CS061 – Lab 7

How does "cin" actually work!

1 High Level Description

The purpose of this lab is to show the intricacies of how keyboard characters are used, and discover that taking in user input is not as easy as it may seem. Even though in C++ we can simply type in a word to *cin*, there's actually quite a bit of instructions necessary to execute an operation that seems so straightforward!

2 Our Objectives for This Week

1. Exercise 1 ~ Demonstrate the shortcomings of the palindrome lab when using backspace
2. Exercise 2 ~ How to "correctly" handle the backspace key
3. Exercise 3 ~ How to further accommodate the backspace key

3.1 Exercises

Exercise 1

Recall that a palindrome is any word or combination of characters that is the same forwards as it is backwards. In the context of C++, a palindrome is any string that is the same forwards as it is backwards. The same logic applies to LC-3 as well, however the key difference between C++ and LC-3 is the method in which we take in input. Let's verify this further below:

Copy your all code from Lab 6 Ex. 3 into Lab 7 Ex. 1. Try typing a palindrome, but click backspace as you type the string in. For example, if I try typing in the palindrome "evilolive", I may accidentally type in the wrong character, and use backspace as following:

e -> **v** -> **o** -> **backspace** -> **i** -> **l** -> **o** -> **l** -> **i** -> **v** -> **e**

Ex 1. Q1. Type in a palindrome, but use the backspace key while doing so. Does your subroutine correctly print the string to the screen

Yes, it does print the string correctly

Ex 1. Q2. Does your program correctly show that the string is a palindrome? Why not? Hint: What is actually happening when you type in a backspace?

It says the string isn't a palindrome and this is because the backspace is also stored into the array and counts as a character of comparison

Discuss with the TA why that is and any possible solution to this problem. (Hint: Look at the next exercise)

Exercise 2

The shortcoming of the code in the previous exercise is that backspace is considered its own character as well! Since it has its own ascii value (Remember, almost every single key you can press on your keyboard has an ascii value associated with it!), the backspace key will take its own address in memory, considering it is not a newline! This sets up our objective for this exercise : *Ex 2 Q1. How do we get the backspace key to work as we typically expect it to work?*

We would have to manually code to remove the previous value stored

We want to ensure that backspaces will not affect the results of our palindrome checker.

Our palindrome checker is actually perfect! The real issue here has to do with the way we take

in input. For this exercise, you will be modifying your SUB_GET_STRING subroutine to correctly handle backspaces. Do you remember how to recognize your “sentinel character”? The backspace character is no different than any other character; when you type it in, it is recognized based on its ascii value.

Instead of putting the backspace into your array like all other characters, you will now have to handle backspaces differently. Rather than adding a new character into the array, you should remove the character at the previous index in the array. When you take the backspace character, make sure to place the value of 0 at that location in memory so you can effectively “backspace” the previous character.

Exercise 3

Ex. 3 Q1. Now, what will happen when you click backspace more than the number of characters in your array?

I think the code would crash since it’s not taking in any real characters.

Let’s try it. Click backspace more than the number of characters you typed in, then type some more characters and see what happens. You don’t need to code a solution in this exercise, but you must be able to explain to us a solution to fix this issue. Please answer the following questions.

Ex. 3 Q2. How would you update your code to resolve this issue?

The code will just output it’s a palindrome because there’s nothing to compare.

Ex. 3 Q3. What would happen if you have important data above your array?

It will all erase and convert the array to 0

3.2 Submission

Your TA will maintain a Google Sheet for questions and demos.
Demo your lab exercises to your TA ***before you leave the lab***.
If you are unable to complete all exercises in the lab, demo the rest during office hours *before* your next lab to get full credit. You can demo during the next lab for -3 points.
Office hours are posted on Canvas (under Syllabus -> Office Hours) and Discord (#office-hours).