```
// ***
// *** You must modify this file
// ***
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "hw08.h"


#ifdef TEST_COUNTVECTOR
int countVector(char * filename)
{
  FILE *fptr; //file pointer
  int count = 0; //count from fread
  Vector v; //not initialized

  fptr = fopen(filename, "r");
  if (fptr == NULL)
  {
   return -1;
  }

  while(!feof(fptr))
  {
   if(fread(&v, sizeof(Vector), 1, fptr) == 1)
   {
    count++;
   }
  }
  // count the number of vectors in the file and return the number
  // The input is a binary file. You must use fread.
  // You must not use fscanf(, "%d", )
  //
  // If fopen fails, return -1
  //
  //
  // For the mode of fopen, you may use "r" without b
  //
  fclose(fptr);
  return count;
}
#endif

#ifdef TEST_READVECTOR
bool readVector(char* filename, Vector * vecArr, int size)
{
  FILE *fptr; //filep pointer
  int count = 0; //count for number of vectors
  Vector val; //placeholder for fread
```

```
  fptr = fopen(filename, "r");
  if(fptr == NULL)
  {
   return false;
  }

  while(fread(&val, sizeof(Vector), 1, fptr) == 1) //ASK IF FREAD
WORKS LIKE FSCANF
  {
   if(count < size)
   {
    vecArr[count] = val; //writes val to vecArr
   }
   count++;
  }

  fclose(fptr);
  if(count == size)
  {
   return true;
  }
  else
  {
   return false;
  }
  // if fopen fails, return false
  // read Vectors from the file.
  //
  //
  // if the number of integers is different from size (too
  // few or too many) return false
  //
  // if everything is fine, fclose and return true
  return true;
}
#endif

#ifdef TEST_COMPAREVECTOR
int compareVector(const void *p1, const void *p2)
{
  const Vector *ptr1 = (const Vector *) p1;
  const Vector *ptr2 = (const Vector *) p2;

  //compare x component
  if(ptr1 -> x < ptr2 -> x)
  {
   return -1;
  }
  if(ptr1 -> x > ptr2 -> x)
```

```
  {
   return 1;
  }

   if(ptr1 -> y < ptr2 -> y)
   {
    return -1;
   }
   if (ptr1 -> y > ptr2 -> y)
   {
    return 1;
   }

    if(ptr1 -> z < ptr2 -> z)
    {
     return -1;
    }
    if (ptr1 -> z > ptr2 -> z)
    {
     return 1;
    }
 return 0;

  // compare the x attribute first
  // If the first vector's x is less than the second vector's x
  // return -1
  // If the first vector's x is greater than the second vector's x
  // return 1
  // If the two vectors' x is the same, compare the y attribute
  //
  // If the first vector's y is less than the second vector's y
  // return -1
  // If the first vector's y is greater than the second vector's y
  // return 1
  // If the two vectors' y is the same, compare the z attribute
  //
  // If the first vector's z is less than the second vector's z
  // return -1
  // If the first vector's z is greater than the second vector's z
  // return 1
  // If the two vectors' x, y, z are the same (pairwise), return 0
}
#endif

#ifdef TEST_WRITEVECTOR
bool writeVector(char* filename, Vector * vecArr, int size)
{
  FILE *fptr; //file pointer

  fptr = fopen(filename, "w");
```

```
  if (fptr == NULL)
  {
   return false;
  }

  int count = fwrite(&vecArr[0], sizeof(Vector), size, fptr);  //ASK
HOW TO USE FWRITE

  if (count != size)
  {
   return false;
  }
  // if fopen fails, return false
  // write the array to file using fwrite
  // need to check how many have been written
  // if not all are written, fclose and return false
  //
  // fclose and return true
  fclose(fptr);
  return(true);
}
#endif

// This function is provided to you. No need to change
void printVector(Vector * vecArr, int size)
{
  int ind = 0;
  for (ind = 0; ind < size; ind ++)
    {
      printf("%6d %6d %6d\n",
             vecArr[ind].x, vecArr[ind].y, vecArr[ind].z);
    }
}
```