First Name: __________________________
Last Name: __________________________
PID (on your ID card): __________________________
MSU Net ID: __________________________

Exam for CSE 450 (Fall 2019)

Answer the questions in the spaces provided on the page. If you run out of room for an answer, continue on the back of the page.

• DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO
• You only need to answer 5 of the 6 questions.
• On one of the questions make a large slash through it, which indicates that the question should not be graded.
• If you start to answer a question and then change your mind, please cross out the attempt and write DO NOT GRADE across it.
• Have your ID ready when you turn in your exam.
Question 1: Arrays and Memory........................................40 points

Note: Both parts (a) and (b) read and write to the same memory at the bottom of the page.

(a) (25 points) For the memory representation below, write in the row “New Value” any values that would change from executing the **LMAO** code instruction. Memory location 0 stores the start of the free memory.

```
AR_COPY a4 a1
ADD s2 s3 s2
TEST_EQU s5 '\'' s9
```

(b) (15 points) For the same memory representation below, write in the row “New Value” any values that would change from executing the series of **ROFL** code instructions.

```
VAL_COPY 0 regB
MEM_COPY 11 13
LOAD 10 regA
label_1:
SUB regA 1 regA
JUMP_IF_0 regA label_2
ADD 1 regB regB
JUMP label_1
label_2:
STORE regA 14
STORE regB 15
```

<table>
<thead>
<tr>
<th>Location</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Value</td>
<td>22</td>
<td>16</td>
<td>7</td>
<td>1</td>
<td>17</td>
<td>'!'</td>
<td>5</td>
<td>'a'</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>New Value</td>
<td>25</td>
<td>22</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Value</td>
<td>2</td>
<td>67</td>
<td>'q'</td>
<td>-13</td>
<td>0</td>
<td>59</td>
<td>0</td>
<td>2</td>
<td>'c'</td>
<td>'a'</td>
</tr>
<tr>
<td>New Value</td>
<td></td>
<td>67</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Value</td>
<td>'t'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New Value</td>
<td>2</td>
<td>'c'</td>
<td>'a'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 2: LMAOcode to ROFLcode ........................................ 40 points
For each LMAOcode instruction below, convert to ROFLcode.

(a) (5 points) IN_CHAR s14

Solution:
# Other equivalent solutions exist for each conversion.
IN_CHAR regA
STORE regA 14

(b) (5 points) OUT_CHAR 'a'

Solution:
OUT_CHAR 'a'

(c) (10 points) ADD s4 3 s2

Solution:
LOAD 4 regA
VAL_COPY 3 regB
ADD regA regB regC
STORE regC 2

(d) (10 points) AR_SET_IDX a9 7 'a'

Solution:
LOAD 9 regA
VAL_COPY 7 regB
VAL_COPY 'a' regC
ADD 1 regA regA
ADD regA regB regA
STORE regC regA

(e) (10 points) AR_GET_IDX a6 s11 s12

Solution:
LOAD 6 regA # ptr to array
LOAD 11 regB # index
ADD 1 regA regA # address of index 0
ADD regA regB regC # address of desired index
MEM_COPY regC 12
Rubric for Q2:

- a. No partial credit
- b. No partial credit
- c.
  - 2pts: load s4
  - 2pts: val_copy 3
  - 3pts: add two registers
  - 3pts: store in memory location 2
- d.
  - 1pt: val_copy
  - 2pts: load a9 pointer
  - 4pts: add index to array pointer
  - 1pt: add one to index pointer to account for size value in array
  - 2pts: store ’a’ in the array using the index register
- e.
  - 1pt: load s11
  - 1pts: load a6 pointer
  - 2pts: add index to array pointer
  - 1pt: add one to index pointer to account for size value in array
  - 3pts: load result from index pointer
  - 2pts: store result in s12
Question 3: Finite Automata .......................................................... 40 points

(a) (20 points) Convert the above Non-deterministic Finite Automata (NFA) to a Deterministic Finite Automata (DFA).

Solution:

(b) (10 points) Fill in the bubbles of the strings that the NFA accepts?

- 0
- 1
- 001
- 0111
- 0010
- 100
- 010110

(c) (10 points) Fill in the bubbles of the following regular expression that accept the same language as the NFA? (choose one or more)

- [01]*0[01]*  
- 0+(01)*  
- 0[01]*  
- 0+(0 | 1)*
Rubric for Q3:

- a.
  - Not a DFA: 0
  - DFA with large differences: 5
  - DFA with minor differences: 10
  - Correct: 20
- b. No partial credit
- c. No partial credit
Question 4: Compile To LMAOcode ................................. 40 points

LOLcode:

HAI 1.450
I HAS A index ITZ A NUMBR AN ITZ QUOSHUNT OF WHATEVR AN 4
I HAS A counts ITZ LOTZ A NUMBRS AN THAR IZ 26
IN counts‘Z index PUT SUM OF counts‘Z index AN 1
I HAS A another ITZ LOTZ A NUMBRS AN THAR IZ 2
another R counts
KTHXBYE

(a) (30 points) Given the LOLcode above, fill in the blanks in the compiled LMAOcode.

<table>
<thead>
<tr>
<th>command</th>
<th>arg1</th>
<th>arg2</th>
<th>arg3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANDOM</td>
<td>s11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>4</td>
<td>s14</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td>s11</td>
<td>s14</td>
<td>s7</td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>s7</td>
<td>s18</td>
<td></td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>26</td>
<td>s26</td>
<td></td>
</tr>
<tr>
<td>AR_SET_SIZE</td>
<td>a2</td>
<td>s26</td>
<td></td>
</tr>
<tr>
<td>AR_GET_IDX</td>
<td>a2</td>
<td>s18</td>
<td>s20</td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>1</td>
<td>s22</td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td>s20</td>
<td>s22</td>
<td>s1</td>
</tr>
<tr>
<td>AR_GET_IDX</td>
<td>a2</td>
<td>s18</td>
<td>s45</td>
</tr>
<tr>
<td>AR_SET_IDX</td>
<td>a2</td>
<td>s18</td>
<td>s1</td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>2</td>
<td>s99</td>
<td></td>
</tr>
<tr>
<td>AR_SET_SIZE</td>
<td>a3</td>
<td>s99</td>
<td></td>
</tr>
<tr>
<td>AR_COPY</td>
<td>a2</td>
<td>a3</td>
<td></td>
</tr>
</tbody>
</table>

(b) (10 points) What are the locations (e.g. s54) of the following symbols?

index:  s18
counts: a2
another: a3
Rubric for Q4:

- a. 2pts per correct answer
- b. 3pts per correct answer, full pts if all correct
Question 5: DFA to CFG

(a) (10 points) Fill in the bubbles of the strings that the DFA accepts?
- 0
- \(0\)
- \(001\)
- \(0111\)
- \(0010\)
- \(010110\)

(b) (30 points) Write a Context Free Grammar that accepts the same language as the
Deterministic Finite State machine above.

**Solution:**

\[
S \rightarrow T \ '1' \\
T \rightarrow \epsilon | '1' | '0' | T \ T
\]
Rubric for Q5

- a. 3pts per correct answer, full pts if all correct
- b.
  - 10pts - CFG accepts all the strings of language $0^*1^+$
  - 10pts - CFG accepts all the strings of language $(0+1^+)^+$
  - 10pts - CFG accepts all the strings of language $(0^*1^+)(0+1^+)^*$
  - +5/-5pts - Whether student recognized the CFG should end with 1
  - +5/-5pts - Whether student recognized if it is necessary that the CFG should start with 0
Question 6: Constructing Regular Expressions .......................... 40 points

(a) (20 points) Design a regular expression that will match telephone numbers with an optional area code.
For example, (123)456-7890 and 456-7890 are matched but 1234567 is not.

**Solution:** \( (\([0-9]{3}\))? [0-9]{3} \- [0-9]{4} \)

(b) (20 points) Design a regular expression that will match hex color values. Colors begin with a hash-sign and have a 3 or 6 hexadecimal digits (with entirely uppercase or lowercase letters)

**Good examples:**
- #FFF
- #12a
- #FF1
- #0ABCDE
- #ABCDEF
- #12ab12

**Bad examples:**
- FFF
- #34AbBB
- #34

**Solution:**
```javascript
#(
  [0-9a-f]{3} | [0-9a-f]{6} | [0-9A-F]{3} | [0-9A-F]{6}
)
```
Rubric for Q6:

- a.
  - 5pts: Three digits 0-9 for area code
  - 5pts: Three digits 0-9 for middle section
  - 5pts: Four digit 0-9 for last section
  - 2pts: Escape parenthesis for area code
  - 2pts: Make area code optional
  - 1pts: Dash between middle and last section

- b.
  - 2pts: Hash-sign at the beginning
  - 5pts: Hex letters are all uppercase with integer digits
  - 5pts: Hex letters are all lowercase with integer digits
  - 4pts: 3 hexadecimal digits
  - 4pts: 6 hexadecimal digits
**LMAOcode Overview**

**Arguments**

VAL: This argument uses a value; it can be a literal integer, a literal char, a label, or a scalar variable.

SCL: This argument must be a scalar variable (which will be written to)

ARR: This argument must be an array variable

**Instructions**

ADD [VAL: num1] [VAL: num2] [SCL: result]  
SUB [VAL: num1] [VAL: num2] [SCL: result]  
MULT [VAL: num1] [VAL: num2] [SCL: result]  
DIV [VAL: num1] [VAL: num2] [SCL: result]  

Apply the given math operation on num1 and num2, and place the answer in result.

VAL_COPY [VAL: from] [SCL: to]  
OUT_NUM [VAL: num]  
OUT_CHAR [VAL: char]  

**TEST** [VAL: num1] [VAL: num2] [SCL: result]  
Options are: TEST_LESS, TEST_GTR, TEST_EQU, TEST_NEQU, TEST_LTE, or TEST_GTE  

Compare num1 and num2. Set result argument to 0 or 1 based on if condition is false or true.

JUMP [VAL: line]  
JUMP_IF_0 [VAL: test] [VAL: line]  
JUMP_IF_NO [VAL: test] [VAL: line]  

RANDOM [SCL: result]  
IN_CHAR [SCL: result]  

AR_GET_IDX [ARR: array] [VAL: index] [SCL: result]  
Look up index in array and store its value as result.

AR_SET_IDX [ARR: array] [VAL: index] [VAL: value]  
Look up index in array and set its value to value.

AR_GET_SIZE [ARR: array] [SCL: result]  
Look up the size of array and store it in result.

AR_SET_SIZE [ARR: array] [VAL: size]  
Create an array with new size space for elements

AR_COPY [ARR: array1] [ARR: array2]  
Copy the contents and size of array1 into array2.

**Labels**

A label is a string of alphanumerical characters, beginning with a letter that is used to reference a line number elsewhere in the code. When a label is created, it must be placed at the beginning of a line of code and it must end with a colon (‘:’). A label will typically be used to indicate the end point in a jump command.

**ROFLcode Overview**

ROFLcode (assembly) is very similar to LMAOcode (intermediate code), with a handful of changes

- The AR_* instructions are not available.
- Scalar variables are not available, but eight registers (regA through regH) take their place.
- Three new instructions are available that allow you to interact with memory. They are:

LOAD [VAL: from] [REG: to]  
STORE [REG: from] [VAL: to]  
MEM_COPY [VAL: from] [VAL: to]  
Load the value in memory position from into register to.  
Store the value in register from into memory position to.  
Copy the value in memory position from into memory position to.