From last time

In the context of a typical Von-Neumann computer, explain the purpose of: the Memory; the processor; and the Bus between them. (3 marks)
Memory holds data (& instructions); processor does calculations & makes decisions; Bus moves data & results between Memory & Processor

What are the similarities and the differences between computer memory (RAM) and processor registers? (3 marks)
Both hold values; RAM is much bigger but slower than Registers, which can only hold 1 value each.

What are the similarities and the differences between the “3-address” and the “Load-Store” instruction styles? (3 marks)
Both: arithmetic instructions = operation, 2 input operands & a result operand; 3-address: operands = memory locations; load-store: operands = registers, extra (load, store) instructions move values between registers & RAM.
Question

What are the contents of the registers (R0, R1, R2) and variables (anne, tom, fred) after each of the following instructions is obeyed:

LDR R0, anne
STR R0, fred
LDR R1, tom
ADD R2, R0, R1
STR R2, anne
SVC 2

The variables start with these values:
anne is 23
tom is 45
fred is 10
### Answer

<table>
<thead>
<tr>
<th></th>
<th>R0</th>
<th>R1</th>
<th>R2</th>
<th>anne</th>
<th>tom</th>
<th>fred</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>23</td>
<td>45</td>
<td>10</td>
</tr>
</tbody>
</table>

LDR R0, anne  

|   | 23 | ?  | ?  | 23   | 45  | 10   |

STR R0, fred  

|   | 23 | ?  | ?  | 23   | 45  | 23   |

LDR R1, tom  

|   | 23 | 45 | ?  | 23   | 45  | 23   |

ADD R2, R0, R1  

|   | 23 | 45 | 68 | 23   | 45  | 23   |

STR R2, anne  

|   | 23 | 45 | 68 | 68   | 45  | 23   |

SVC 2  

|   | 23 | 45 | 68 | 68   | 45  | 23   |
Question

Assuming the first instruction is at address 0, what is the address of each instruction, and what will be the contents of the PC register after each instruction is obeyed?

<table>
<thead>
<tr>
<th>PC</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LDR R0, anne</td>
</tr>
<tr>
<td>4</td>
<td>STR R0, fred</td>
</tr>
<tr>
<td>8</td>
<td>LDR R1, tom</td>
</tr>
<tr>
<td>12</td>
<td>ADD R2, R0, R1</td>
</tr>
<tr>
<td>16</td>
<td>STR R2, anne</td>
</tr>
<tr>
<td>20</td>
<td>SVC 2</td>
</tr>
</tbody>
</table>
Question

Write ARM instructions to do the following:

If $a$ is greater than $b$, then subtract $b$ from $a$.

i.e. if $a > b$ then $a = a - b$;

```
LDR R0, a
LDR R1, b
CMP R0, R1
BLE no
SUB R0, R0, R1
STR R0, a
no  .  .  .
```
Write ARM instructions to calculate $a = (b \times c) - (b + c)$;

(hint: try to avoid loading b or c more than once)

```
LDR R0, b
LDR R1, c
MUL R2, R0, R1
ADD R0, R0, R1 or SUB R2, R2, R0
SUB R2, R2, R0 SUB R2, R2, R1
STR R2, a
```
Question

Write ARM instructions to put zero into a variable e.g. \( a = 0; \)

(hint: you could use a memory location that contains zero, or try to think of an arithmetic instruction that always gives an answer of zero)

\[
\begin{align*}
\text{LDR R0, zero} \\
\text{STR R0, a} \\
\text{...}
\end{align*}
\]

\text{zero DEFW 0}

or:

\[
\begin{align*}
\text{SUB R0, R0, R0} \\
\text{STR R0, a}
\end{align*}
\]

(or use MOV instruction – lecture 5)