

Written Problems

Problem 1

At line 16, the values of A and B are

A = { 'a', 'b', 'c', 'q' }

B = { 'd', 'e', 'f' }

Memory for both A and B is statically allocated. The arguments passed to f are the addresses of A and B. At line three, the program goes to the address of where A[0] is stored in memory at puts 'q' in the third index. In line 4, the variable B (different than the B in main) is dynamically reassigned to point at a new address in memory. Thus, the B in f and the B in main no longer point to the same address. As a result, the values stored in the B in main are not overwritten (see below for program execution).

The issue with f is that it is dynamically allocating memory and not returning the address of such memory. Once the f returns control to main, that memory is lost and the program no longer has means of accessing it. This could cause memory leaks if it happens on too great scale.

state of environment as program executes:

1. main { A --> addr₁ for 8 bytes in memory (which contains the char array ['a', 'b', 'c', 'm']) }
2. main { A --> addr₁ , B --> addr₂ for 6 bytes in memory (which contain the char array ['d', 'e', 'f']) }
3. f { A --> addr₁ , B --> addr₂ }; main { A --> addr₁ , B --> addr₂ }

4. `f { A --> addr1 , A[3] --> 'q' , B --> addr2 }; main { A --> addr1 , B --> addr2 }`
5. `f { A --> addr1 , B --> addr3 for 6 bytes in memory }; main { A --> addr1 , B --> addr2 }`
`}`
6. `f { A --> addr1 , B --> addr3 , B[0] --> 'x' }; main { A --> addr1 , B --> addr2 }`
7. `f { A --> addr1 , B --> addr3 , B[1] --> 'y' }; main { A --> addr1 , B --> addr2 }`
8. `f { A --> addr1 , B --> addr3 , B[2] --> 'z' }; main { A --> addr1 , B --> addr2 }`
9. `main { A --> addr1 (which contains the char array ['a', 'b', 'c', 'q']) , B --> addr2`
`(which contains the char array ['d', 'e', 'f']) }; addr3 (which contains the char array`
`['x', 'y', 'z'])`

State 9 is the final state. Note that the program no longer has access to the memory at `addr3`.