Andrew Plum

CS 120-06

Lab #11

11/10/21

Hypotheses and Explanation for Lab #11

**How much I completed:**

I did all of the regular lab assignment; there was no extra credit.

**Hypotheses:**

* a[0][0] = 99
* a[0][1] = 3
* a[0][2] = 55
* a[0][3] = 5
* a[0][4] = 44
* a[0][5] = 6
* a[1][0] = 77
* a[1][1] = 5
* a[1][2] = 7
* a[1][3] = 9
* a[1][4] = 10
* a[1][5] = 12
* a[2][0] = 4
* a[2][1] = 66
* a[2][2] = 10
* a[2][3] = 13
* a[2][4] = 15
* a[2][5] = 18
* Hypothesis of Exact Output:

99, 3, 55, 5, 44, 6

77, 5, 7, 9, 10, 12

4, 66, 10, 13, 15, 18

**Correct Output:**

* a[0][0] = 99
* a[0][1] = 3
* a[0][2] = 4; got wrong, I said it was 55
* a[0][3] = 5
* a[0][4] = 44
* a[0][5] = 6
* a[1][0] = 77
* a[1][1] = 5
* a[1][2] = 7
* a[1][3] = 9
* a[1][4] = 10
* a[1][5] = 12
* a[2][0] = 55; got wrong, I said it was 4
* a[2][1] = 66
* a[2][2] = 10
* a[2][3] = 13
* a[2][4] = 15
* a[2][5] = 18
* Exact

99, 3, 4, 5, 44, 6

77, 5, 7, 9, 10, 12

55, 66, 10, 13, 15, 18

**Explanation:**

* a[0][0] = 99
  + The code that reads, “p = (int \*)a; \*p = 99;” says go to where the array starts in memory which is a[0][0] and assign it the value 99.
* a[0][1] = 3
  + fill() assigns a[0][1] the value 2 because 1\*(1 + 1) = 2, and increment() adds 1 to its value to make it 3.
* a[0][2] = 4; got wrong, I said it was 55
  + a[0][2] should equal 4 because fill() assigns a[0][1] the value 3 because 1\*(1 + 2) = 3, and increment() adds 1 to its value to make it 4. I got this wrong because thought the code that reads “p = (int \*)(a+2); \*p = 55;” said to make two jumps in the second array from pointer a, and assign this address in memory with the value 55.
* a[0][3] = 5
  + fill() assigns a[0][3] the value 4 because 1\*(1 + 3) = 4, and increment() adds 1 to its value to make it 5.
* a[0][4] = 44
  + The code that reads, “p = (int \*)(a[0]+4); \*p = 44;” says make zero jumps in the first array and four jumps in the second array from pointer a which is the memory address at a[0][4] and assign it the value of 44.
* a[0][5] = 6
  + fill() assigns a[0][5] the value 6 because 1\*(1 + 5) = 6.
* a[1][0] = 77
  + The code that reads, “int \*p; p = (int \*)a; p[6] = 77;” says create pointer p and assign it the memory address where array a starts, make six jumps from that memory address and assign the value at this memory address which is a[1][0] the value 77.
* a[1][1] = 5
  + fill() assigns a[1][1] the value 4 because (1 + 1) \* (1 + 1) = 4, and increment() adds 1 to its value to make it 5.
* a[1][2] = 7
  + fill() assigns a[1][2] the value 6 because (1 + 1) \* (1 + 2) = 6, and increment() adds 1 to its value to make it 7.
* a[1][3] = 9
  + fill() assigns a[1][3] the value 8 because (1 + 1) \* (1 + 3) = 8, and increment() adds 1 to its value to make it 9.
* a[1][4] = 10
  + fill() assigns a[1][4] the value 10 because (1 + 1) \* (1 + 4) = 10.
* a[1][5] = 12
  + fill() assigns a[1][5] the value 12 because (1 + 1) \* (1 + 5) = 12.
* a[2][0] = 55; got wrong, I said it was 4
  + a[2][0] should equal 55 because the code that reads, “p = (int \*)(a+2); \*p = 55;” says jump two lengths of the second array from pointer a which is the second jump over of the first array from pointer a, and assign this address in memory with the value 55. I thought the code would just be make two jumps in the second array from pointer a and assign this address in memory with the value 55.
* a[2][1] = 66
  + The code that reads, “p = &a[2][2]; p--; \*p = 66;” says assign the pointer p the memory address at array value a[2][2], make one negative jump in memory addresses, and assign the memory address here the value 66.
* a[2][2] = 10
  + fill() assigns a[2][2] the value 9 because (1 + 2) \* (1 + 2) = 9, and increment() adds 1 to its value to make it 10.
* a[2][3] = 13
  + fill() assigns a[2][3] the value 12 because (1 + 2) \* (1 + 3) = 12, and increment() adds 1 to its value to make it 13.
* a[2][4] = 15
  + fill() assigns a[2][4] the value 15 because (1 + 2) \* (1 + 4) = 15.
* a[2][5] = 18
  + fill() assigns a[2][5] the value 18 because (1 + 2) \* (1 + 5) = 18.