1) (10 pts) DSN (Dynamic Memory Management in C)

Suppose we have a stack implemented with an array as shown in the structure below. Write a function called grow_stack that will increase the stack's capacity while preserving the exact values currently in the stack and their current locations. Your function should take 2 parameters: a pointer to the current stack and an integer representing the amount to increase the stack's capacity by. **You may not use the realloc function.** You may assume s isn't NULL and pts to a valid struct stack. You may assume that capacity stores the current size of the array that the pointer array is pointing to and that top represents the number of items currently in the stack (items are stored in indexes 0 through top-1).

```
struct Stack {
     int *array;
     int top;
     int capacity;
};

void grow_stack(struct Stack *s, int increase) {

     // Calculates new size as an increase to current capacity
     // 1 point
     int new_size = s->capacity + increase;

     // Has a mechanism to prevent "losing" the current array pointer
     // 2 points
     int *hold = s->array;
     int i;

     // Allocates space for the increased stack array
     // 2 points
     s->array = malloc(sizeof(int) * new_size);

     // Copies values from old array to new array
     // 3 points
     for(i = 0; i < s->top; i++)
               s->array[i] = hold[i];

     // Cleans up old memory space
     // 1 point
     free(hold);

     // Updates capacity
     // 1 point
     s->capacity = new_size;



}
```