CSC 209 Review 6 Solution

August 23, 2020

1. I need to create a wrapper function `my_malloc` that does the following:

   - ask `my_malloc` it to allocate `n` bytes
   - call `malloc`
   - test `malloc` doesn't have a null pointer
   - return pointer from `malloc`

   The solution to this problem is:

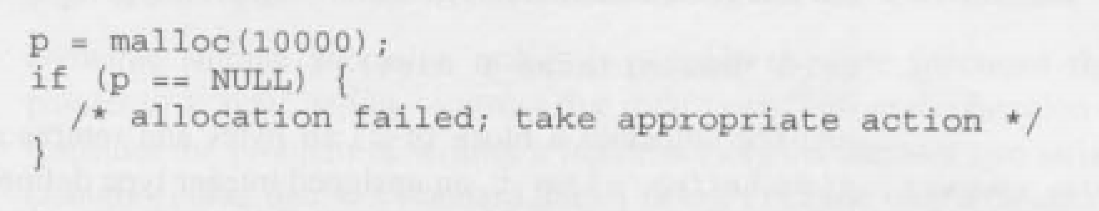
```c
    void *my_malloc(int n) {
        void *p;

        p = malloc(n);

        if (!p) {
            printf("ERROR: Malloc allocation failed");
        }

        return p;
    }
```

   Notes

   - Learned that void function can return value
   - **Dynamic Storage Allocation**
     - Allows to allocate storage during program execution
     - Allows to create data structures and shink and grow array as needed
     - e.g. `malloc`, `calloc`, `realloc`
   - **Memory Allocation Functions**
     - `malloc` - Allocates a block of memory but doesn't initialize it
       * doesn't initialize the allocated memory

    - more efficient than `calloc`
    - accessing the content → segmentation fault (accessing value at invalid mem. location) or garbage values
  - `calloc` - Allocates a block of memory and clears it
    - allocates memory and initializes the memory block to zero
    - accessing the content of blocks would return 0
  - `realloc` - Resizes a previously allocated block of memory
- **Null Pointer**
  - is returned when it fails to allocate a block of memory large enough to satisfy the request

Example



```
p = malloc(10000);
if (p == NULL) {
  /* allocation failed; take appropriate action */
}
```