// File Name: assign2\_tah138.cpp

//

// Author: Trenton Hohle

// Date: 10/4/17

// Assignment Number: 2

// CS 2308.256 Fall 2017

// Instructor: Yijuan Lu

//

// Takes a store's inventory from an outside file, then arranges it

// into a struct array. After that is sorts it.

// The program then offers the user menu options including, (1) viewing

// the entire inventory, (2) searching for an item by sku, (3) searching

// for an item by its name, and (4) quitting the program.

// for options 2 and 3 the user provides input, and the program

// replies accordingly.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

struct Inventory

{

string product;

int sku;

int quantity;

double price;

};

int readTheFile(Inventory[]);

void sortInventory(Inventory[], int);

void displayInventory(Inventory[], int);

void findBySku(Inventory[], int);

void findByName(Inventory[], int);

void printProductInfo(Inventory[], int);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Main: Contains the menu options as well as calls all other functions

//

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main()

{

Inventory inventoryCounts[100];

int size = readTheFile(inventoryCounts);

int answer = -1;

sortInventory(inventoryCounts, size-1);

do

{

cout << "1. Display the inventory" << endl;

cout << "2. Lookup a product by sku" << endl;

cout << "3. Lookup a product by name" << endl;

cout << "4. Quit" << endl;

cout << "Which one do you choose: ";

cin >> answer;

//cout << endl;

//cin.ignore('\n', 100);

switch (answer)

{

case 1:

displayInventory(inventoryCounts, size);

break;

case 2:

findBySku(inventoryCounts, size-1);

break;

case 3:

findByName(inventoryCounts, size-1);

break;

case 4:

break;

default:

cout << "\nThat is not a valid answer try again!\n" << endl;

}

}while(answer != 4);

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// readTheFile: Reads the data file into the struct array

//

// inventoryCounts[]: the struct array the data will be stored in

// returns: the size of the data array

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int readTheFile(Inventory inventoryCounts[])

{

string item;

int sku;

int howMuch;

double value;

int size = 0;

ifstream fin;

fin.open("inventory.dat");

if(fin)

{

while(fin)

{

fin >> item;

fin >> sku;

fin >> howMuch;

fin >> value;

inventoryCounts[size].product = item;

inventoryCounts[size].sku = sku;

inventoryCounts[size].quantity = howMuch;

inventoryCounts[size].price = value;

size++;

}

fin.close();

}

else

{

cout << "Failed to load inventory list!" << endl;

}

return size;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// sortInventory: sorts the inventory using a selection sort

//

// inventoryCounts[]: struct array of inventory data

// size: size of the struct array

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void sortInventory(Inventory inventoryCounts[], int size)

{

int minIndex, minValue, minQuantity;

string minProduct;

double minPrice;

for(int i = 0; i < size-2; i++)

{

minIndex = i;

minValue = inventoryCounts[i].sku;

for(int j = i+1; j < size; j++)

{

if(inventoryCounts[j].sku < minValue)

{

minValue = inventoryCounts[j].sku;

minProduct = inventoryCounts[j].product;

minQuantity = inventoryCounts[j].quantity;

minPrice = inventoryCounts[j].price;

minIndex = j;

}

}

inventoryCounts[minIndex].sku = inventoryCounts[i].sku;

inventoryCounts[minIndex].product = inventoryCounts[i].product;

inventoryCounts[minIndex].quantity = inventoryCounts[i].quantity;

inventoryCounts[minIndex].price = inventoryCounts[i].price;

inventoryCounts[i].sku = minValue;

inventoryCounts[i].product = minProduct;

inventoryCounts[i].quantity = minQuantity;

inventoryCounts[i].price = minPrice;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// displayInventory: displays the inventory in labeled columns

//

// inventoryCounts[]: struct array of inventory data

// size: size of the struct array

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void displayInventory(Inventory inventoryCounts[], int size)

{

cout << "Sku" << setw(12) << "Quantity" << setw(8) << "Price"

<< setw(17) << "Product" << endl;

for(int i = 0; i<size-1; i++)

{

cout << inventoryCounts[i].sku << setw(6) << " "

<< inventoryCounts[i].quantity << setw(7) << right << fixed

<< setprecision(2) << "$" << inventoryCounts[i].price

<<setw(20) << inventoryCounts[i].product << endl;

}

cout << "\n" << endl;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// findBySku: locations if possible a product by sku, then outputs accordingly

//

// inventoryCounts[]: struct array of inventory data

// size: size of the struct array

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void findBySku(Inventory inventoryCounts[], int size)

{

int finding;

bool found = false;

int location;

int first = 0;

int last = size;

int middle = (first+last)/2;

cout << "What sku are you looking for: ";

cin >> finding;

cout << endl;

while(first<=last && !found)

{

if(finding == inventoryCounts[middle].sku)

{

location = middle;

found = true;

}

else if (finding < inventoryCounts[middle].sku)

{

last = middle-1;

middle = (first+last)/2;

}

else

{

first = middle + 1;

middle = (first+last)/2;

}

}

if(!found)

cout << "The sku you seek is not here!\n" << endl;

else

{

printProductInfo(inventoryCounts, location);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// findByName: receives an input product name, checks inventory data,

// replies accordingly

//

// inventoryCounts[]: struct array of inventory data

// size: size of the struct array

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void findByName(Inventory inventoryCounts[], int size)

{

string product;

bool found = false;

int location = -1;

cout << "What product are you looking for: ";

cin >> product;

cout << endl;

for(int i = 0; i <size; i++)

{

if(product == inventoryCounts[i].product)

{

found = true;

location = i;

}

}

if(found)

printProductInfo(inventoryCounts, location);

else

cout << product << " was not found.\n" << endl;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// printProductInfo: If product found in other functions, this function

// outputs all the data on the product

//

// inventoryCounts[]: struct array of inventory data

// location: the spot in the array where the product is located

// returns: null

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void printProductInfo(Inventory inventoryCounts[], int location)

{

cout << "product name: " << inventoryCounts[location].product << endl;

cout << "sku: " << inventoryCounts[location].sku << endl;

cout << "quantity: " << inventoryCounts[location].quantity << endl;

cout << "price: $" << inventoryCounts[location].price << endl;

cout << endl;

}