Strings as function arguments, malloc, and structures

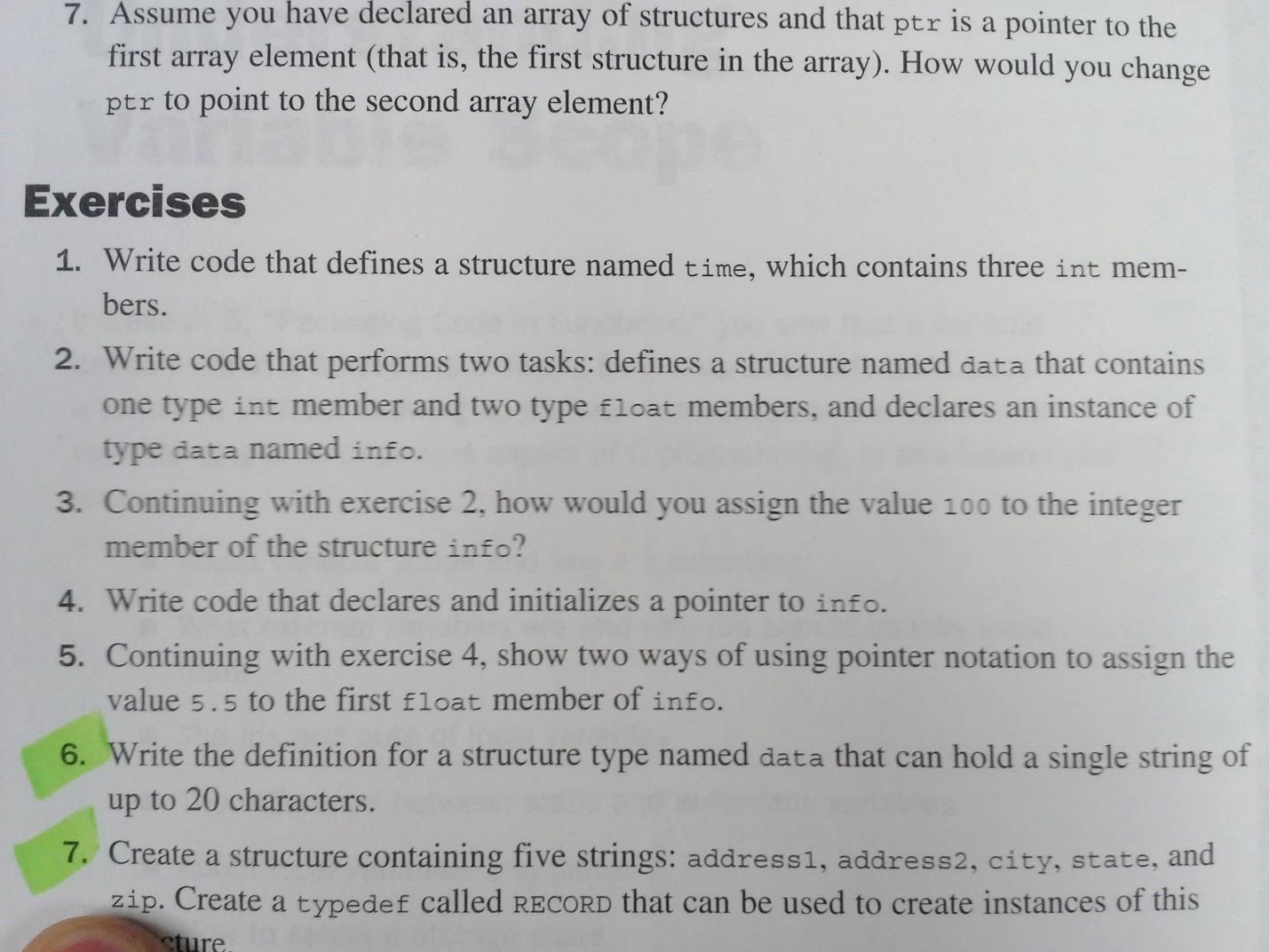
1] Write a function **underscore()** that accepts one string **st** as argument.

* The length of **st** is unknown in advance. Use the library function **strlen()** to determine it.
  + Don’t forget to include **string.h**!
* Use **malloc()** to allocate enough memory to hold a string that is one character longer than **st**.
* Copy **st** to the block of memory allocated, and add the character *underscore* \_ at the end.
  + Use a loop to copy character-by-character. What type of loop?
  + Don’t forget the string terminator!
* **underscore()** returns a pointer to the new string.

In the main program, ask the user to enter a string, and then print the string returned by **underscore()**; for example, if the user enters **abcd**, the program prints **abcd\_**

Remember to liberate the allocated string before ending the program.

2] Write a program that solves Exercise 2 / p.277:



Initialize info with some data.

Then create a pointer **infoPtr** to **info** and print **info** two ways:

* using the variable name **info**
* using **infoPtr**

3] Using the previous program:

* Declare pInt, pFloat1, and pFloat2 to be pointers to integer and floats.
* Initialize so they will point to the integer and the float members of info.
* Print those values using the pointers.

4] Circle all that apply:

A structure is a data type in which:

A. each element must have the same data type. B. each element must have a pointer type only

C. each element may have a different data type D. No element is defined

**Problems for the lab report**

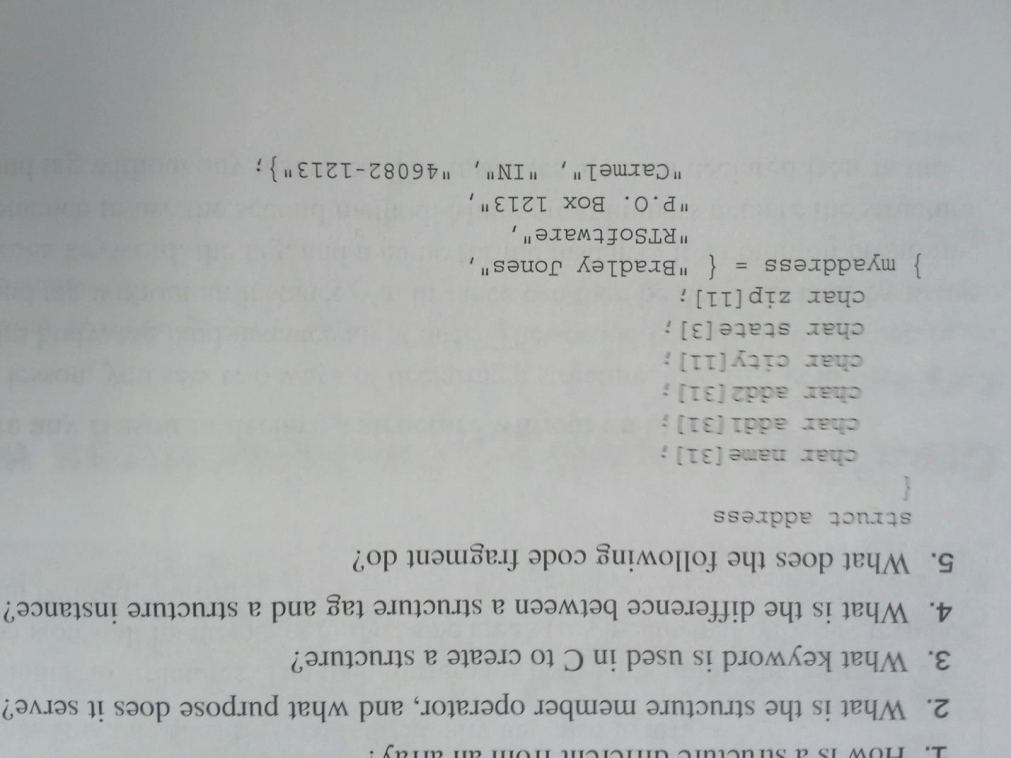
For all problems marked with ►, take screenshots of both code and output, paste the screenshots in a report PDF file, and submit it to Canvas.

5] ► Solve Exercise 7 at the end of Ch.10:

Write a function that accepts 2 strings. Use **malloc()** to allocate enough memory to hold the 2 strings concatenated. Copy the strings to the block of memory allocated. Return a pointer to the new string.

Print the new string in the main program.

Hint: Use **strlen()** to determine the lenghts of the string arguments. (Don’t forget to include **string.h**!)

6] ► Write a program that creates the structure from QUIZ 5 / p.276:

Create a second variable **youraddress** of the same structure type, and initialize it with data of your choice.

Create two pointers **ptr1**, and **ptr2**. Make **ptr1** point to **myaddress** and **ptr2** to **youraddress**.

Print the structures two ways:

* using the variable names
* using the pointers.

7] Circle all that apply:

If one or more members of a structure are other structures, the structure is known as:

A. nested structure B. invalid structure

C. self-referential structure D. unstructured structure

Do all the members of a structure need to have the same size? A. Yes B. No