

Problem 9.5

5. Consider the following program written in C syntax:

```
void swap(int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}
void main() {
    int value = 2, list[5] = {1, 3, 5, 7, 9};
    swap(value, list[0]);
    swap(list[0], list[1]);
    swap(value, list[value]);
}
```

For each of the following parameter-passing methods, what are all of the values of the variables `value` and `list` after each of the three calls to `swap`?

- Passed by value
- Passed by reference
- Passed by value-result

If passed by value, there is no mechanism for the calls to `swap()` to affect the values stored in the variables in `main()` since `swap()` doesn't return any value (even if it did, it is not used in `main()`).

If passed by reference, the final call passes the address of `value` and `list[1]` and subsequent changes to value within `swap()` do not affect the addresses passed.

If passed by value-result, the results are potentially ambiguous depending on whether the address of `list[value]` is resolved at call-time only or at both call-time and return-time. Call-time only makes the most sense, in which case no alias is formed and the results are the same as pass-by-reference.

	part (a)		part (b)		part (c)	
after execution of	value	list[]	value	list[]	value	list[]
<code>int value=2, list[5]={1,3,5,7,9}</code>	2	{1,3,5,7,9}	2	{1,3,5,7,9}	2	{1,3,5,7,9}
<code>swap(value, list[0]);</code>	2	{1,3,5,7,9}	1	{2,3,5,7,9}	1	{2,3,5,7,9}
<code>swap(list[0], list[1]);</code>	2	{1,3,5,7,9}	1	{3,2,5,7,9}	1	{3,2,5,7,9}
<code>swap(value, list[value]);</code>	2	{1,3,5,7,9}	2	{3,1,5,7,9}	2	{3,1,5,7,9}