This exam has 3 questions, for a total of 10 points.

1. **2 points** Write down the set of variables that are live before and after each of the instructions in the following pseudo-x86 program.

```plaintext
call _input
movl %eax, w
movl $1, x
movl $2, 1_tmp
addl w, 1_tmp
movl 1_tmp, y
movl $3, z
movl x, 2_tmp
addl y, 2_tmp
addl z, 2_tmp
pushl 2_tmp
call _print_int_nl
addl $4, %esp
```

**Solution:**

```
{}
call _input
{}
    movl %eax, w
    movl $1, x
    movl $2, 1_tmp
    addl w, 1_tmp
    movl 1_tmp, y
    movl $3, z
    movl x, 2_tmp
    addl y, 2_tmp
    addl z, 2_tmp
    pushl 2_tmp
call _print_int_nl
addl $4, %esp
```
2. **2 points** Draw the interference graph for the program in question 1.

```
addl z, 2_tmp
{2_tmp}
pushl 2_tmp
{}
call _print_int_nl
{}
addl $4, %esp
{}
```

Solution:

![Interference Graph](image-url)
3. **6 points** Assign each variable in the program from question 1 to registers or stack locations using the greedy saturation-based algorithm. You are compiling for a computer architecture that has only two general purpose registers, %eax and %ebx, but is otherwise just like the x86. Write down the following information for each step in the algorithm: 1) how many available registers there are for each variable at the beginning of the step, 2) what variable is chosen for assignment in this step, and 3) which register or stack location is assigned to that variable.

**Solution:** (The following is not the only solution.)

<table>
<thead>
<tr>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>1_tmp</th>
<th>2_tmp</th>
<th>chosen variable</th>
<th>chosen color</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>w</td>
<td>eax</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>x</td>
<td>ebx</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1_tmp</td>
<td>-4(%ebp)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>y</td>
<td>eax</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>1</td>
<td>z</td>
<td>-4(%ebp)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>2_tmp</td>
<td>ebx</td>
</tr>
</tbody>
</table>