/\*

Michael Dobachesky

SE 124.12

Program 6

PURPOSE:

Using good interactive programming techniques, prepare a program for use by the ABC TRAVEL AGENCY

which will ask the user to enter and process information about the number of passengers who board

a flight for a particular airline.

VARIABLE DICTIONARY:

REPRESENTS TYPE VARIABLE

run reply char run\_reply

airline flight number int airline\_flight\_number

charter number int charter\_number

index\_number int index\_number

maximum passengers int max\_passengers

minimum passengers int min\_passengers

passenger load int passenger\_load

subscript number int subscript\_number

run data sorting loop int swap\_data

total passengers int total\_passengers

airline name string airline\_name

destination location string destination\_location

duplicate data string duplicate\_data

\*/

#include <iostream>

#include <string>

using namespace std;

char run\_reply;

int airline\_flight\_number[100];

int charter\_number;

int index\_number[100];

int max\_passengers;

int min\_passengers;

int passenger\_load[100];

int subscript\_number;

int swap\_data;

int total\_passengers;

string airline\_name[100];

string destination\_location[100];

string duplicate\_data[100];

void setup\_program();

void load\_arrays();

void display\_by\_airline();

void display\_by\_destination();

void sort\_data();

int main()

{

cout << "Do you want to run the ABC Travel Agency program? (Y/N) ";

cin >> run\_reply;

run\_reply = toupper(run\_reply);

while (run\_reply != 'Y' && run\_reply != 'N')

{

cout << "Invalid response" << endl;

cout << "Please enter either a Y or N: ";

cin >> run\_reply;

run\_reply = toupper(run\_reply);

}

system("cls");

if (run\_reply == 'Y')

{

setup\_program();

load\_arrays();

display\_by\_airline();

display\_by\_destination();

}

}

void setup\_program()

{

charter\_number = 0;

min\_passengers = 50;

max\_passengers = 275;

subscript\_number = -1;

total\_passengers = 0;

}

void load\_arrays()

{

while (run\_reply == 'Y')

{

cout << "ABC Travel Agency" << endl << endl;

subscript\_number = subscript\_number + 1;

charter\_number = charter\_number + 1;

cout << "Charter Number: " << charter\_number << endl << endl;

cout << "Airline Name: ";

cin >> airline\_name[subscript\_number];

cout << "Airline Flight Number: ";

cin >> airline\_flight\_number[subscript\_number];

cout << "Destination: ";

cin >> destination\_location[subscript\_number];

cout << "Passenger Load: ";

cin >> passenger\_load[subscript\_number];

while (passenger\_load[subscript\_number] < min\_passengers || passenger\_load[subscript\_number] > max\_passengers)

{

cout << "Invalid response " << endl;

cout << "Passenger load must be between " << min\_passengers << " and " << max\_passengers << endl;

cout << "Passenger load: ";

cin >> passenger\_load[subscript\_number];

}

total\_passengers = total\_passengers + passenger\_load[subscript\_number];

cout << endl;

cout << "Do you want to add another charter flight? (Y/N) ";

cin >> run\_reply;

run\_reply = toupper(run\_reply);

while (run\_reply != 'Y' && run\_reply != 'N')

{

cout << "Invalid response " << endl;

cout << "Please enter either a Y or N: ";

cin >> run\_reply;

run\_reply = toupper(run\_reply);

}

system("cls");

}

}

void display\_by\_airline()

{

cout << "ABC Travel Agency" << endl;

cout << "Charter Passenger Load Report" << endl;

cout << "(Sorted by Airline Name)" << endl << endl;

cout << "Airline Name" << " " << "Flight Number" << " " << "Destination" << " " << "Passenger Load" << endl << endl;

for (subscript\_number = 0; subscript\_number < charter\_number; subscript\_number++)

{

duplicate\_data[subscript\_number] = airline\_name[subscript\_number];

}

sort\_data();

for (subscript\_number = 0; subscript\_number < charter\_number; subscript\_number++)

{

cout << airline\_name[index\_number[subscript\_number]] << " " << airline\_flight\_number[index\_number[subscript\_number]] << " " << destination\_location[index\_number[subscript\_number]] << " " << passenger\_load[index\_number[subscript\_number]] << endl;

}

cout << endl;

cout << "Total Passengers: " << total\_passengers << endl << endl;

cout << "End of Passenger Report " << endl << endl;

system("pause");

system("cls");

}

void display\_by\_destination()

{

cout << "ABC Travel Agency" << endl;

cout << "Charter Passenger Load Report" << endl;

cout << "(Sorted by Destination)" << endl << endl;

cout << "Airline Name" << " " << "Flight Number" << " " << "Destination" << " " << "Passenger Load" << endl << endl;

for (subscript\_number = 0; subscript\_number < charter\_number; subscript\_number++)

{

duplicate\_data[subscript\_number] = destination\_location[subscript\_number];

}

sort\_data();

for (subscript\_number = 0; subscript\_number < charter\_number; subscript\_number++)

{

cout << airline\_name[index\_number[subscript\_number]] << " " << airline\_flight\_number[index\_number[subscript\_number]] << " " << destination\_location[index\_number[subscript\_number]] << " " << passenger\_load[index\_number[subscript\_number]] << endl;

}

cout << endl;

cout << "Total Passengers: " << total\_passengers << endl << endl;

cout << "End of Passenger Report " << endl << endl;

system("pause");

}

void sort\_data()

{

for (subscript\_number = 0; subscript\_number < charter\_number; subscript\_number++)

{

index\_number[subscript\_number] = subscript\_number;

}

swap\_data = 'Y';

while (swap\_data == 'Y')

{

swap\_data = 'N';

for (subscript\_number = 0; subscript\_number < charter\_number - 1; subscript\_number++)

{

if (duplicate\_data[subscript\_number] > duplicate\_data[subscript\_number + 1])

{

swap (duplicate\_data[subscript\_number], duplicate\_data[subscript\_number + 1]);

swap (index\_number[subscript\_number], index\_number[subscript\_number + 1]);

swap\_data = 'Y';

}

}

}

}