CIS 560 Quiz 10

Question 1 (2 pts)
C++ only allows functions to return one object. Without going through the trouble of making a struct to hold multiple kinds of data and returning that struct, how might one define a function so that when it has returned, multiple variables have been assigned values? Your answer should be **one sentence.**
Have the function take in references to objects and assign values to those references in the body of the function.

Question 2 (4 pts)
Describe two situations in which one would want to use dynamic memory allocation (i.e. pointers) instead of allocating space on the stack. Your entire explanation should be **no more than two sentences.**

1. You need the object you're creating to outlive the scope of its allocation and don't want to pass a copy of it outside its allocation scope.
2. You want to allocate a lot of memory, enough that you may run out of stack space.

Question 3 (4 pts)
Explain why in most cases, aside from the ones you described in the previous question, stack-allocated data is preferable to dynamically allocated memory. Your answer should be **no more than two sentences.**
Stack allocated memory is automatically freed once the scope in which the memory was allocated has been left, e.g. once a function terminates, while dynamically-allocated memory must be freed by hand lest you end up with a memory leak. Stack memory is also faster to access and faster to allocate, though these benefits are less pronounced than the benefit of automatic memory management.

Question 4 (4 pts)
What are the benefits to designing a ray tracing algorithm to be iterative instead of recursive? An iterative ray tracer uses a *while loop* to evaluate the path and color output of a ray instead of a recursive call to some tracing function. Provide **two examples in two sentences**.

- Uses far less stack space (only keeps the current ray iteration on the stack instead of all iterations) which may be a concern if you're using many threads to render the scene
- Even more parallelizable; if a ray splits at any point, each path can be evaluated in a different thread
- (Point of interest, not an expected answer) Can be made to run on a GPU, which does not support recursive functions