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

Actual Exam for CSE 450 (2018)

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 them, which indicates that they 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 **LMAOcode** instruction. Memory location 0 stores the start of the free memory.

<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>87</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>63</td>
<td>0</td>
<td>81</td>
<td>‘b’</td>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>New Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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 **ROFLcode** instructions.

<table>
<thead>
<tr>
<th>Location</th>
<th>60</th>
<th>61</th>
<th>62</th>
<th>63</th>
<th>64</th>
<th>65</th>
<th>66</th>
<th>67</th>
<th>68</th>
<th>69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>‘*’</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>80</th>
<th>81</th>
<th>82</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Value</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 2: Multidimensional Arrays ........................................... 40 points
Many languages allow for multidimensional arrays (also called arrays-of-arrays), but out of simplicity, I’ve not had you implement them in the projects. But I didn’t say anything about on the exam.

You want to implement a new literal in LOLcode that represents 2-dimensional arrays of TROOF literals. These arrays will be used to implement complicated binary logic.

Examples of some legal 2-dimensional array literals are:

- `[WIN]` : 2D array can hold TROOF literals
- `[[WIN], [FAIL]]` : 2D array can also hold arrays of TROOFs
- `[WIN, [FAIL, WIN]]` : 2D arrays don’t need to be regular
- `[WIN, [FAIL, WIN, WIN], WIN]` : 2D array can have many elements

Examples of **ILLEGAL** 2-dimensional array literals are:

- `[]` : 2D arrays can’t be empty
- `[WIN, []]` : 2D arrays can’t have empty arrays as elements
- `[4]` : 2D arrays elements can’t be NUMBR literals
- `[WIN FAIL]` : Commas are not optional
- `[WIN, FAIL, FAIL,]` : No trailing comma
- `[WIN, [WIN, [FAIL]]]` : Can have at most 2 dimensions (not 3)

Write a set of Context-Free Grammar rules that capture the grammar for `2d_array_literal`. The only terminal symbols you may use are: `LETTR_LITERAL, NUMBR_LITERAL, TROOF_LITERAL, OPEN_BRACKET, CLOSE_BRACKET, COMMA` and `EPSILON`. 
Question 3: Non-deterministic Finite Automata

(a) (10 points) Fill in the bubbles of the strings that the above Non-deterministic Finite Automata (NFA) accepts?

- 1
- 0
- 100
- 001
- 101001
- 1111
- CB
- \( \varepsilon \)

(b) (25 points) Convert the above NFA to a Deterministic Finite Automata (DFA).

(c) (5 points) Convert the DFA you created above into a table. Use a slash to denote a halt on failure. Note: There may be additional rows that you may not need for your solution.

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 4: Parsing Python Code ..................................................... 40 points
x = 5
print("x = ", x)
while 7:
    print(10 - x)
    x = x + 1
    if x > 20:
        break

(a) (5 points) How many symbols are added to the symbol table when the Python
source code above is compiled?

(b) (5 points) How many literals are present in the source code?

(c) (30 points) Draw an abstract syntax tree which may be generated by parsing the
source code.
Question 5: AR\_SET\_SIZE\_ZEROES ................................. 40 points

AR\_SET\_SIZE is used to create an array of a particular size at the start of free memory. Normally, this command doesn’t adjust any memory associated with the space set aside for holding the elements of the array. The interpreter provided for LMAOcode and ROFLcode guarantees that all memory positions initially hold a zero, until changed to other values by the program. However, my boss just told me that needing to ensure that memory was zero’ed out before the program could use it is too slow, as most programs don’t rely on that feature. So, we removed that feature, meaning that memory initially is filled with garbage/random values. But, we were asked to add a modified version of AR\_SET\_SIZE (named “AR\_SET\_SIZE\_ZEROES”) to LMAOcode that performs the same role of AR\_SET\_SIZE and then also ensures that all the elements of the array are initialized to zero.

Write the ROFLcode instructions that should be added to the AR\_SET\_SIZE instructions below that performs “AR\_SET\_SIZE\_ZEROES a99 s66”. Be sure to add comments denoting what registers and instructions are performing which roles so we can award partial credit if needed.

LOAD 0 regA
STORE regA 99
LOAD 66 regB
STORE regB regA
ADD regA regB regC
ADD 1 regC regC
STORE regC 0
# What instructions should be added here?
Question 6: Compile LOLcode To LMAOcode ............................. 40 points
Given the source code, fill in the blanks in the LMAOcode generated by my compiler.

```
HAI 1.450
I HAS A dog ITZ LOTZ A TROOFS AN THAR ITZ 2
O RLY? SAEM GIMMEH 'A'
YA RLY
    VISIBLE IN dog`Z 1 PUT FURSTBIGGR 10 WHATEVR
NO WAI
    I HAS A dog ITZ A NUMBR AN ITZ 4
    VISIBLE dog !
OIC
KTHXBYE
```

<table>
<thead>
<tr>
<th>command</th>
<th>arg1</th>
<th>arg2</th>
<th>arg3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL_COPY</td>
<td>2</td>
<td>s2</td>
<td></td>
</tr>
<tr>
<td>IN_CHAR</td>
<td>s3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>s3</td>
<td>s4</td>
<td>s5</td>
</tr>
<tr>
<td>JUMP_IF_0</td>
<td>s5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>10</td>
<td>s6</td>
<td></td>
</tr>
<tr>
<td>RANDOM</td>
<td>s7</td>
<td>s6</td>
<td>s7</td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>1</td>
<td>s9</td>
<td></td>
</tr>
<tr>
<td>AR_GET_IDX</td>
<td>a1</td>
<td>s9</td>
<td>s10</td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>s8</td>
<td>s10</td>
<td></td>
</tr>
<tr>
<td>OUT_CHAR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>label_a:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAL_COPY</td>
<td>s11</td>
<td>s12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>label_b:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>