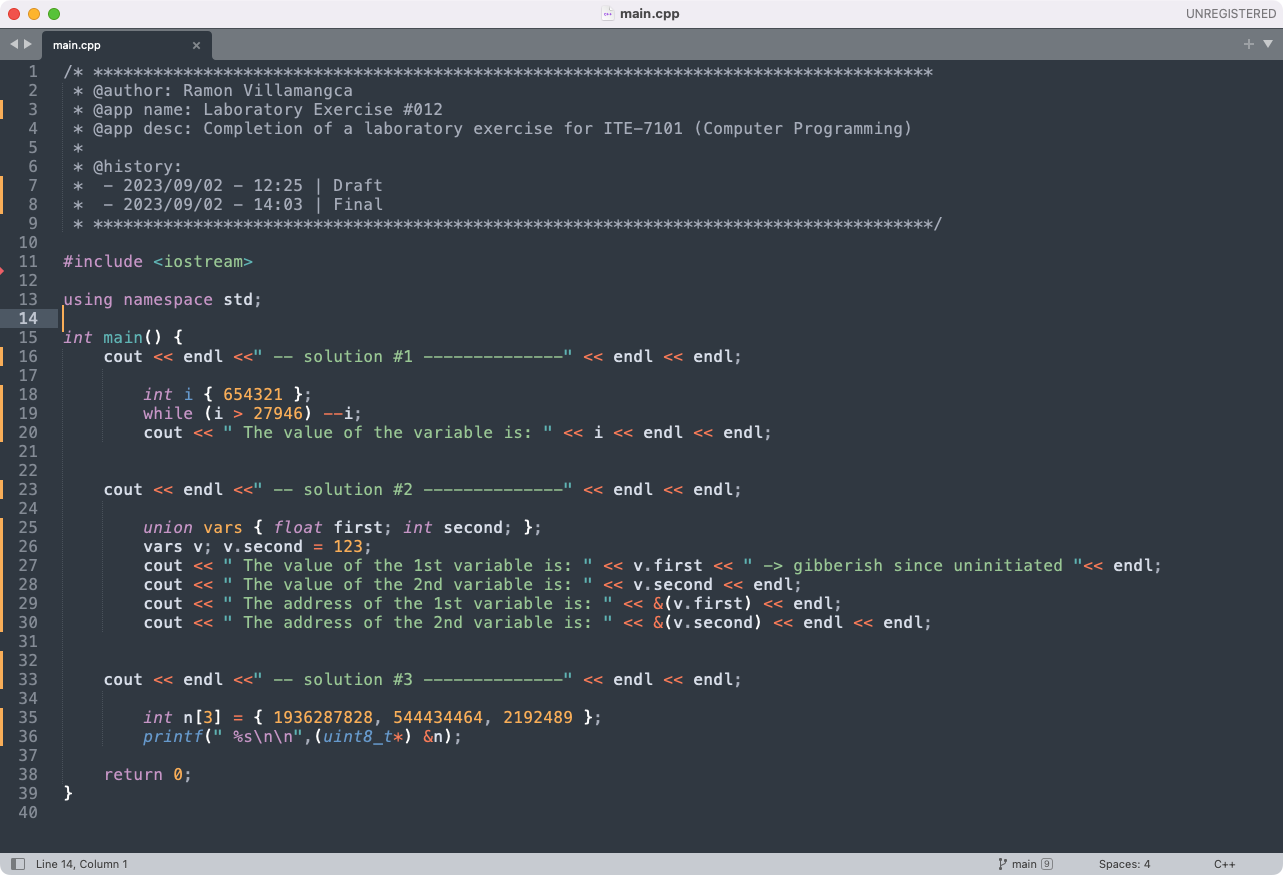
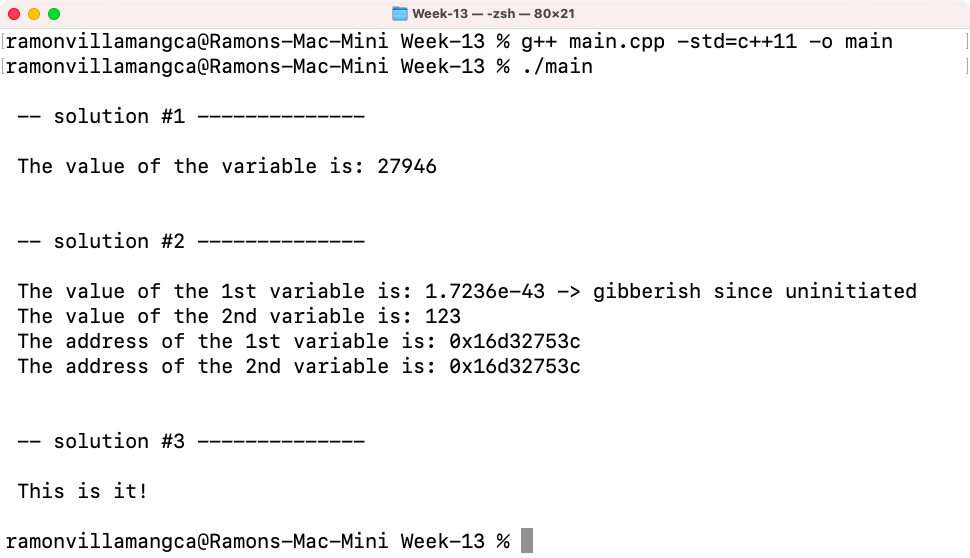
**Screenshot of Solutions and Testing**

****



-----------------------

**Problem 01**

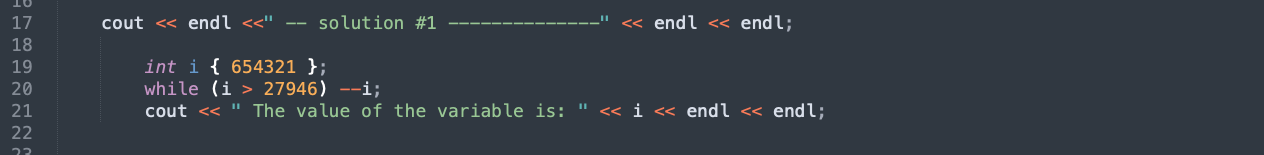
Write a program that will change the value of an integer variable with initial value of 654,321 to 27,946 without directly assigning a value to the variable. You cannot create any pointers or references in the main function.

**Solution 01**

Let us make this straight. You cannot use assignment operators (=, +=, -=, etc.). You cannot use pointers or references. But, I thought we are learning pointers and references in this chapter. Oh well, I’ll use while loop to decrement the variable until we reach the desired value. Since we cannot use assignment operator we will use the unified initialization operator “{}” instead of “=” to initialize the variable.

Note that “decrement” is not an assignment. Decrement maps to the machine/assembly opcode “DEC” while assigment maps to “MOV”. They are totally operations.

Below is the screenshot of the solution program as well as testing in the terminal.





-----------------------

**Problem 02**

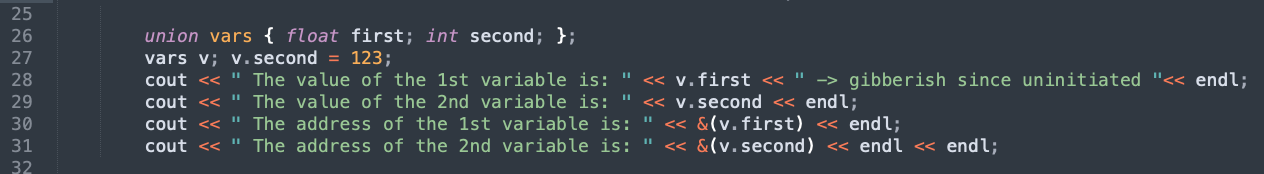
Write a program that will display the address of a float variable and another variable that shares the same address and value as the first variable. Do not initialize the first variable.

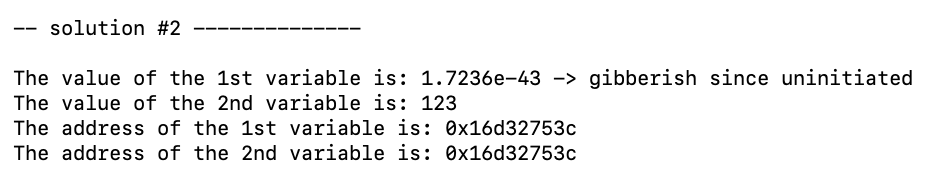
**Solution 02**

Here, the problem is asking us to make two variables with address. This is impossible in C++ primitive types because the compiler will assign a unique address to each variable. The problem maker maybe thinking about pointers or references. But, as we learned so far: pointers and references are NOT variables!

Having said that, the only non-primitive variables in C++ that may share the same address are the member variables inside a “union” data structure. We will therefore, create a union. And to illustrate that the variables do share the same address, we will declare the first variable as “float” and the second as “int”. Furthermore, we will initiate the second but not the first.

Below is the screenshot of the solution program as well as testing in the terminal.





-----------------------

**Problem 03**

Write a program that will display the words “This is it!” from a variable without assigning any characters to the variable. You cannot use cout << “This is it!” << endl; or any variants of it.

**Solution 03**

Now, this one is tricky. Lets see if we can understand this:

* We cannot assign any character (which should include the use of the “char” identifier).
* We cannot use “cout” (and probably any streaming functions that use “<<”).

The second one is easy we just use “printf” from pure C.

For the first one we noticed the the ASCII code for “This is it” is:

[0x54 | 0x68 | 0x69 | 0x73 | 0x20 | 0x69 | 0x73 | 0x20 | 0x69 0x74 | 0x21].

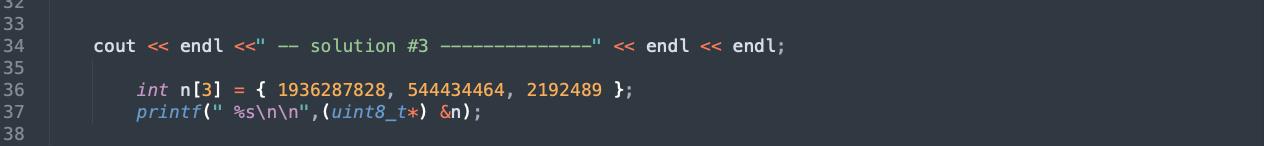
Grouping the bytes into 4 and considering “endianness”, it becomes the following hex integers:

[0x73696854, 0x20736920, 0x217469].

Which is equivalent in decimal integer is: [1936287828, 544434464, 2192489].

So our strategy is to create a integer array with the values above and feed it to printf. We will cast the int array into a byte (uint8\_t) pointer so that printf will access it byte-by-byte. And then force printf to consider byte pointer as a character array (i.e. string) by giving it the identifier “%s”.

Below is the screenshot of the solution program as well as testing in the terminal.





-----------------------

**Problem 04**

What can you conclude from this activity?

**Answer 04**

The solutions presented may not be the exact solutions the problem creator had in mind. But the solutions work and do **satisfy the problem specification**. There is always more than one way to solve a programming problem.

Pointers and references are the trickiest concept in C/C++. It is the stumbling block of most newbies. I remember one professor said that “You cannot consider yourself a master of C/C++ if you haven’t understood the concept of pointers.”

------------------------

**NOTE FOR THIS PROBLEM:**

Source code can be found on my GitHub page: <https://github.com/rvillamangca/>.