Practice Exam 2:

I. Debugging The following program has one compile time and one logical run-time error. Find and fix both. The program is supposed to find the words in a vector that start with a certain letter and print them to the screen. Feel free to check the code by running it inside your development environment just as you would if you were working on a lab.

1.

```
#include <iostream>
#include <vector>

using namespace std;



vector<string> filter(const vector<string>, char);


int main(){
  vector<string> wikipedia = {"alphabet", "babble", "constant", "decoding", "definition", "encryption", "encoding", "femto", "giraffe", "hankerchief"};


  vector<string> filtered_wiki = filter(wikipedia, 'd');


  for(string str : filtered_wiki){
    cout << str << endl;
  }

  cout << endl;

  return 0;
}



vector<string> filter(const vector<string> v, char c){
  vector<string> result;
  int i = 0;

  for(string str : v){
    if(test(v[i], c)){
      result.push_back(v[i]);
    }
    i++;
  }

  return result;
}


bool test(string str, char c){
  return str[0] != c;
}
```

II. **Use Existing Code:** Using the functions already present in the program, write code inside of the main function that creates a new deck of cards, shuffles that deck, and then deals each player a new hand of cards. Then, at the end of the program, display the cards remaining in the deck as well as each player's hand of cards.

2.

```
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

const int NUMBER_OF_CARDS = 52;

void show(const vector<vector<string>>&);
void shuffleDeck(vector<vector<string>>&);
vector <vector<string>> createDeck();
vector <string> draw(vector<vector<string>>&);

void deal(vector<vector<string>>&,
          vector<vector<string>>&,
          vector<vector<string>>&);


int main(){
  vector<vector<string>> player1_hand;
  vector<vector<string>> player2_hand;



  return 0;
}
```

```
vector<vector<string>> createDeck(){

  vector<vector<string>> deck = {
    {"A", "hearts"}, {"A", "spades"}, {"A", "clovers"}, {"A", "diamonds"},
    {"2", "hearts"}, {"2", "spades"}, {"2", "clovers"}, {"2", "diamonds"},
    {"3", "hearts"}, {"3", "spades"}, {"3", "clovers"}, {"3", "diamonds"},
    {"4", "hearts"}, {"4", "spades"}, {"4", "clovers"}, {"4", "diamonds"},
    {"5", "hearts"}, {"5", "spades"}, {"5", "clovers"}, {"5", "diamonds"},
    {"6", "hearts"}, {"6", "spades"}, {"6", "clovers"}, {"6", "diamonds"},
    {"7", "hearts"}, {"7", "spades"}, {"7", "clovers"}, {"7", "diamonds"},
    {"8", "hearts"}, {"8", "spades"}, {"8", "clovers"}, {"8", "diamonds"},
    {"9", "hearts"}, {"9", "spades"}, {"9", "clovers"}, {"9", "diamonds"},
    {"10", "hearts"}, {"10", "spades"}, {"10", "clovers"}, {"10", "diamonds"},
    {"J", "hearts"}, {"J", "spades"}, {"J", "clovers"}, {"J", "diamonds"},
    {"Q", "hearts"}, {"Q", "spades"}, {"Q", "clovers"}, {"Q", "diamonds"},
    {"K", "hearts"}, {"K", "spades"}, {"K", "clovers"}, {"K", "diamonds"}
  };

  return deck;
}


void shuffleDeck(vector<vector<string>> &deck){

  int index1;
  int index2;
  vector <string> temp;

  srand(time(0));

  for(int i = 0; i < 1000; i++){
    index1 = rand() % deck.size();
    index2 = rand() % deck.size();

    temp = deck.at(index1);
    deck.at(index1) = deck.at(index2);
    deck.at(index2) = temp;
  }

}



void show(const vector<vector<string>> &cards){

  cout << endl;

  for(vector<string> card : cards){
    for(string value : card){
      cout << value << " ";
    }
    cout << endl;
  }

  cout << endl;
}


vector<string> draw(vector<vector<string>> &deck){

  vector<string> card = deck.at(deck.size() - 1);
  deck.pop_back();

  return card;
}


void deal(vector<vector<string>> &hand1,
          vector<vector<string>> &hand2,
          vector<vector<string>> &deck){

  for(int i = 0; i < 5; i++){
    hand1.push_back(draw(deck));
    hand2.push_back(draw(deck));
  }
}
```

III. Tiny Programs

3. Write a function that counts how many x’s and X’s are in an array and returns the count.

4. Write a function that returns true if an array contains the integer 42 and false if it does not

5. Write a function that multiples every element of a two dimensional array by 3.

6. Write a function that takes a two dimensional array of integers as input and returns a vector with the averages for each column.

7. Write a function that finds the largest value in a two dimensional array of doubles and returns that value.

IV. Write Program

8.

You are in charge of writing a program for a movie theatre that sells tickets and books seats for customers. The program will have two theatres each with seating for 5 rows and 10 columns. The program will load the state of either theatre from one of two files, one named theatre1.txt and one named theatre2.txt.

The program will ask the user which theatre they want to buy a ticket for. After the user chooses between theatre1 and theatre 2, the appropriate theatre's state will be loaded from the file into a two-dimensional array. Each theatre will be represented as columns and rows of X's and 0's where X represents a seat that is taken and 0 represents a seat that is available.

For example, theatre1.txt may look like this:

```
0 0 0 0 X X 0 0 X X
X X X X X X 0 0 0 X
0 0 0 0 0 X X X 0 0
X 0 X 0 X 0 X 0 X X
0 0 0 X X X X X 0 0
```

The program will display a console print out of the seats that are taken and empty after the two-dimensional array is generated. Empty seats in the array will be represented by a 0 and purchased seats will be represented by an X.

The user will select a seat by entering the row and column of the seat they desire. If the seat is taken the program will inform them that that seat is not available. If the seat is out of range, the program will instruct them that their seat choice does not exist. If the seat they choose is empty, the program will say confirmed! and fill that seat of the theatre with an X (by modifying the two-dimensional array). The program will then ask the user if they would like another seat. If they say 'yes', they will be asked to choose another, if they say no, the program will say goodbye and exit.

Before the program exits, it will write the state of the theatre back to the same theatre file and print a receipt of the coordinates of each seat the customer purchased to the console. You should keep track of each seat purchase's coordinates by storing that information in a vector containing strings. For instance, if a user purchases three seats, the vector should contain data similar to this:

{"Row 3 Col 5", "Row 3 Col 6", "Row 3 Col 7"}

And the receipt print-out at the end might look like this:

Thank you for your purchase! Your seats are:

Seat 1: Row 3 Col 5
Seat 2: Row 3 Col 6
Seat 3: Row 3 Col 7

Make sure you close the theatre file after reading from it and before re-opening it for writing to avoid an error.

Your program should contain the following functions:

bool isSeatAvailable(int row, int column, const char theatre[][10]) – returns true if the seat at that location has a '0' and false otherwise

bool bookSeat(int row, int column, char[][10]) – attempts to book a seat at that row and column. It returns true if successful and false if not successful for any reason.

void loadTheatre(char theatre[][10], string filename) – populates the given array with data from the given file.

void saveTheatre(char theatre[][10], string filename) - saves the given array to the given file name.

Write Program here:

(continued)