Name: Abdelrahman Hamdy Ahmed

Section: 1 BN: 36

**Pointers.c**

1. Since the name of the pointer represents the address of the first indexed element, the first printf outputted index 0 address for A and B, and garbage for C because it hasn’t been initialized by malloc for dynamic memory allocation or not been given a constant size.

Memory Locations: a = 000000943f3ff940

b = 000002172a781400

c = 0000000000000010

Contents: Un-initialized so garbage

1. C now points to A so when C[0] = 200, A[0] content becomes 200 as well. Due to the for loop, A[1] = 100 + 1, A[2] = 100 + 2, A[3] = 100 + 3

Address of C becomes the same as A and B remains unchanged

Memory Locations: a = 000000943f3ff940

b = 000002172a781400

c = 000000943f3ff940

1. C[1] = 300 makes A[1] = 300 too. \*(c + 2) means C[2] = 301 and 3[c] -> access location C + 3 which means C[3] = 302. Same effect on A because they point to the same location

Memory Locations: a = 000000943f3ff940

b = 000002172a781400

c = 000000943f3ff940

1. C pointer is now moved 1 slot further so it points to C[1]. \*C means C[1] = 400. This only changed A[1] to be 400 while the rest remains the same.

Memory Locations: a = 000000943f3ff940

b = 000002172a781400

c = 000000943f3ff944

1. Casting the c pointer to char\* means that +1 effect will move c ptr by 1 (char) not 4 bytes (int), then it is casted back to int\* and \*c = 500

0001F490 🡪 a[1] = 128144

00000100 🡪 a[2] = 256

Rest is unchanged

Explanation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Content |  | \*c = 500 | |
| 44 | 90 | -> | 44 | 90 |
| 45 | 01 | -> | 45 | F4 |
| 46 | 00 | -> | 46 | 01 |
| 47 | 00 | -> | 47 | 00 |
| 48 | 2D | -> | 48 | 00 |
| 49 | 01 | -> | 49 | 01 |
| 4A | 00 | -> | 4A | 00 |
| 4B | 00 | -> | 4B | 00 |

Memory Locations: a = 000000943f3ff940

b = 000002172a781400

c = 000000943f3ff945

1. Pointer names are printed so address a is still unchanged (000000943f3ff940) and b = (int\*)a + 1 makes b point to a[1] which is the same address of a + 4 bytes = 000000943f3ff944 and finally for C:

A is casted to char\* so +1 effect will move the C pointer by 1 byte (size of char is 1 not 4 bytes like int) then cast back to int\* and the address becomes the same address of a + 1 byte => 000000943f3ff941