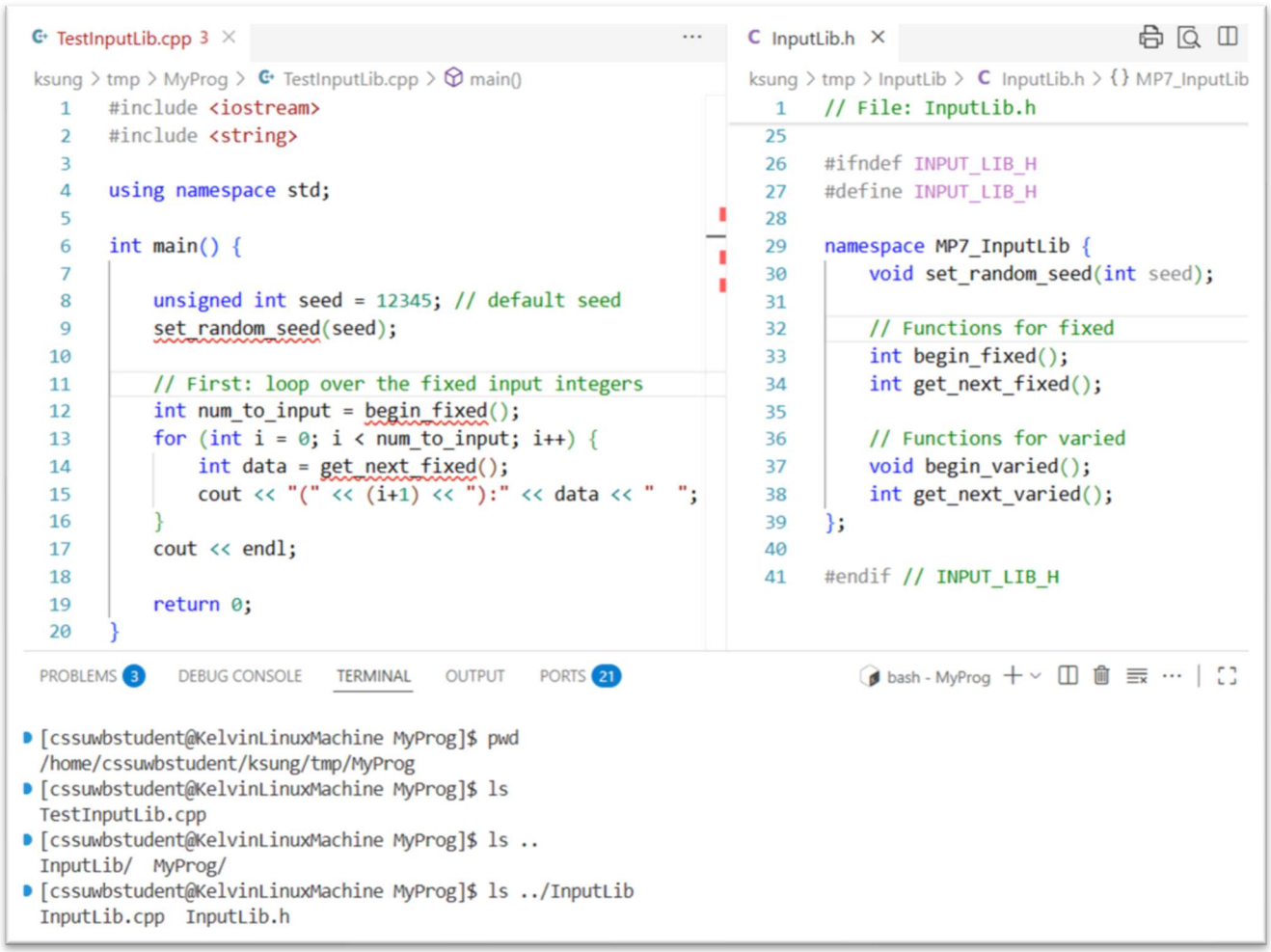
Quiz 7 Name:_____

Question 1. (10pt) Here is my working session of an attempt to investigate MP7 InputLib:



In the bottom Terminal window, notice the commands I have issued and the corresponding output. Make sure you understand the relationships between the folders, output of the pwd (print-working-directory) command in the Terminal window, and the name of the folder that contains my source code file, `TestInputLib.cpp`.

a. (5pt) In the above source code file, `TestInputLib.cpp`, show the changes you have to make in order for the file to compile.

Two things: 1) #include "../InputLib/InputLib.h"
2) qualify with namespace: **MP7_InputLib::** for the functions:
set_random_seed(), begin_fixed(), and get_next_fixed()

b. (2pt) Please list all the commands, one per line, that you will issue in the Terminal window in order to compile the library.

Staying in MyProg:

```
g++ -c ../InputLib/InputLib.cpp -o ../InputLib/InputLib.o
ar -rcs ../InputLib/Input.a ../InputLib/InputLib.o
```

cd over:

```
cd ../InputLib
g++ -c InputLib.cpp -o InputLib.o
ar -rcs Input.a InputLib.o  (this step is actually optional)
cd ../MyProg
```

c. (3pt) Continue from (b), please list all the commands, one per line, that you will issue in the Terminal window to compile your source code file into an object file, and then, link with your library from (b) into a program named, `InputTest`.

```
g++ -c TestInputLib.cpp -o TestInputLib.o
g++   TestInputLib.o        ../InputLib/InputLib.a          -o InputTest
```

Question 2. (20pt) Given:

```
struct Node {
    int data;
    Node* next;
};
```

And that,

```
Node *header = new Node(-1, nullptr);  // dummy header
…
// code for inserting/remove nodes from the linked list
```

a. (4pt) Now, I want to print out the content of the list with this function call:

```
PrintList(header);  // remember header points to a dummy header
```

Please show, with clear and detailed parameter declaration, the implementation of `PrintList()` function. Note, the data-filed of the dummy header should be ignored.

```
void PrintList(Node *dummyHeader) {
    Node *current = dummyHeader->next; // Start from the first real node
    while (current != nullptr) {
        cout << current->data << " ";
        current = current->next;
    }
    std::cout << endl;
}
```

b. (6pt) Based on the above list, I want to define a function to search the list to find the node for a given number, n. If the number is not found I want the pointer to the very last node in the list. For example, if my list contains 3 nodes with data: 1, 7, 5.

```
Node *node = FindNode(header, 7); // returns the pointer to the node that contains 7
```

While,

```
Node *node = FindNode(header, 11); // returns the pointer to the node that contains 5
        // since 11 is not in the list, return pointer is the last node
        // In this case, searching for 5 or not found will return the same pointer
```

Please show, with clear and detailed parameter declaration, how you would implement the `FindNode()` function.

```
Node* FindNode(Node *dummyHeader, int target) {
    Node* current = dummyHeader; // Start from the first real node
    while (current->next != nullptr) {
        if (current->next->data == target) {
            return current->next; // Node found
        }
        current = current->next;
    }
    return current; // Node not found
}
```

c. (4pt) After the above function call, show how you would allocate memory for a new Node with the value searched, n (assuming properly initialized), and insert the new node into the list after the returned node pointer.

```
Node *node = FindNode(header, n);
Node *newNode = new Node(n, nullptr);
newNode->next = node->next;
node->next = newNode;
```

d. (6pt) Assuming all Nodes in the list are dynamically allocated. Show your implementation, with clear and detailed parameter declaration, of a destroyList() function to release all of the memory allocated for the list.

```
void DestroyList(Node *dummyHeader) {
    Node *current = dummyHeader;
    while (current != nullptr) {
        Node *temp = current;
        current = current->next;
        delete temp;
    }
}
```