   if (!b1) goto break_10_20;
   int i2 = --x * 10;
   int i3 = --i * --i;
   int i4 = i2 + i3;
   --x = i4;
   goto while_10_20;
break_10_20:
   --puti (--x);
}
```

4. Write a scanner and a parser using `flex` and `bison` for a Lisp-like language, as described here. Clearly identify which of your code belongs in `scanner.l` and which belongs in `parser.y`. Don't create tokens or assemble an abstract syntax tree. Semantic actions in the scanner will consist only of `return` statements when necessary. **[10✓]**
   - (α) A number is a sequence of one or more decimal digits.
     Example: `12345`
   - (β) A string is a sequence of zero or more characters, other than a double quote (") preceded and followed by double quotes.
     Example: `"foobar"`
   - (γ) An identifier is any sequence of one or more letters, digits, and underscores, unless it matches a number.
     Example: `123foo_bar`
   - (δ) An operator is any character from the set `+ - * / = < >`
   - (ε) White space consists of spaces, tabs, and newlines.
   - (ζ) Comments are Java-style, starting with a double slash (//) followed by all characters up to but not including the trailing newline.
   - (η) Any other character is invalid.
   - (θ) A program is a sequence of zero or more elements.
   - (ι) An element is an atom or a list.
   - (κ) An atom is a number, a string, an identifier, or an operator.
   - (λ) A list is a left parenthesis, followed by zero or more elements, followed by a right parenthesis.

```
%%
[0-9]+                  return NUM
\"[^"]*\"               return STR
[0-9_a-zA-Z]+           return ID
"+"                     return '+'
"-"                     return '-'
"*"                     return '*'
"/"                     return '/'
"="                     return '='
"<"                     return '<'
">"                     return '>'
[ \t\n]+                { }
"//".*                  { }
.                       error
```

```
%token NUM STR ID
%%
prog : prog elt
     |
     ;
elt : atom
    | list
    ;
atom : NUM
     | STR
     | ID
     | '+'
     | '-'
     | '*'
     | '/'
     | '='
     | '<'
     | '>'
     ;
list : '(' elts ')'
     ;
elts : elts elt
     |
     ;
```

Multiple choice. To the ***left*** of each question, write the letter that indicates your answer. Write **Z** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✔]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

1. Which statement is true about these languages ?
   (A) $LR(0) \subset LALR(1) \subset SLR(1) \subset LR(1)$
   (B) $LR(0) \subset LR(1) \subset SLR(1) \subset LALR(1)$
   (C) $LR(0) \subset SLR(1) \subset LALR(1) \subset LR(1)$
   (D) $LR(1) \subset LALR(1) \subset SLR(1) \subset LR(0)$

2. For a grammar $G = \langle V_N, V_T, P, S \rangle$, used to construct an LR(k) parse table with n states, what is the size of the parse table ?
   (A) $n \times |V_N|^k$
   (B) $n \times |V_T|^k$
   (C) $n \times |P|^k$
   (D) $n \times |S|^k$

3. If `malloc`(3) and `free`(3) use the boundary tag method to manage storage, what is likely the size of a single boundary tag ?
   (A) `8 * sizeof (char)`
   (B) `2 * sizeof (int)`
   (C) `2 * sizeof (struct node)`
   (D) `2 * sizeof (void *)`

4. To solve a shift/reduce conflict, we should shift if the precedence of the rule is [x] than the precedence of the lookahead symbol, or if the precedences are the same but they are [y] associative.
   (A) [x] = higher, [y] = left
   (B) [x] = higher, [y] = right
   (C) [x] = lower, [y] = left
   (D) [x] = lower, [y] = right

5. Which of the following grammars is unambiguous, recognizes an arbitrarily large number of symbols, but uses only $O(1)$ stack space ?
   (A) A → A A
       A → x
   (B) A → A x
       A → x
   (C) A → x A
       A → x
   (D) A → x x
       A → x

6. For a grammar $G = \langle V_N, V_T, P, S \rangle$:
   (A) $P \in V_N$
   (B) $P \in V_T$
   (C) $S \in V_N$
   (D) $S \in V_T$

7. Given the C global static variable declaration
   ```
   int a[10];
   ```
   which is an ***lvalue*** ?
   (A) `**a`
   (B) `a`
   (C) `a+2`
   (D) `a[2]`

8. What is permitted in an NFA but not a DFA ?
   (A) epsilon transitions
   (B) input alphabet
   (C) nonterminal symbols
   (D) undeclared identifiers

9. Given the following grammar, assuming appropriate declarations in section 1 of a `bison` grammar, which should fill in the blank to ensure proper (i.e. C or Java) parsing ?
   ```
   stmt : IF '(' expr ')' stmt ELSE stmt
        | IF '(' expr ')' stmt _____
        | other
        ;
   ```
   (A) `%left ELSE`
   (B) `%prec ELSE`
   (C) `%right ELSE`
   (D) `%token ELSE`

10. What might be an address returned by a successful call to `malloc` when run in the lab if printed with `%p` format ?
    (A) `0x0`
    (B) `0x1206010`
    (C) `0x1234567`
    (D) `0xdefghij`

11. The name `bison` is a pun on an earlier program whose name is a homonym for :
    (A) Buffalo : a kind of African stag or gazelle.
    (B) Camel : a ruminant used for carrying burdens and for riding.
    (C) Minotaur : a mythical monster in the labyrinth on Crete.
    (D) Yak : a bovine mammal native to the high plains of central Asia.

Multiple choice. To the *left* of each question, write the letter that indicates your answer. Write Z if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✔]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

1. A large number of allocated and unallocated chunks of memory randomly intermixed is called:
   (A) fragmentation
   (B) memory leak
   (C) storage regeneration
   (D) thrashing

A

2. A method of garbage collection that results in all reachable objects being compacted into as few pages as possible is:
   (A) copying with semispaces
   (B) incremental parallel threads
   (C) mark and sweep
   (D) reference counting

A

3. The weakest form of grammar that can detect matching parentheses is:
   (A) context free
   (B) context sensitive
   (C) regular
   (D) unrestricted rewriting

A

4. What system call can extend the data segment in order to increase the size of the heap?
   (A) `brk`
   (B) `execve`
   (C) `fork`
   (D) `malloc`

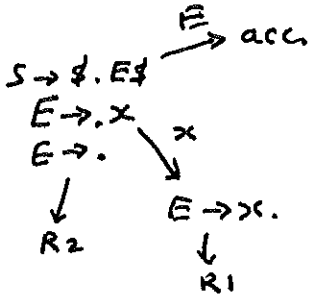
A

5. In a language allowing nested functions, what will allow an inner function to access the local data of an outer function?
   (A) dynamic link
   (B) return link
   (C) stack pointer
   (D) static link

D

6. Which grammar will recognize any number of left parentheses followed by any number of right parentheses, provided that there are the same number of each, and will fail to recognize anything else?
   (A) **B** → ( ) **B**
   B →
   (B) **B** → (**B**) **B**
   B →
   (C) **B** → (**B**)
   B →
   (D) **B** → (**B**
   B → **B**)
   B →

C



7. The following grammar:
   E → x
   E →
   (A) is both LR(0) and SLR(1).
   (B) is LR(0) but not SLR(1).
   (C) is neither LR(0) nor SLR(1).
   (D) is SLR(1) but not LR(0).

D

8. What part of a compiler will figure out if the expression `x + 3` is a valid expression?
   (A) lexical analyzer
   (B) parser — syntax valid
   (C) symbol table manager
   (D) type checker — opnd compat

B D

9. The largest class of grammars whose parse tables are the same size as an LR(0) parse table is:
   (A) LR(0)
   (B) SLR(1)
   (C) LALR(1)
   (D) LR(1)

C

10. The largest possible Unicode character, according to the current definition is:
    (A) `0x7F`
    (B) `0xFF`
    (C) `0x10FFFF`
    (D) `0xFFFFFFFF`

C

11. The _____book discusses compilers.
    (A) Camel
    (B) Dæmon
    (C) Dragon
    (D) Penguin

C