# Comp 3350: Computer Organization & Assembly Language

# HW # 10: Theme: Strings and Arrays

*(All main questions carry equal weight. Credit awarded to only those answers for which work has been shown.)*

**Table of Contents:**

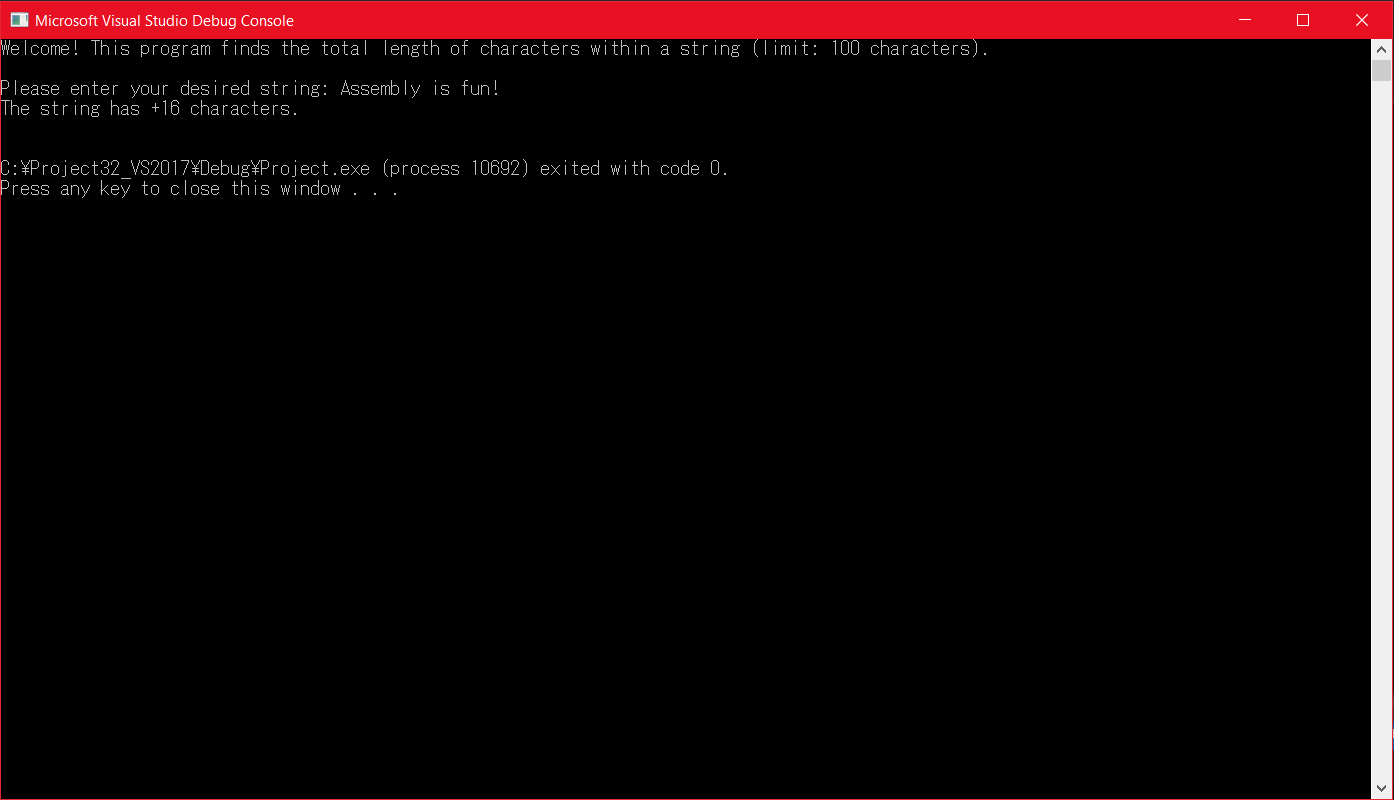
* Question 1 is answered on page 1
* Question 2 is answered on page 2
* Question 3 is answered on page 3
* Question 4 is answered on page 4

1. [Interrupts] What are hardware and software interrupts? Give examples of each. What are maskable and non-maskable interrupts? Provide examples of each.

**Answer:** Interrupt handlers, otherwise known as interrupt service routines, are ways of interrupting a process for the purpose of simplifying input/output as well as performing basic system tasks. An example of using a general interrupt handler would be to allow your program to activate when a hot key, or a specific keyboard input, is pressed; this would interrupt whatever task is running and instead boot your program up. An interrupt handler is more specifically used to handle hardware/software interrupts. A hardware interrupt occurs when a hardware device (asynchronous port, keyboard, timer, and so on) sends a signal to the Programmable Interrupt Controller chip. The hardware interrupt is then generated by the Intel 8259 Programmable Interrupt Controller (PIC), which signals the CPU to suspend execution of the current program and execute an interrupt service routine. A good example of this would be when characters received from the serial port would be lost; an interrupt-driven routine instead stores them in a buffer. A software interrupt is a call to an operating system interrupt handler, effectively providing input-output capability to application programs. Specific examples of software interrupts would be when a file needs to be read or written to, when the program needs to set/retrieve the system time and date, or when characters/strings need to be displayed. A maskable interrupt is an interrupt that can be ignored by the processor while it is performing its operations. An example of a maskable interrupt would be when mouse clicks need to be interpreted during a program’s runtime, or when memory needs to be read; the interrupt caused by these actions are instead overridden so that they can occur in the background while the program is running. A non-maskable interrupt is the job of the external hardware and is activated when a memory error occurs (these are also generally specified as software interrupts). Unlike a maskable interrupt, a non-maskable interrupt cannot be ignored using interrupt masking techniques. Examples of this would be when memory gets corrupted, a program tries to divide by zero, or parity errors arise within a program; however, the best example would be when a user presses ctrl + alt + del to immediately signal the system while the computer is in a non-responding state.

1. [Strings] Write a program that computes the number of characters in any string. Test the robustness of your program using different strings of including those of size 0.

**Screenshot:**



**Note:** This program uses the ReadString procedure that can be found in the Irvine32 library, as it is the only program that reads in a string (but it also determines the amount of characters entered). Since the question asked for me to write a program that determines how many characters are in the string, I will immediately clear the eax register after the procedure is called and generate the string length through my own program. Additionally, the asm file will be attached with the submission (Question2HW10.asm).

1. [Structures] Using the structure example discussed in the book and slides, write a program that displays the *x*-coordinates of several points given as an array of coordinates in the data segment. Test your program with various *(x, y)* inputs. Use base-indexed addressing to implement the program.

**Screenshots:**



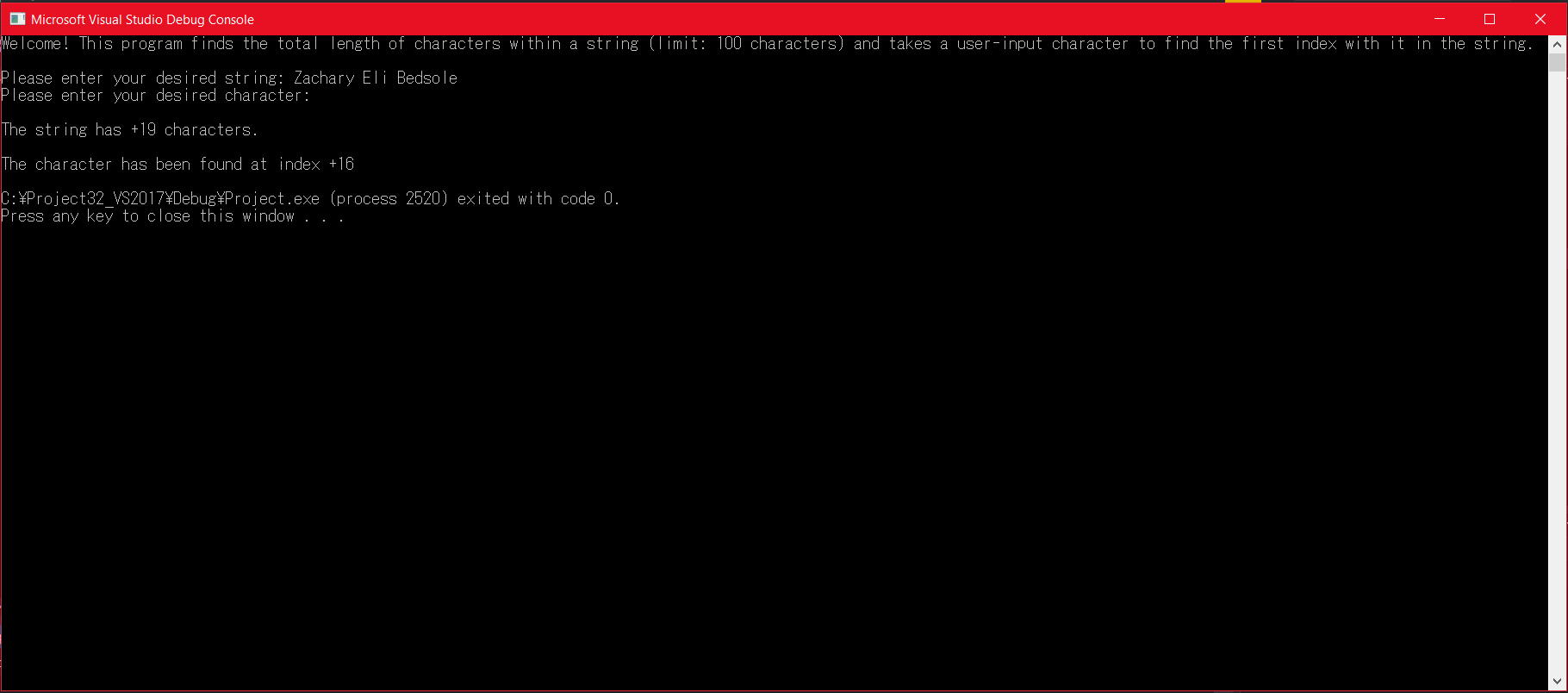




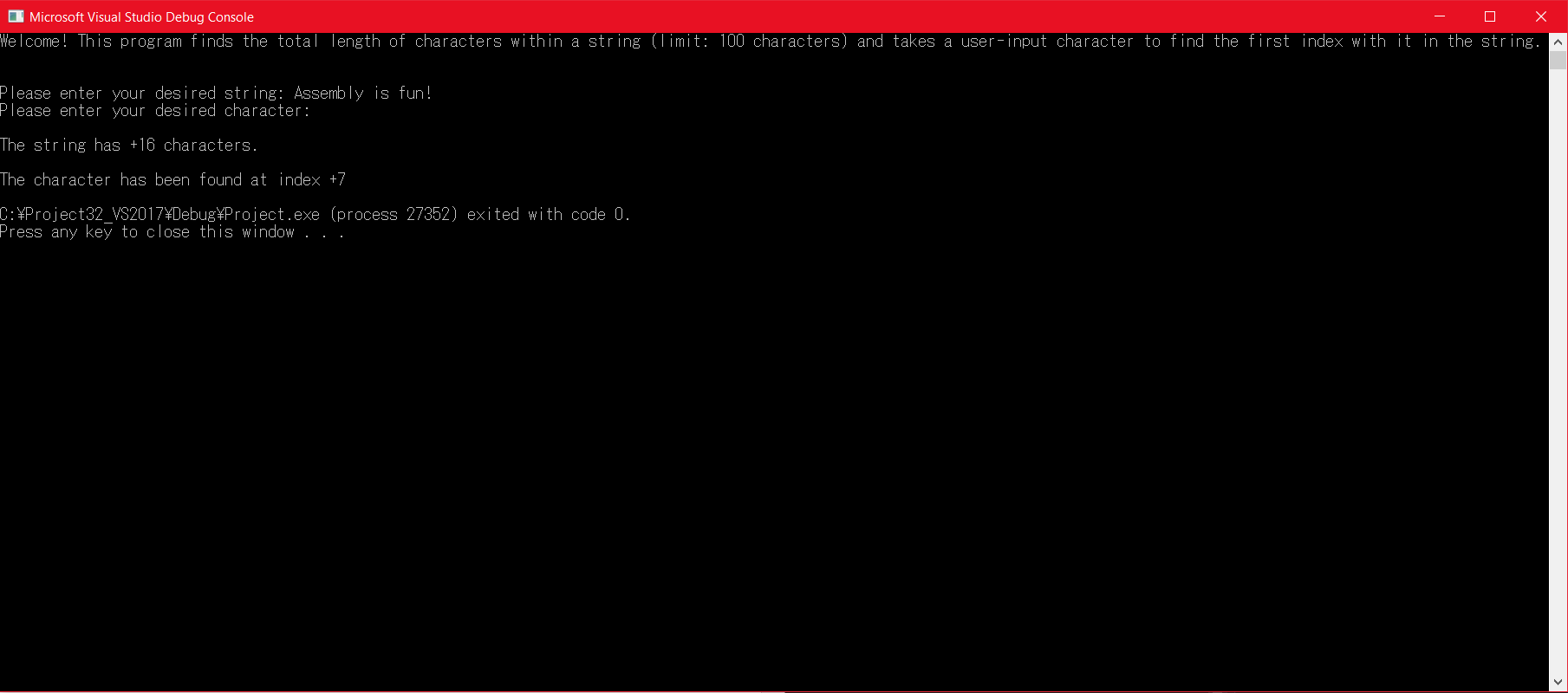
**Note:** The asm file will be included in the submission (Question3HW10.asm).

1. [Strings] Write a program that searches for a character in a string. You should set the EDI pointer to point to the character found. Test the program thoroughly using various strings, including your name. Provide screen shots of the runs along with your program. You must use string instructions in your program.

**Screenshots:**



**Note:** The above program is looking for the character ‘o’.



**Note:** The above program is looking for the character ‘y’.



**Note:** The above is a run with an empty string and no character provided.

**Last Note:** The asm file will be included with the submission (Question4HW10.asm). The ReadChar procedure also does not show what value was entered, so I have included which characters I searched the strings for.