Homework Coding the Call Stack

Modify the starter code to print the **name and memory address** of **each name binding** in the program, and its address.

To print the address of an integer variable, for example, you can use

int k = 5;

printf("k %p", &k);

& means **address of**.   
%p means **pointer**.

Once you have the names and addresses printed out, **manually sort them** in hexadecimal order.

Then, manually write which section they are a part of from amongst **static, code, program heap, program call stack. There are bindings for things from each the 4 section categories.**

**Example:**

j 0x7ffee3a869ec call stack

k 0x7ffee3a869eg call stack

Original Output:

**arr (prgram heap) location: 0000000CF82FF6C8**

**x (static) location: 00007FF6CF07E200**

**y (code) location: 00007FF6CF07E228**

**hello function (code) 00007FF6CF07133E**

**next function (code) 00007FF6CF071154**

**char string[] (call stack) location: 0000000CF82FF5A4**

**inc (call stack) location: 0000000CF82FF5A4**

**j (static) location: 00007FF6CF07E000**

**i location: 0000000CF82FF6A0**

Accending Order:

**char string[] (call stack) location: 0000000CF82FF5A4**

**inc (call stack) location: 0000000CF82FF5A4**

**i location: 0000000CF82FF6A0**

**arr (prgram heap) location: 0000000CF82FF6C8**

**hello function (code) 00007FF6CF07133E**

**next function (code) 00007FF6CF071154**

**j (static) location: 00007FF6CF07E000**

**x (static) location: 00007FF6CF07E200**

**y (code) location: 00007FF6CF07E228**