```
// ***
// *** You must modify this file
// ***

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "hw08.h"

#ifdef TEST_MAIN
int main(int argc, char * * argv)
{
  int numElem = 0;
  if (argc != 3)
  {
   return EXIT_FAILURE;
  }

  numElem = countVector(argv[1]);

  if(numElem <= 0)
  {
   return EXIT_FAILURE;
  }

  Vector * vecArr = malloc(sizeof( * vecArr) * numElem); //allocates
memory for array


  bool rtv = readVector(argv[1], vecArr, numElem);
  if(rtv == false)
  {
   free(vecArr);
   return EXIT_FAILURE;
  }

  qsort(vecArr, numElem, sizeof(* vecArr), &compareVector);

  rtv = writeVector(argv[2], vecArr, numElem);
  if (rtv == false)
  {
   free(vecArr);
   return EXIT_FAILURE;
  }

  // argv[1]: name of input file (binary)
  // argv[2]: name of output file (binary)

  // check whether there are three arguments.
```

```
  // If not, return EXIT_FAILURE. DO NOT print anything

  // use argv[1] as the input to countVector, save the result

  // if the number of vector is 0 or negative, return EXIT_FAILURE

  // otherwise, allocate memory for an array of vectors

  // read the vectors from the file whose name is argv[1]. save the
  // results in the allocated array
  // if reading fails, release memory and return EXIT_FAILURE



#ifdef DEBUG
  printVector(vecArr, numElem);
#endif


#ifdef DEBUG
  printf("\n");
  printVector(vecArr, numElem);
#endif

  // write the sorted array to the file whose name is argv[2]
  // if writing fails, release memory and return EXIT_FAILURE


  // releave memory, return EXIT_SUCCESS
  free(vecArr);
  return EXIT_SUCCESS;
}
#endif
```