

1. Consider the difference between statically and dynamically allocated memory.

- How can you tell in code that a value is being stored on the stack?
- When does the memory for a variable on the stack get deleted?
- Similarly, how can you tell in code that a value is being stored on the heap?
- When does memory for a value on the heap get deleted?

A value is stored on the stack when the variable is not initialized with the ``new`` keyword. Values stored on the stack are scoped locally. When the function block is returned, the memory for the variable is automatically deleted.

A value is stored on the heap when the variable is initialized with the ``new`` keyword. Memory for a value on the heap gets deleted when the ``delete`` statement is called manually.

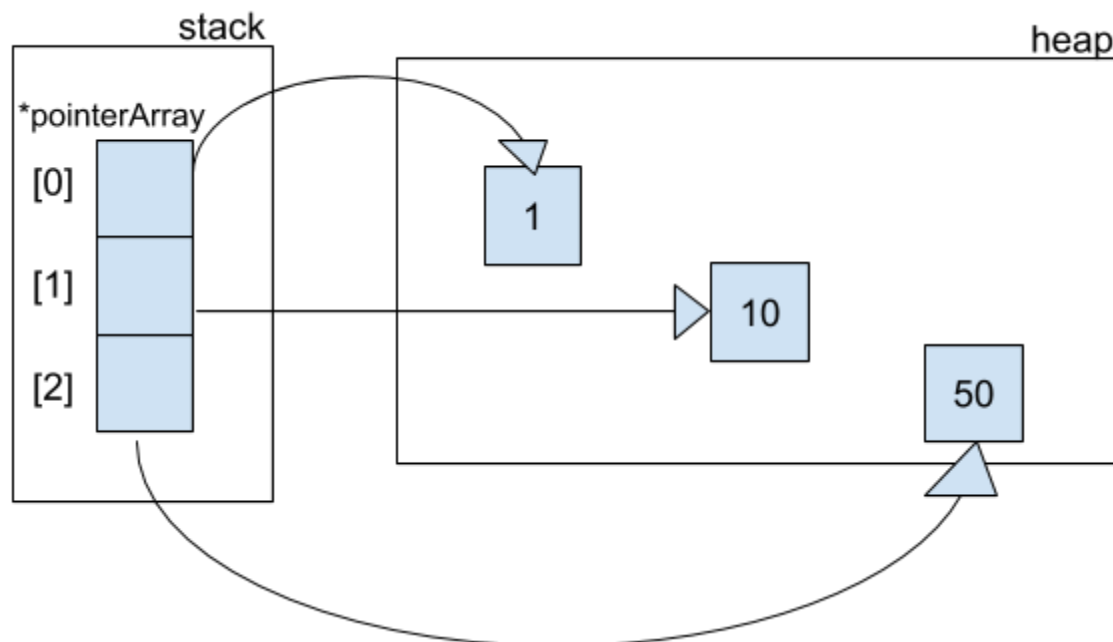
(Answer to Question 2 on next page)

2. Draw a box-and-arrow diagram that illustrates how the array will look after running the following code. Keep in mind that while the array itself will be stored contiguously, boxes allocated for the integers on the heap will be placed wherever there is room (so don't draw them contiguously).

Code:

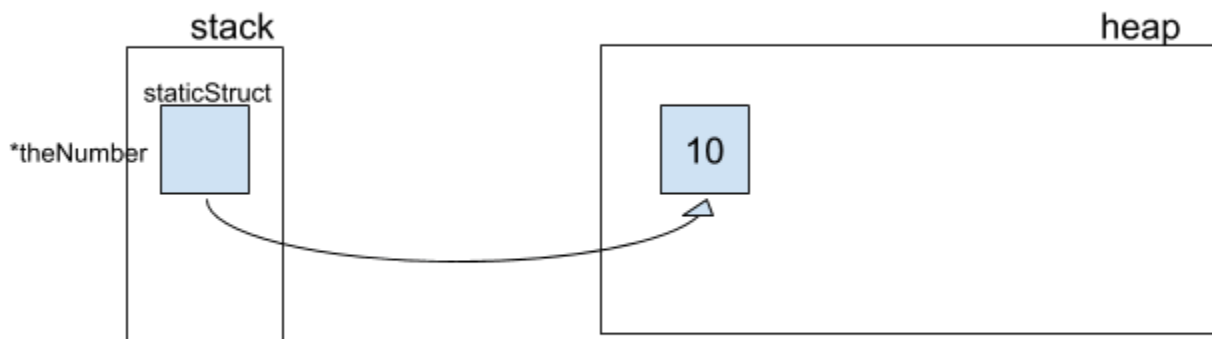
```
int * pointerArray[3];  
  
pointerArray[0] = new int;  
pointerArray[1] = new int;  
pointerArray[2] = new int;  
  
*(pointerArray[0]) = 1;  
*(pointerArray[1]) = 10;  
*(pointerArray[2]) = 50;
```

Box-and-arrow diagram:

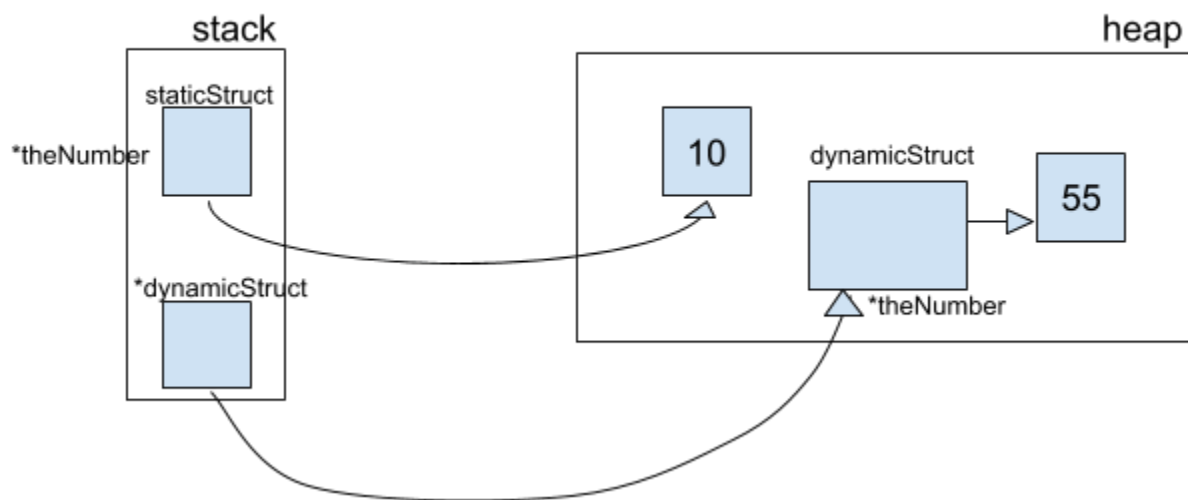


3. Draw a box-and-arrow diagram that illustrates the state of the program's memory after lines 1, 2, and 3.

After line 1:



After line 2:



After line 3:

