Patrick Kupcha (netid kupcha)
CS214, Section 04
Asst1 - ++Malloc

In this assignment we are simulating main memory using an array myblock. In order to aid in the implementation, I used a struct metadata that is used to hold information about the current state of myblock.

Upon the first call of malloc(), myblock is initialized by setting the initial metaData to describe the block. On further calls of malloc(), the block will be fragmented into blocks of memory based on the requested size of memory. Every fragment created will have a metaData block associated with it. Struct metaData holds a char "allocated" to represent whether or not the block has been allocated (malloc()ed), and an int "size" that holds the block size in bytes. In my implementation sizeof(meta) = 8 bytes. I chose to use a char to represent allocated, and chose to not store a pointer to the previous or next blocks in order to reduce the size of the metaData blocks.

So upon initialization of myblock, there would be a single metaData struct with char allocated = '0' and int size = 4088 (4096 bytes total in myblock – sizeof(meta)). If we were to malloc(1), my implementation will then change the first meta block to allocated = '1' and size = 1. This would be followed by another meta block with allocated = '0' and size = 4079.

When free() is called on an allocated memory block, meta→allocated is set to '0'. The function then looks to see first if there is a memory block before and after the current one, depending on the address in myblock. If there is, it checks to see if the previous and/or next block also are unallocated. If they are, then the function will consolidate the allocated blocks into a single block. In the end, there will again be a single metaData block to describe a single unallocated memory block. There will never be two consecutive unallocated memory blocks.

struct metaData: allocated = '0', size = 4088	(unallocated memory block of size 4088)

char* example = malloc(1); →

meta: allocated = '1', size = 1	(allocated memory block of size 1)	meta: allocated = '0', size = 4079	(unallocated memory block of size 4079)

free(example); →

struct metaData: allocated = '0', size = 4088	(unallocated memory block of size 4088)