

Machine Learning Assignment 98

Elijah Tarr

March 5, 2021

1 Question 98-2

(Question) Suppose you create an array that contains all the lowercase letters of the English alphabet in alphabetical order. What would the length of this array be? (If your answer is 26, please re-read the paragraphs above.)

The actual contents of the array would contain each of the lowercase letters in the alphabet, meaning 26 elements, but C++ doesn't store the length of the string in memory. The way it gets around this restriction is through a null terminator, which is just a unique byte (or series of bytes) in memory that signifies the array ending. Usually, it's just a bunch of zeros. This, however, adds another element to our array, increasing its length by 1, making the total length 27.

(1) Suppose you use `int x 5` to set the variable x to have the value of 5. What is the difference between x and x?

Adding an ampersand in front of a variable name gets the memory address of the variable, instead of the value at that memory address. So, printing x would yield the value 5, but printing x would yield some (seemingly) random string of bytes. This string of bytes is the memory address in your RAM that holds the value of x, so 5. It can and probably will change every time you run the program, since your operating system performs many tasks in the background, each of which reserve some space in memory, meaning C++ has to store the value elsewhere.

(2) Suppose you want to make a pointer p that points to the memory address of x (from question 1). How do you initialize p?

The format for making a pointer is by adding an asterisk after the variable type in its declaration, so you would say `int* p = &x;`

(3) Suppose you have `int v{ 5 }; int ptr{ v }; Without using the symbol v, what notation can you use to get the value of v? (Hint: get the value stored at the memory address of v)`*

The asterisk is the dereferencing operator, so you would say `*ptr`.

(4) Suppose you initialize a pointer as an int. Can you use it to point to the memory address of a variable that is a char?

Yes. When C++ allows you to look in certain positions in memory, it grants you the responsibility of being safe with it. Since the character type is just

a number, which corresponds to a character, you can reference a number in memory and cast it to a char.