Lab 2

Jack Jiang

February 24, 2021

Questions

(a) The data necessary to fully specify the Vector type is an int N that represents the length of the Vector and a float pointer to data, float* data, which points to an array of float values that store the components of the vector.

(b) It was stated that memset() can be used to initialize the vectors in the initialize() function instead of a loop because the vector is being initialized to 0's, which memset() can handle. If it were any other value other than 0 and -1, we would not be able to directly use memset() in the initialize() function.

(c) We pass the structures by pointer but not constants because in the C language, passing structures with constants will not actually change the value when the constant is passed into a function as an argument. If a pointer is passed into a function as an argument, the value the pointer is pointing to will change and so we pass structures by pointers and not constants.

(d) In my normalize() function, I handled the zero-vector by including an if statement with the condition that if length was equal to 0, then the function would return 1 and exit. If the zero-vector were passed in, the length would be 0 and we would be dividing the vector components by 0 and dividing by 0 is undefined so we want to make a special case for the zero-vector such that we do not divide by 0.

(e)

```
[jackj21@tc005 code]$ ./test
Test 1: Pass
Test 2: Pass
Test 3: Pass
Test 4: Pass
Test 5: Pass
Test 6: Pass
Test 7: Pass
Test 8: Pass
Test 9: Pass
Test 10: Pass
Test 11: Pass
Test 12: Pass
Test 13: Pass
Test 14: Pass
Test 15: Pass
Test 16: Pass
Test 17: Pass
All Tests: Pass
```

All my tests pass as can seen above and this was due to having correct and safe code. My code properly uses arrays on the heap for vectors, correctly allocating and deallocating memory and checks for certain errors, such as making sure vectors are of same length before having calculations performed on them and that dividing by 0 is not allowed by the program.