INSTRUCTIONS

Read the following carefully before answering any question.

- Make sure you have filled in your name and USB ID number in the space above.
- Write your answers in the space provided; Keep your answers brief and precise.
- The exam consists of 4 questions, in 8 pages (including this page) for a total of 30 points.

Question 1 has two pages.
Question 2 has one page.
Question 3 has two pages.
Question 4 has two pages.

<table>
<thead>
<tr>
<th>Question</th>
<th>Max.</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>
1. [Total: 9 points] For this question, consider the Proto(1) program shown below:

```plaintext
1: x = input();
2: a = x + 1;
3: b = x * 2;
4: c = a + b;
5: d = c + x;
6: e = d * b;
7: f = e - a;
8: g = e * f;
9: h = g - x;
10: print (h);
```

(a) [3 points] Using liveness analysis, compute the set of variables live at each program point in the above program.

(b) [3 points] Show the register interference graph for the above program.
(c) [3 points] Can we allocate all variables using only 4 registers using the Simplify-Select-Spill algorithm \textit{without} spilling? If so, show the allocation. If not, show where the allocation will fail.
2. [Total: 4 points] For this question, consider grammar $G_2$ with start symbol $S$ given below:

$$
S \rightarrow A U \\
S \rightarrow B V \\
S \rightarrow \epsilon \\
U \rightarrow S A \\
V \rightarrow S B \\
A \rightarrow a \\
B \rightarrow b
$$

(a) [2 points] Compute $FIRST$ and $FOLLOW$ of each of the following symbols.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>$FIRST$</th>
<th>$FOLLOW$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$U$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) [2 points] Show that $G_2$ is not $LL(1)$. 

3. [Total: 7 points] Consider the following fragment of an imperative programming language’s grammar:

\[
S \rightarrow \text{if } e \text{ then } S \text{ else } S \\
S \rightarrow \text{if } e \text{ then } S \\
S \rightarrow o
\]

(a) [2 points] Show that this grammar is ambiguous by showing two distinct parse trees for the sentence \textit{if } e \text{ then if } e \text{ then } o \text{ else } o

(b) [3 points] Build the SLR(1) action table for this grammar.
(c) [2 points] Your answer for part (b) will have a shift-reduce conflict. If we want to associate an “else” with the nearest enclosing “if”, how will you resolve this conflict? Why?
4. [10 points total] Consider the following grammar:

\[
S \rightarrow SSa \\
S \rightarrow Sb \\
S \rightarrow c
\]

(a) [3 points] Show that the above grammar is not $LL(1)$.

(b) [3 points] Is the grammar $LL(k)$ for any $k$? That is, is there any constant lookahead that will help us construct a deterministic predictive parser for this grammar? Justify.
(c) [4 points] Is the grammar $SLR(1)$? Justify.