**TESTS USED TO MARK LAB SUBMISSION 3**

These are the tests used to mark your submission for Lab 3:

**Hybrid Sort:**

|  |
| --- |
| void printArray(int arr[], int size) {  for (int i = 0; i < size; i++) {  std::cout << arr[i] << " ";  }  std::cout << std::endl;  }  int main() {  //TEST 1  int Test1[1] =  {6};  int size = sizeof(Test1) / sizeof(Test1[0]);  try {  std::cout << "Given array:  ";  printArray(Test1, size);  hybridSort(Test1, size);  // Test1 = hybridSort(Test1, size);  std::cout << "Array sorted: ";  printArray(Test1, size);  } catch (const std::exception &e) {  std::cerr << "Error for array: ";  printArray(Test1, size);  std::cerr << e.what() << std::endl;  }  //TEST 2  int Test2[8] = {6, 2, 8, 5, 7, 5, 0, 2};  size = sizeof(Test2) / sizeof(Test2[0]);  try {  std::cout << "Given array:  ";  printArray(Test2, size);  hybridSort(Test2, size);  // Test2 = hybridSort(Test2, size);  std::cout << "Array sorted: ";  printArray(Test2, size);  } catch (const std::exception &e) {  std::cerr << "Error for array: ";  printArray(Test2, size);  std::cerr << e.what() << std::endl;  }  //TEST 3  int Test3[8] = {2,2,2,2,2,2,2,2};  size = sizeof(Test3) / sizeof(Test3[0]);  try {  std::cout << "Given array:  ";  printArray(Test3, size);  hybridSort(Test3, size);  // Test3 = hybridSort(Test3, size);  std::cout << "Array sorted: ";  printArray(Test3, size);  } catch (const std::exception &e) {  std::cerr << "Error for array: ";  printArray(Test3, size);  std::cerr << e.what() << std::endl;  }  //TEST 4  int Test4[15] = {4, -7, 2, 1, 0, 2, 4, 2, -3, 3, -3, 7, -2, 7, 7};  size = sizeof(Test4) / sizeof(Test4[0]);  try {  std::cout << "Given array:  ";  printArray(Test4, size);  hybridSort(Test4, size);  // Test4 = hybridSort(Test4, size);  std::cout << "Array sorted: ";  printArray(Test4, size);  } catch (const std::exception &e) {  std::cerr << "Error for array: ";  printArray(Test4, size);  std::cerr << e.what() << std::endl;  }  //TEST 5  int Test5[15] = {-2, -2, -4, -9, -1, -6, -1, -14, -3, -15, -12, -12, -2, -8, -9};  size = sizeof(Test5) / sizeof(Test5[0]);  try {  std::cout << "Given array:  ";  printArray(Test5, size);  hybridSort(Test5, size);  // Test5 = hybridSort(Test5, size);  std::cout << "Array sorted: ";  printArray(Test5, size);  } catch (const std::exception &e) {  std::cerr << "Error for array: ";  printArray(Test5, size);  std::cerr << e.what() << std::endl;  } |

**Mountain Sort:**

The same arrays for Hybrid Sort were used for Mountain Sort.