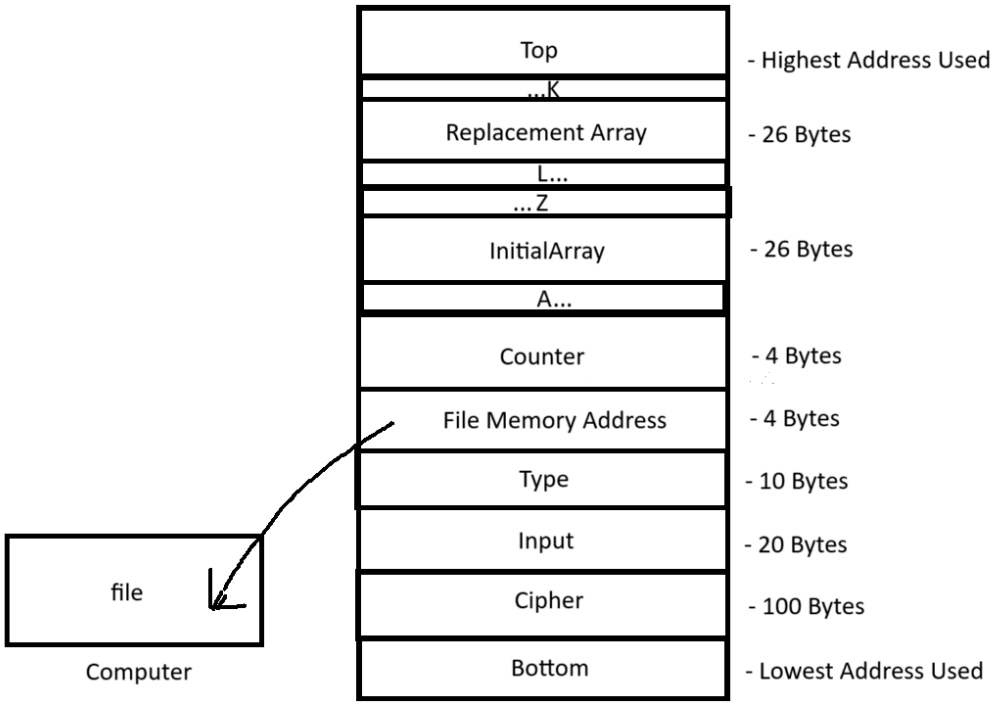
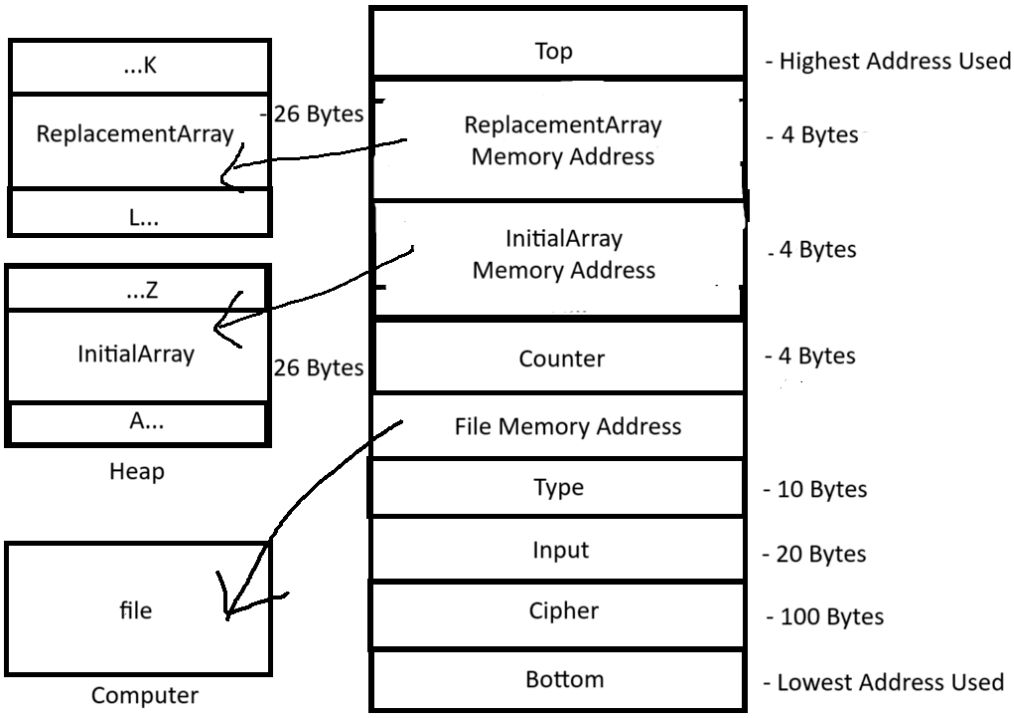
Luke Demi

1. Provide a visual representation of the state of the stack immediately prior to the first program terminating.




2. Provide a visual representation of the state of the stack and the heap immediately prior to the second program terminating.



1. In the problem as written, does storing the ciphers on the heap fundamentally change the way the program executes or provide any significant benefit?
    a. Storing the ciphers on a heap does not fundamentally change the way the program executes in this scenario, since the size of the cipher is known. If the size of the cipher were to be unknown, then the heap would provide a benefit because memory would be allocated more efficiently. But as the program instructions currently define, a heap does not fundamentally change the way the program executes and does not provide any significant benefit.
2. Under what circumstances would storing the ciphers on the heap fundamentally be necessary to ensuring the program works properly and efficiently?
    a. If the size of the cipher is unknown at compile time, it is absolutely essential to use a heap rather than a stack. The heaps dynamic memory allocation allows the C program to properly allocate the needed amount of memory, which a stack is unable to do since information is stored at compile time.
3. Would it ever make sense to store the plain text string on the heap?
    a. If the length of the string is unknown at compile time, then it would make sense to store plain text string on a heap. However, if the length of the string was known then it would not make sense to use a heap, since a stack would be more efficient. To eliminate wasted space, certain string functions can be imported into the C program to properly measure the length of the string, including strlen and sizeof.