Assignment 1: ++Malloc

* The program implements custom malloc() and free() library calls for dynamic memory allocation that detects common programming and usage errors utilizing a large array to simulate main memory.
  + It is able to detect and gracefully handle the following exceptions.
    - Freeing addresses that were not allocated by malloc()
    - Freeing pointers that were not allocated by malloc()
    - Redundant freeing of the same pointer
    - Saturation of dynamic memory
  + When these exceptions occur, an error message is created via an informational print statement which includes the calling file name and line number and NULL is returned.
  + To manage the memory, a metadata structure was created to store the following information for each block:
    - Integer size: size of the memory block
    - Integer prev\_size: size of the previous block
    - Integer is\_allocated: Is the block allocated NO: 0 YES 1
    - Integer is\_last: Is this the last block created NO: 0 YES 1
  + When compiling the code there are no errors and warning about the code.
  + A total of 5 workload test cases have been tested, the first three tests are predefined and the last two are described in the testcases.txt file.