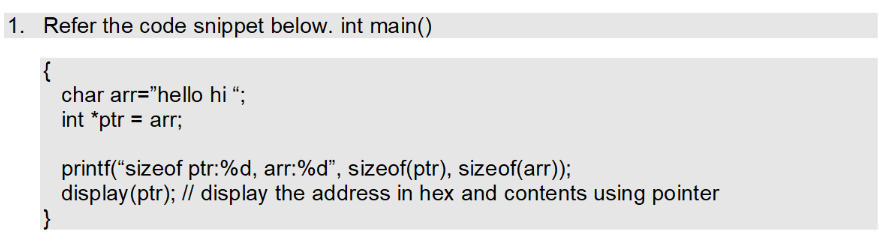
**Introduction to Pointers Assignments**

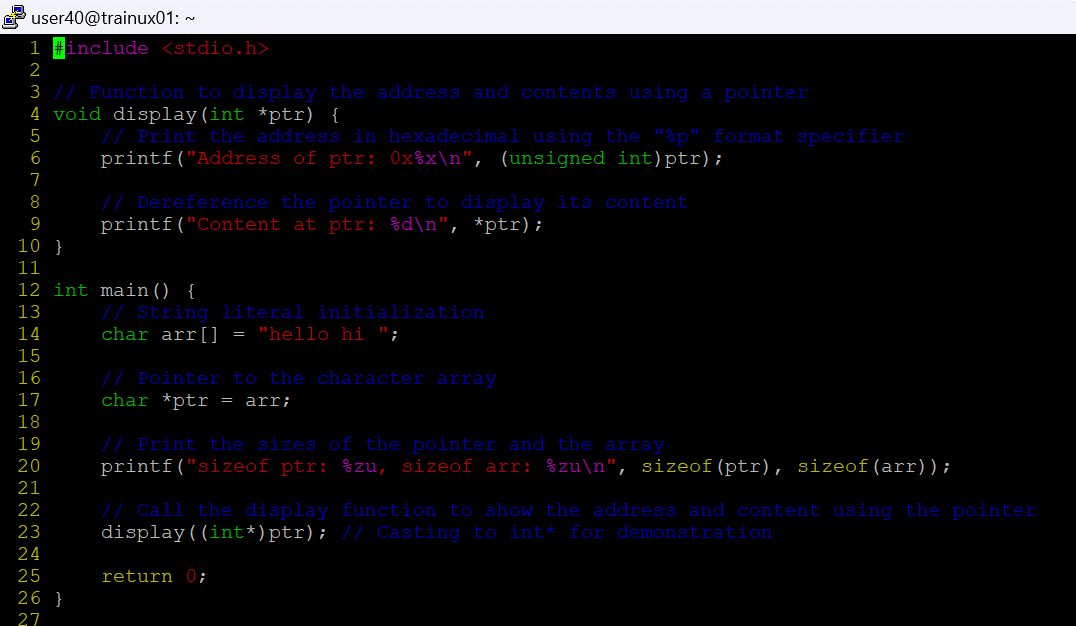
Mandatory



Perform the following.

a. Implement the display() function (Use the “0x%x” formatting specifier to print addresses in hexadecimal.)

b. comment on the sizeof(ptr) and sizeof(arr)



A computer screen with colorful text

Description automatically generated

**sizeof(ptr)** prints the size of the pointer variable, not the data it points to. It’s the size of the pointer itself, which is usually 4 bytes on a 32-bit system and 8 bytes on a 64-bit system.

**sizeof(arr)** gives the size of the entire character array, including the null terminator. For the string "hello hi ", the length of the string (not counting the null terminator) is 9 characters, so sizeof(arr) will return 10 bytes.

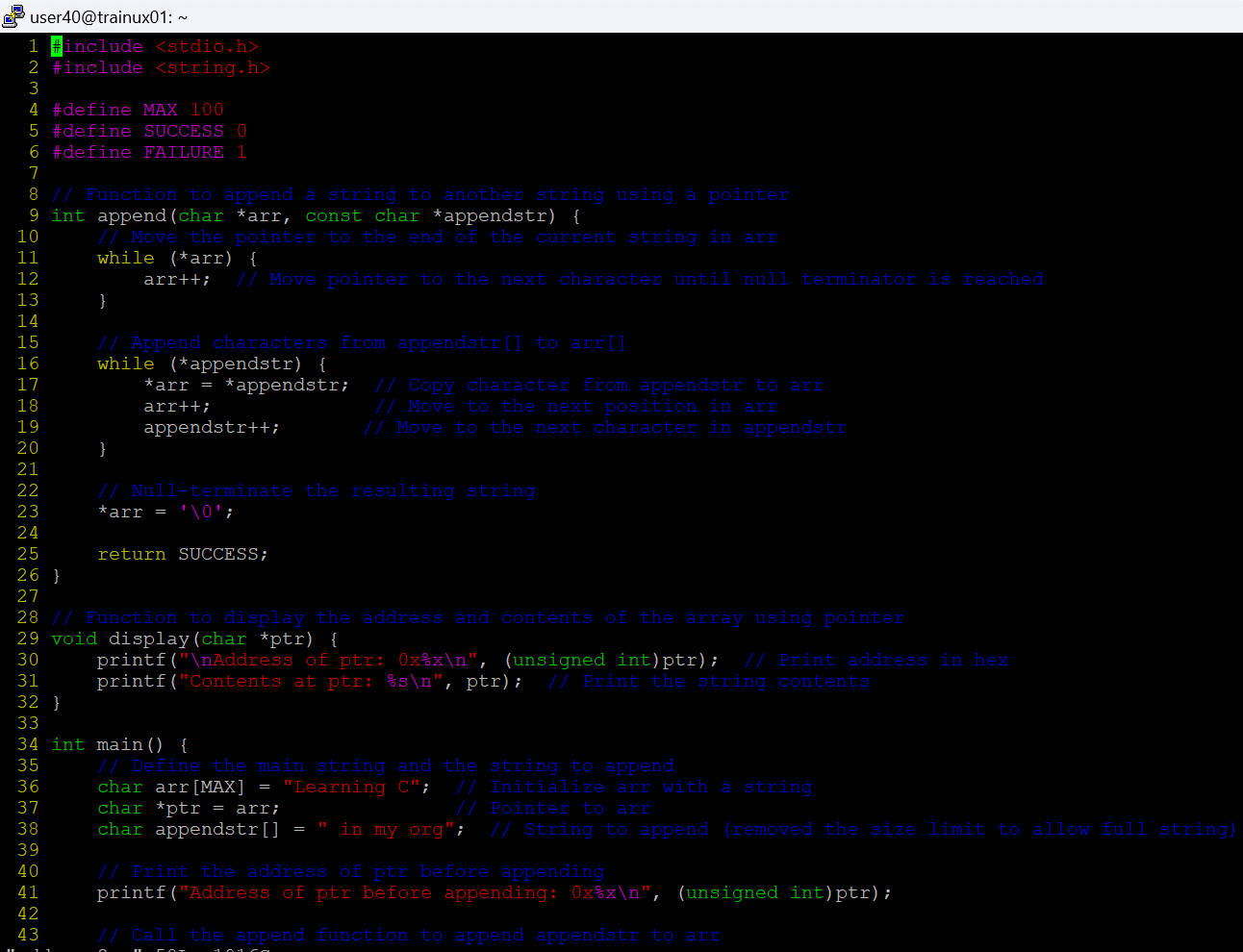
A screenshot of a computer program

Description automatically generated

Perform the following.

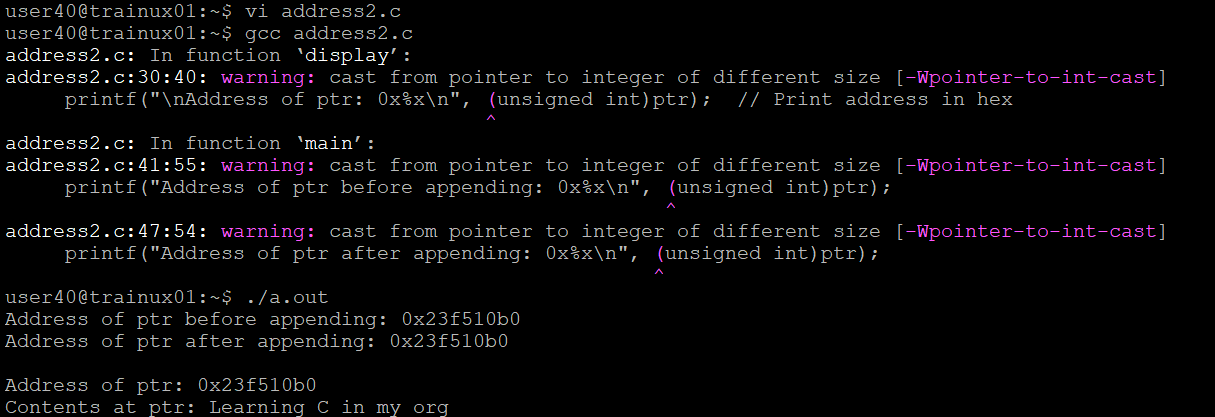
a. Implement the append() function to append the contents of the appendstr[] to arr using pointer.

[Note: append() should only use its content and not manipulate it. Contents should be retained even after the call]



A screen shot of a computer

Description automatically generated



3. Refer the code in “pointer\_prg.c”. The functions swap\_nums() and swap\_pointers() are expected to swap the numbers and pointers respectively. But swap\_pointers() is currently not giving the expected results. Analyse and the fix the issue.

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

