

First Item - The Address of Some Variable

Now, look more closely at what LA did. It put the address of the variable into the register you chose. Give the address in hexadecimal.

0x10010058

Second Item - The Two Instructions that Make Up LA

Next, look at the two instructions which make up the LA pseudoinstruction (the first is LUI). Write down, for your report, exactly what MARS shows in the Basic column of the Text Segment for these two instructions. The register names won't look familiar - this column uses register numbers instead of names - but the second instruction should have a number that you've seen before.

```
lui $1, 0x00001001
```

```
ori $16, $1, 0x00000058
```

Third Item - Hex Encoding

Finally, write down the actual 32-bit hex encodings of these two instructions (which you can find in the Code column). Do you see the constants used by these instructions stored inside those encodings somewhere?

0x3c010001

0x34300058

The constants are stored in the instruction encoding itself. The constant for the LUI instruction is 0x00001001 and the constant for the ORI instruction is 0x00000058. The constant for the LUI instruction is stored in the lower 16 bits of the instruction encoding, and the constant for the ORI instruction is stored in the lower 16 bits of the instruction encoding as well.

Fourth Item - The Entire String Constant

The constant I used was defined as such:

```
minimumString:      .asciiz "minimum: "
```

Fifth Item - The ASCII Codes

Look at the first four characters of that string. Compare these four to an ASCII table (I link to one from the class webpage, but basically any ASCII table will work). What are the ASCII codes (in hex) for these four characters?

0x6d 0x69 0x6e 0x69 (corresponding to m, i, n, i)

Sixth Item - The Address of the String

Now, find the LA instruction in your code which reads this string. Run the program to that point, and use it to figure out the address of this string in memory. (It probably starts with 0x1001....)

0x10010000

Seventh Item - The Word in Memory

The word (or maybe 2 words) in memory which contained the ASCII codes

They were spread across two words:

0x696d000a 0x756d696e

Investigation

Finally, write at least one paragraph about your investigation. You can discuss any one (or more) of the following questions: □ What was something new that you learned from this exercise, and how might you use it in the future? □ What questions do you have that were inspired by this exercise? □ Explain your process for finding the information required by this exercise. Include screenshots to show what you saw, and discuss a bit of what you did. □ I mentioned that the constant loaded by the LA instruction is present in the two instructions that LA becomes. Looking at the instruction encodings, what can you conclude about where the constants are stored? Your answer should be written idiomatically and in a way that is not obviously written by an LLM - don't go for a basic run-of-the-mill answer.

I learned that the LA instruction is actually two instructions, LUI and ORI. I can use this information in the future to understand how the LA instruction works and how it loads the address of a variable. In general, I think that understanding how the instructions work at a lower

level will help me write more efficient code. When it comes to instruction encodings, my hypothesis would be that the constants are stored in the instruction encoding itself. This is because the instruction encoding is 32 bits long, and the constants are 32 bits long as well. This would make sense because the instruction encoding needs to be able to store the constant in order to load it into a register.