Carter Smith – Lab1 (No lab group)

a.

A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering. Include screen shot(s) from Task 1.

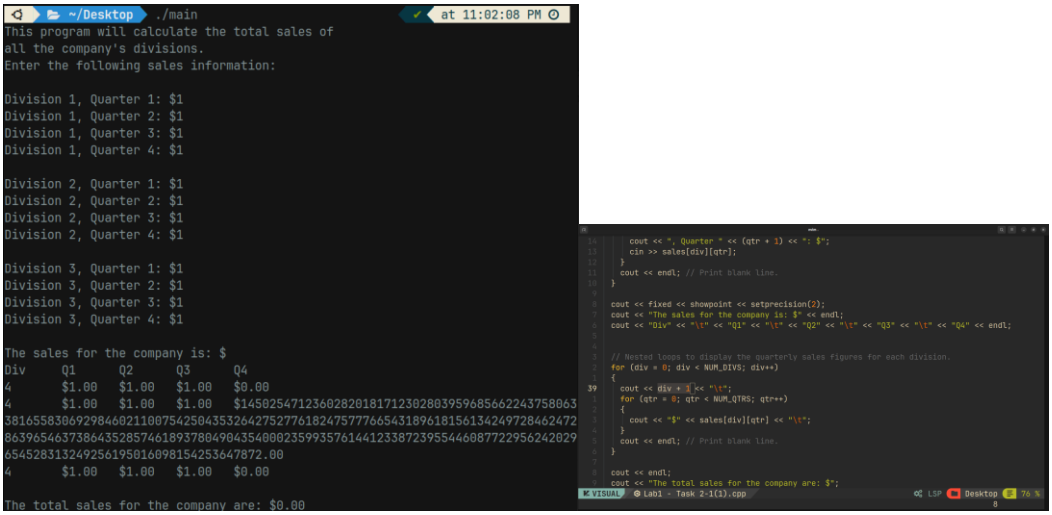
The most important concepts explored in this lab was navigating someone else's code which is having problems. Learning how to fix someone else's mistake by looking over someone's work is a very important skill for engineers as well as cs students. For cs students especially because most jobs upon graduation require maintaining already existing infrastructure and utilizing the debugger and diagnostic strategies will make you a more effective employee.

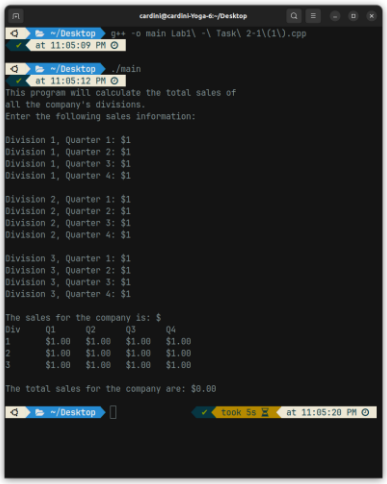
```
Exception thrown: read access violation.
**_Val** was 0xFFFFFFFFFFFFFFFF.

Unhandled exception thrown: read access violation.
**_Val** was 0xFFFFFFFFFFFFFFFF.
```

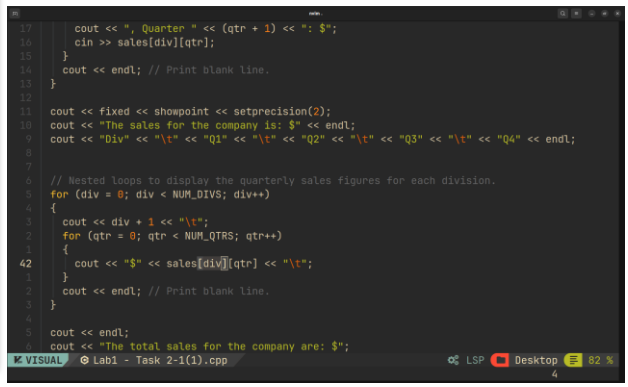
b.

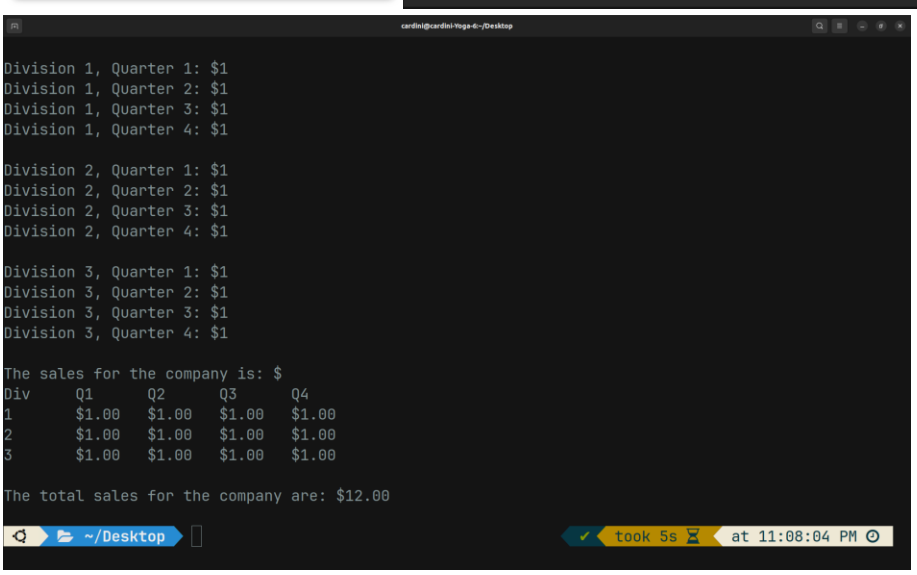
A description of how you approached debugging Task 2, why you think a programmer may have made the mistakes and how you think they can be avoided in the future. Include screen shot(s) from Task 2.

For task 2, the program compiled and appeared to be working until the amount for qtr 4 was messed up. This programmer made very simple mistakes that are easy to make like messing up the indices on a 2d array or forgetting to implement a totalSale's . To avoid these kinds of mistakes in the future, it's important to not be in a hurry and to really understand the work you're doing.


~/Desktop ./main
at 11:02:08 PM
This program will calculate the total sales of
all the company's divisions.
Enter the following sales information:
Division 1, Quarter 1: $1
Division 1, Quarter 2: $1
Division 1, Quarter 3: $1
Division 1, Quarter 4: $1
Division 2, Quarter 1: $1
Division 2, Quarter 2: $1
Division 2, Quarter 3: $1
Division 2, Quarter 4: $1
Division 3, Quarter 1: $1
Division 3, Quarter 2: $1
Division 3, Quarter 3: $1
Division 3, Quarter 4: $1
The sales for the company is: $
Div Q1 Q2 Q3 Q4
4 $1.00 $1.00 $1.00 $0.00
4 $1.00 $1.00 $1.00 $1450254712360282018171230280395968566224375806 3
381655830692984602110075425043532642752776182475777665431896181561342497284624 72
863965463738643528574618937804904354000235993576144123387239554460877229562420 29
654528313249256195016098154253647872.00
4 $1.00 $1.00 $1.00 $0.00
The total sales for the company are: $0.00
cout << ", Quarter " << (qtr + 1) << ": $";
cin >> sales[div][qtr];
}
cout << endl; // Print blank line.
}
cout << fixed << showpoint << setprecision(2);
cout << "The sales for the company is: $" << endl;
cout << "Div" << "\t" << "Q1" << "\t" << "Q2" << "\t" << "Q3" << "\t" << "Q4" << endl;
// Nested loops to display the quarterly sales figures for each division.
for (div = 0; div < NUM_DIVS; div++)
{
cout << div + 1 << "\t";
for (qtr = 0; qtr < NUM_QTRS; qtr++)
{
cout << "$" << sales[div][qtr] << "\t";
}
cout << endl; // Print blank line.
}
cout << endl;
cout << "The total sales for the company are: $";
VISUAL Lab1 - Task 2-1(1).cpp
LSP Desktop 76 %


cardini@cardini-Yoga-6:~/Desktop
~/Desktop g++ -o main Lab1\ -\ Task\ 2-1\(1\).cpp
at 11:05:09 PM
~/Desktop ./main
at 11:05:12 PM
This program will calculate the total sales of
all the company's divisions.
Enter the following sales information:
Division 1, Quarter 1: $1
Division 1, Quarter 2: $1
Division 1, Quarter 3: $1
Division 1, Quarter 4: $1
Division 2, Quarter 1: $1
Division 2, Quarter 2: $1
Division 2, Quarter 3: $1
Division 2, Quarter 4: $1
Division 3, Quarter 1: $1
Division 3, Quarter 2: $1
Division 3, Quarter 3: $1
Division 3, Quarter 4: $1
The sales for the company is: $
Div Q1 Q2 Q3 Q4
1 $1.00 $1.00 $1.00 $1.00
2 $1.00 $1.00 $1.00 $1.00
3 $1.00 $1.00 $1.00 $1.00
The total sales for the company are: $0.00
~/Desktop
took 5s
at 11:05:20 PM


main
cout << ", Quarter " << (qtr + 1) << ": $";
cin >> sales[div][qtr];
}
cout << endl; // Print blank line.
}
cout << fixed << showpoint << setprecision(2);
cout << "The sales for the company is: $" << endl;
cout << "Div" << "\t" << "Q1" << "\t" << "Q2" << "\t" << "Q3" << "\t" << "Q4" << endl;
// Nested loops to display the quarterly sales figures for each division.
for (div = 0; div < NUM_DIVS; div++)
{
cout << div + 1 << "\t";
for (qtr = 0; qtr < NUM_QTRS; qtr++)
{
cout << "$" << sales[div][qtr] << "\t";
}
cout << endl; // Print blank line.
}
cout << endl;
cout << "The total sales for the company are: $";
VISUAL Lab1 - Task 2-1(1).cpp
LSP Desktop 82 %

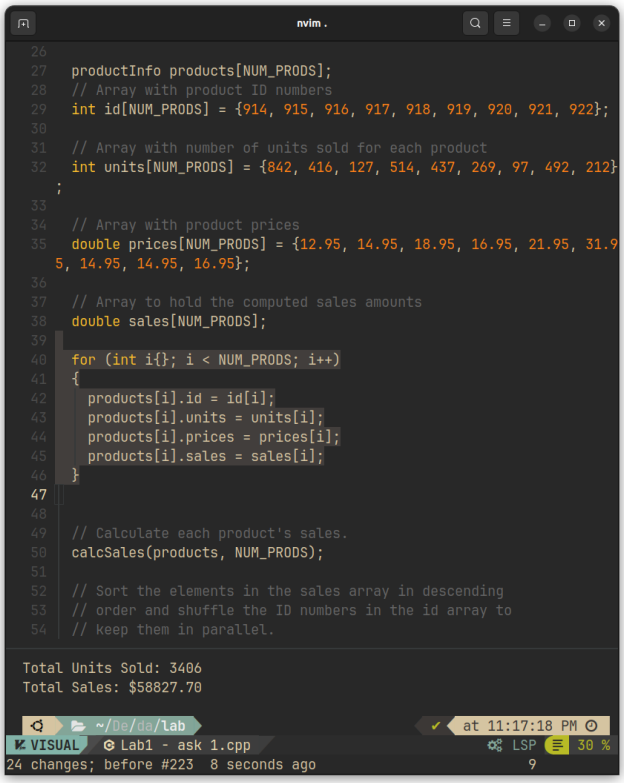

cardini@cardini-Yoga-6:~/Desktop
Division 1, Quarter 1: $1
Division 1, Quarter 2: $1
Division 1, Quarter 3: $1
Division 1, Quarter 4: $1
Division 2, Quarter 1: $1
Division 2, Quarter 2: $1
Division 2, Quarter 3: $1
Division 2, Quarter 4: $1
Division 3, Quarter 1: $1
Division 3, Quarter 2: $1
Division 3, Quarter 3: $1
Division 3, Quarter 4: $1
The sales for the company is: $
Div Q1 Q2 Q3 Q4
1 $1.00 $1.00 $1.00 $1.00
2 $1.00 $1.00 $1.00 $1.00
3 $1.00 $1.00 $1.00 $1.00
The total sales for the company are: $12.00
~/Desktop
took 5s
at 11:08:04 PM

C.

A description of what you had to do in Task 3, including any bugs you may have introduced and had to fix. Include screen shot(s) from Task 3.

To complete task 3 there were several things that needed to be changed. Firstly, I added the struct as required and created the array of structs. Then fill the values in the struct with the values in the arrays in the main function using a for loop. I then had to replace any instance of the old arrays from the code and then compiled and ran the code and it worked.

```
// This function performs a descending order selection sort on *
// the sales array. The elements of the id array are exchanged *
// identically as those of the sales array. size is the number *
// of elements in each array. *
//*****************************************************************

void dualSort(productInfo products[], int size)
{
  int startScan, maxIndex, tempid;
  double maxValue;

  for (startScan = 0; startScan < (size - 1); startScan++)
  {
    maxIndex = startScan;
    maxValue = products[startScan].sales;
    tempid = products[startScan].id;
    for(int index = startScan + 1; index < size; index++)
    {
      if (products[index].sales > maxValue)
      {
        maxValue = products[index].sales;
        tempid = products[index].id;
        maxIndex = index;
      }
    }
    products[maxIndex].sales = products[startScan].sales;
    products[maxIndex].id = products[startScan].id;
    products[startScan].sales = maxValue;
    products[startScan].id = tempid;
  }
}

//*****************************************************************
// Definition of showOrder function. Accepts sales and id arrays *
// as arguments. The size of these arrays is passed into num. *
// The function first displays a heading, then the sorted sales *
```



```
  productInfo products[NUM_PRODS];
  // Array with product ID numbers
  int id[NUM_PRODS] = {914, 915, 916, 917, 918, 919, 920, 921, 922};

  // Array with number of units sold for each product
  int units[NUM_PRODS] = {842, 416, 127, 514, 437, 269, 97, 492, 212};

  // Array with product prices
  double prices[NUM_PRODS] = {12.95, 14.95, 18.95, 16.95, 21.95, 31.95, 14.95, 14.95, 16.95};

  // Array to hold the computed sales amounts
  double sales[NUM_PRODS];

  for (int i{}; i < NUM_PRODS; i++)
  {
    products[i].id = id[i];
    products[i].units = units[i];
    products[i].prices = prices[i];
    products[i].sales = sales[i];
  }


  // Calculate each product's sales.
  calcSales(products, NUM_PRODS);

  // Sort the elements in the sales array in descending
  // order and shuffle the ID numbers in the id array to
  // keep them in parallel.

Total Units Sold: 3406
Total Sales: $58827.70
```

```
// This program produces a sales report for DLC, Inc.
#include <iostream>
#include <iomanip>
using namespace std;

struct productInfo
{
  int id;
  int units;
  double prices;
  double sales;
};

// Function prototypes
void calcSales(productInfo [], int);
void showOrder(const productInfo [], int);
void dualSort(productInfo [], int);
void showTotals(const productInfo [], int);

// NUM_PRODS is the number of products produced
const int NUM_PRODS = 9;


int main()
{

  productInfo products[NUM_PRODS];
  // Array with product ID numbers
  int id[NUM_PRODS] = {914, 915, 916, 917, 918, 919, 920, 921, 922};

  // Array with number of units sold for each product

Total Units Sold: 3406
Total Sales: $58827.70
```

2. Compiled with g++