Grayson Koy – n10480811

cab301 assignment report

Tool library system

Contents

[Introduction 3](#_Toc72692452)

[Design and Analysis of Algorithms 4](#_Toc72692453)

[Top Three algorithm 4](#_Toc72692454)

[Introduction 4](#_Toc72692455)

[Pseudocode 4](#_Toc72692456)

[Algorithm description 4](#_Toc72692457)

[Efficiency analysis 5](#_Toc72692458)

[Software Test Plans and Test Results 6](#_Toc72692459)

[Unit Tests 6](#_Toc72692460)

[Tests of functionality 6](#_Toc72692461)

[Appendix 9](#_Toc72692462)

[Test images 9](#_Toc72692463)

[Staff login 9](#_Toc72692464)

[Staff failed login 9](#_Toc72692465)

[Member login 9](#_Toc72692466)

[Member login wrong username 9](#_Toc72692467)

[Member login wrong PIN 9](#_Toc72692468)

[Adding a new tool 10](#_Toc72692469)

[Adding tool stock 10](#_Toc72692470)

[Adding a negative amount of tool stock 10](#_Toc72692471)

[Removing tool stock 10](#_Toc72692472)

[Removing a negative amount of tool stock 11](#_Toc72692473)

[Removing too much tool stock 11](#_Toc72692474)

[Member registration 11](#_Toc72692475)

[Member removal 11](#_Toc72692476)

[Finding member contact phone number 11](#_Toc72692477)

[Finding member contact number with non-existent name 11](#_Toc72692478)

[Going back to main menu 11](#_Toc72692479)

[Display tools by category 12](#_Toc72692480)

[Borrow tool from library 12](#_Toc72692481)

[Trying to borrow a tool when 3 are already borrowed 12](#_Toc72692482)

[Borrowing the same tool more than once 12](#_Toc72692483)

[Return tool to library 12](#_Toc72692484)

[List borrowed tools 12](#_Toc72692485)

[List borrowed tools with no borrowed tools 12](#_Toc72692486)

[Display most frequently borrowed tools 13](#_Toc72692487)

[Display most frequently borrowed tools when no tools have been borrowed 13](#_Toc72692488)

[Display most frequently borrowed tools when only 2 tools have been borrowed 13](#_Toc72692489)

[Incorrect page is selected in menu 13](#_Toc72692490)

[Incorrect entry in list selector is selected 13](#_Toc72692491)

[Entering a string into an int input 13](#_Toc72692492)

[Entering a mobile number with letters when registering 13](#_Toc72692493)

[Entering a PIN of incorrect length when registering 13](#_Toc72692494)

# Introduction

For this assignment, a software system for managing a tool library system was built. This system allows for a number of different types of tools to be stored, and lets members of the library keep track of their borrowed tools digitally. There is also a section in the program for staff of the library to manage the contents of the library, for example by adding new tools or modifying the stock of existing tools.

The implementation of the system into a C# program was fairly simple, and there were no major problems in implementing the required functionality. However, some restrictions due to the interfaces which the program had to utilise were limiting, as these interfaces could not be modified. This was an issue since some of them did not contain all of the required functionality, most notably the ToolLibrarySystem interface, which did not contain some essential functions which were needed to complete some of the requested functionality. These issues could be circumvented by creating a database class which holds the program’s data. This meant that the program could interact with the data to complete any functionality which was outside the scope of the given ToolLibrarySystem interface.

In the future, with the ability to modify the interfaces, a better solution than to expose the program’s data as public would be to fully flesh out the ToolLibrarySystem interface and class with more methods for accessing and manipulating the data in the system. This would remove the need to do so directly from the program’s code, ensuring the data cannot be incorrectly manipulated.

This report was written as a technical description of the program, providing extra information on algorithms and testing. The report will go through the key algorithms used in the program, with an explanation and justification of each algorithm’s design, a pseudocode version of each algorithm, and an analysis of each algorithm’s efficiency. It will also contain a test plan for the program, which will go over and validate all the functionality of the program.

# Design and Analysis of Algorithms

## Top Three algorithm

### Introduction

An algorithm was required to retrieve the top three most-borrowed tools in the system. The algorithm chosen was one which loops through the tools once and gets the most-borrowed tools while looping through. This was chosen rather than a sorting algorithm as it is a more efficient solution, since most sorting algorithms would have to sort the whole array before the top 3 most-borrowed tools could be selected. However, if the heap sort algorithm was used the sorting could have stopped after the three highest values were sorted. There were no challenges implementing the chosen algorithm as it was quite simple, and there are no bugs present.

### Pseudocode

|  |
| --- |
| **ALGORITHM** TopThree(*A*[0..*n* – 1])  // Given an array A containing the amount of times each tool has been borrowed,  // returns the top three most-borrowed tools.  *topBorrowed* ← []  **for** *i* ← 0 **to** *n* – 1 **do**  **for** *j* ← 0 **to** 3 **do**  **if not** topBorrowed[*j*] **or** A[*i*] > topBorrowed[*j*] **do**  **for** *x* ← 2**to** j **do**  *topBorrowed*[*x* + 1] ← topBorrowed[*x*]  *topBorrowed*[*j*] ← A[*i*]  **break**  **return** *topBorrowed* |

### Algorithm description

This algorithm takes in an array *A* containing the number of borrows for each tool in the system. The algorithm then creates an array *topBorrowed* which will store the top three most-borrowed tools. Then it loops through each tool in the array *A* and iterates three times to check against each place in the *topBorrowed* array. Then, it checks if either there’s nothing in the current place, or if the times the tool has been borrowed is greater than the tool in the *topBorrowed* array at the given place. If one of these things is the case, this tool will take the current place. To do this, the tools in the *topBorrowed* array from the current place to the end of the array are shifted down one place to make room for the new value. Then, the new value is inserted at the current place. The loop then breaks, since a place has been found for the tool. This repeats until all of the tools have been looped through, and then the *topBorrowed* array is returned.

### Efficiency analysis

#### Time complexity

Since the algorithm only loops through the tools once, it has a time complexity of **O(n)**. The basic operation in the algorithm is the comparison between *A*[*i*] and *topBorrowed*[*j*]. The worst case scenario for this algorithm would be when every tool has the same or a lower borrowed amount than the last.

This would result in the basic operation not being called for the first tool, since the first element is null, the operation being called once for the second tool, since the second element is null, twice for the third tool since the third is null, and three times for every subsequent tool. Therefore, the time complexity of the algorithm is **t(n) = 1 + 2 + 3 \* ((n – 1) - 2)**, which simplifies down to **O(n)**.

#### Empirical analysis

For a sample worst-case scenario run of the algorithm, 5 tools were added to the system with the borrowed amounts being the same for each tool, and the implemented version of the algorithm was called with them as the input. The number of times the basic operation was called was then collected, giving a value of 9. Checking this against the formula **t(n) = 1 + 2 + 3 \* ((n – 1) - 2)** gives , which matches. This confirms the time complexity of the algorithm being **O(n)**.

Further validation of the time complexity of the algorithm was done by creating a test program which calls an implentation of the algorithm for increasing amounts of tools. As visible in the graph below, the execution times collected from this program confirm that the algorithm is of linear complexity.

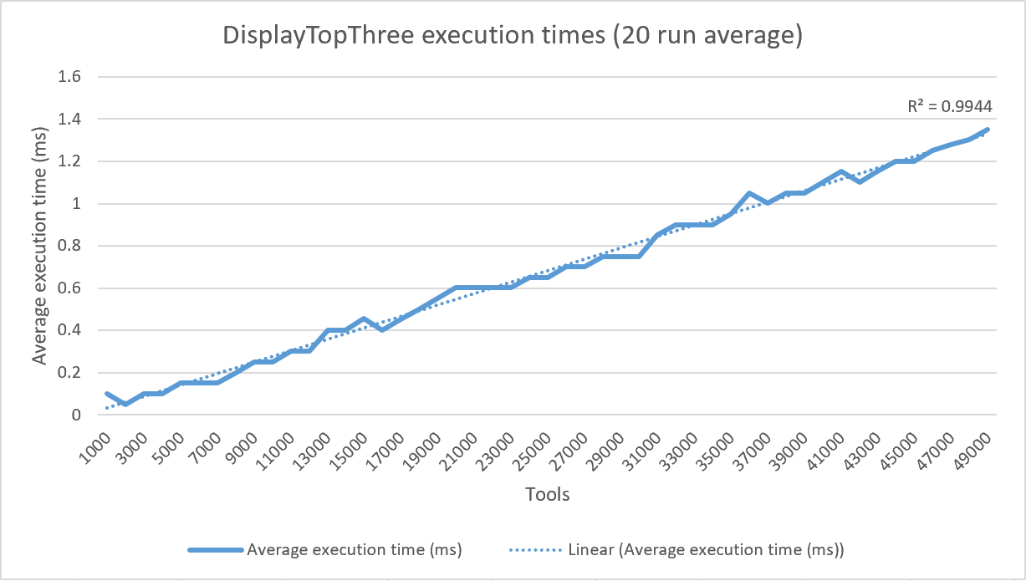


Fig 1: Execution times for the implemented top three algorithm

#### Space efficiency

The only variable which is created and stored in this algorithm is the *topBorrowed* array. This means the algorithm has a constant space complexity, or a space complexity of **O(1)**.

# Software Test Plans and Test Results

### Unit Tests

A number of unit tests were made to confirm the functionality of the system programmatically. These unit tests consisted of testing all of the functions in the ToolLibrarySystem which involved modifying data. A screenshot of the results from these tests can be seen below.

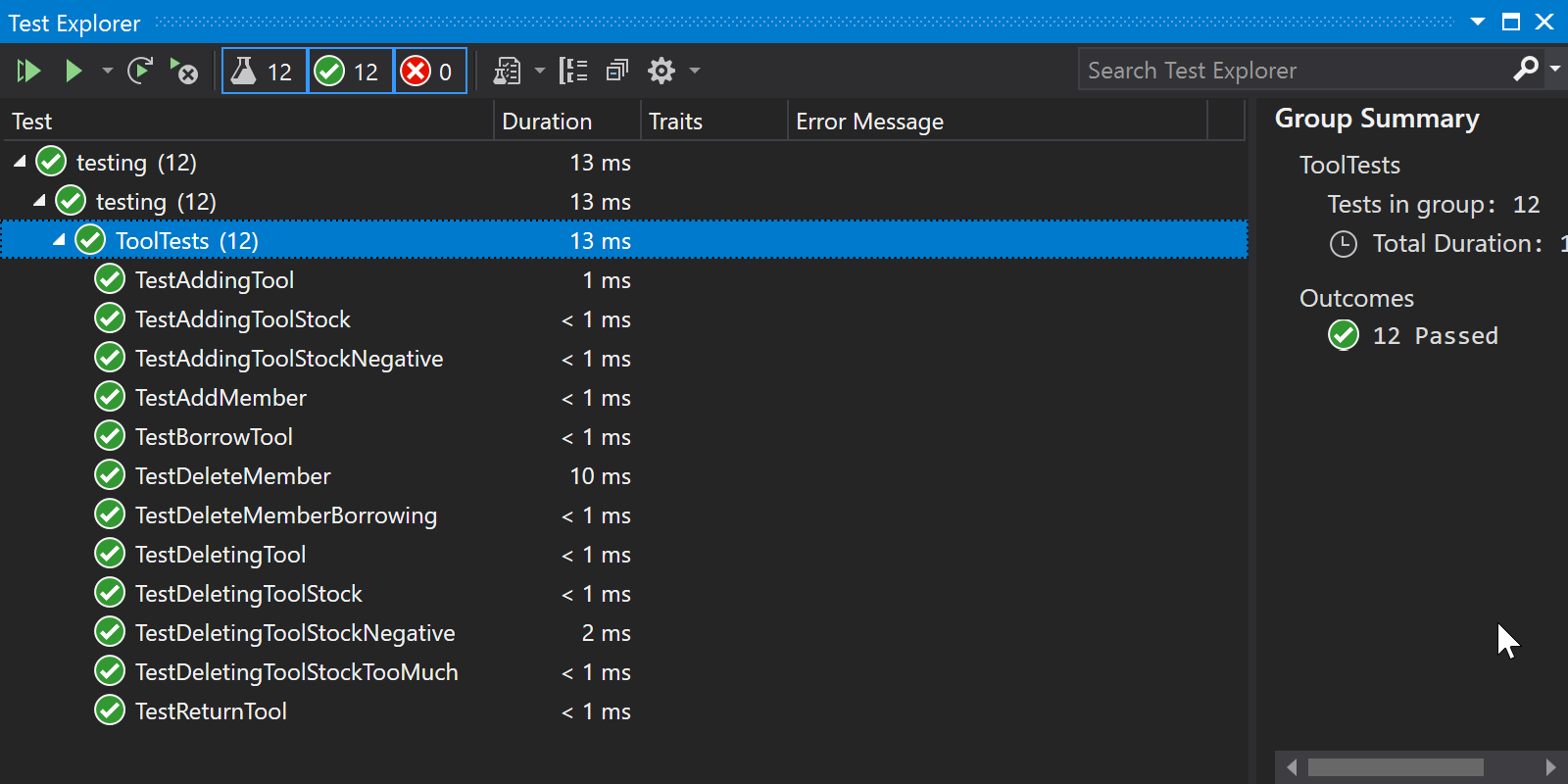


Fig 2: Results of unit tests

### Tests of functionality

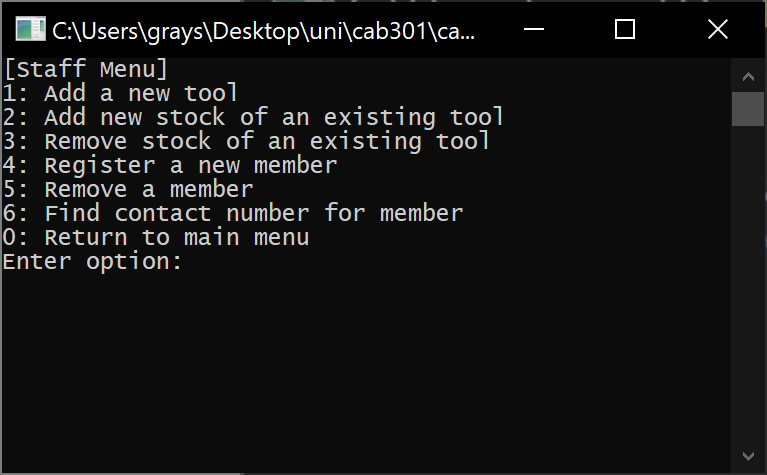
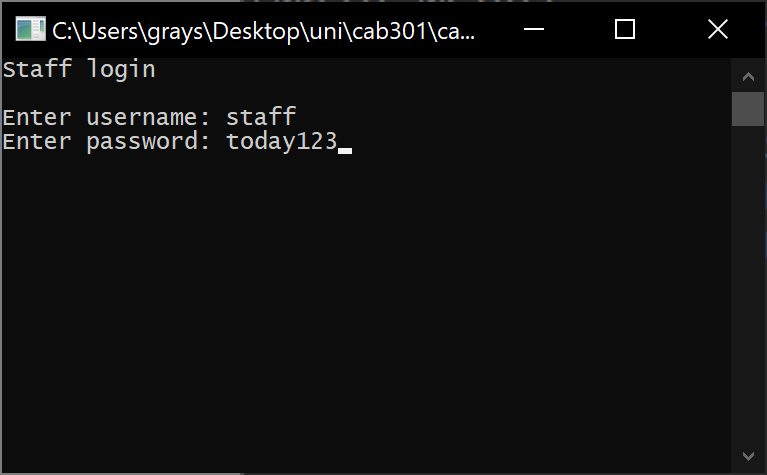
Screenshots for each test scenario can be found in the appendix under Test Images.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test scenario** | **Test data** | **Expected result** | **Result** |
| **Menu pages** | | | |
| Staff login | Username: staff  Password: today123 | User is logged in as staff | Pass |
| Staff failed login | Username: staff  Password: password | User is not logged in and a message is displayed | Pass |
| Member login | Username: KoyGrayson  PIN: 1234 | Member is logged in | Pass |
| Member login wrong username | Username: KovGrayson  PIN: 1234 | Member is not logged in and a message is displayed | Pass |
| Member login wrong PIN | Username: KoyGrayson  PIN: 2549 | Member is not logged in and a message is displayed | Pass |
| **Staff pages** | | | |
| Adding a new tool | Tool name: New tool  Tool quantity: 10  Category: Gardening Tools  Type: Lawn Mowers | The tool is added to the system | Pass |
| Adding tool stock | Category: Gardening Tools  Type: Lawn Mowers  Tool: New tool  Additional stock: 12 | The stock is added to the tool | Pass |
| Adding a negative amount of tool stock | Category: Gardening Tools  Type: Lawn Mowers  Tool: New tool  Additional stock: -10 | The stock is not added to the tool and a message is displayed | Pass |
| Removing tool stock | Category: Gardening Tools  Type: Lawn Mowers  Tool: New tool  Stock to remove: 12 | The stock is removed | Pass |
| Removing a negative amount of tool stock | Category: Gardening Tools  Type: Lawn Mowers  Tool: New tool  Stock to remove: -10 | The stock is not removed and a message is displayed | Pass |
| Removing too much tool stock | Category: Gardening Tools  Type: Line Trimmers  Tool: Bad Line Trimmer  Stock to remove: 200 | The stock is not removed and a message is displayed | Pass |
| Member registration | First name: Frank  Last name: Walker Mobile number: 1300733000  PIN: 4832 | The member is added | Pass |
| Member removal | User: Bob Jeff | The member is removed | Pass |
| Finding member contact phone number | First name: Grayson  Last name: Koy | The member is found and their contact number is displayed | Pass |
| Finding member contact number with non-existent name | First name: Grayson  Last name: Jeffingtons | The member is not found and a message is displayed | Pass |
| Going back to main menu |  | Goes back to the main menu | Pass |
| **Member pages** | | | |
| Display tools by category | Category: Gardening Tools  Type: Line Trimmers | The tools in the selected category of the selected type are displayed | Pass |
| Borrow tool from library | Category: Gardening Tools  Type: Line Trimmers  Tool: Bad Line Trimmer | The tool is borrowed | Pass |
| Trying to borrow a tool when 3 are already borrowed |  | A message is displayed saying you cannot borrow any more tools | Pass |
| Borrowing the same tool more than once | Category: Gardening Tools  Type: Line Trimmers  Tool: Bad Line Trimmer | The tool is borrowed | Pass |
| Return tool to library | Tool: Bad Line Trimmer | The tool is returned | Pass |
| List borrowed tools |  | The member’s borrowed tools are listed | Pass |
| List borrowed tools with no borrowed tools |  | A message saying that no tools have been borrowed is displayed | Pass |
| Display most frequently borrowed tools |  | The top 3 most frequently borrowed tools are displayed | Pass |
| Display most frequently borrowed tools when no tools have been borrowed |  | A message saying that no tools have been borrowed is displayed | Pass |
| Display most frequently borrowed tools when only 2 tools have been borrowed |  | The top 3 most frequently borrowed tools are displayed, with the third tool with no borrowings still being shown. | Pass |
| **User input validation** | | | |
| Incorrect page is selected in menu | Page number: 4 | A message is displayed telling the user to input a valid option, and input is requested again | Pass |
| Incorrect entry in list selector is selected | Selected index: 100 | A message is displayed telling the user to input a valid option, and input is requested again | Pass |
| Entering a string into an int input | Int input: gdfdg | A message is displayed telling the user to input a valid number, and input is requested again | Pass |
| Entering a mobile number with letters when registering | Mobile number: 4239asadsd453 | A message is displayed telling the user to input only numbers, and input is requested again | Pass |
| Entering a PIN of incorrect length when registering | PIN: 12345 | A message is displayed telling the user to input a 4-digit pin, and input is requested again | Pass |

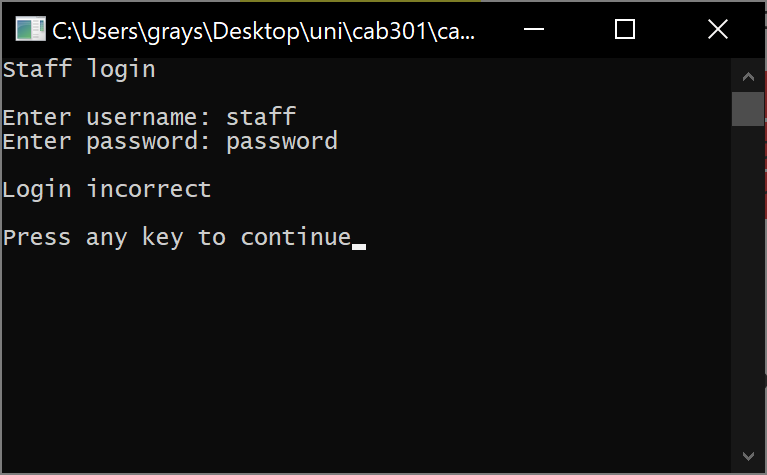
# Appendix

## Test images

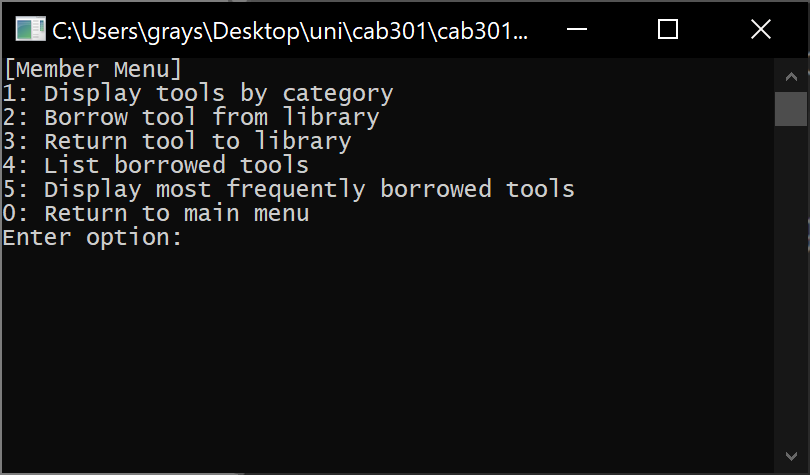
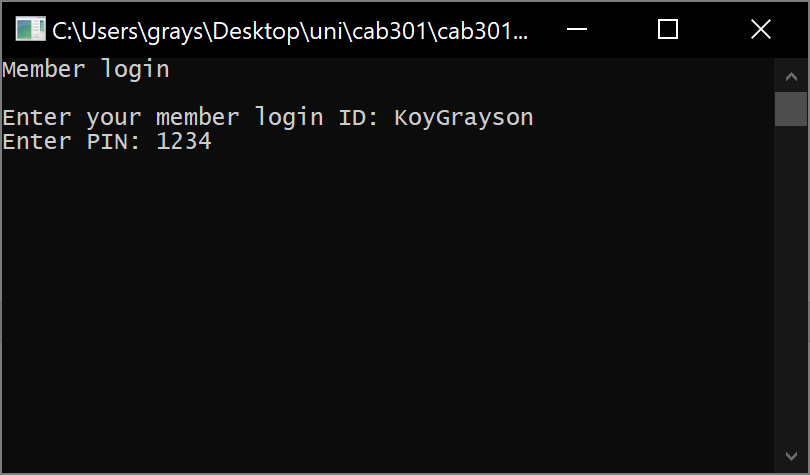
### Staff login



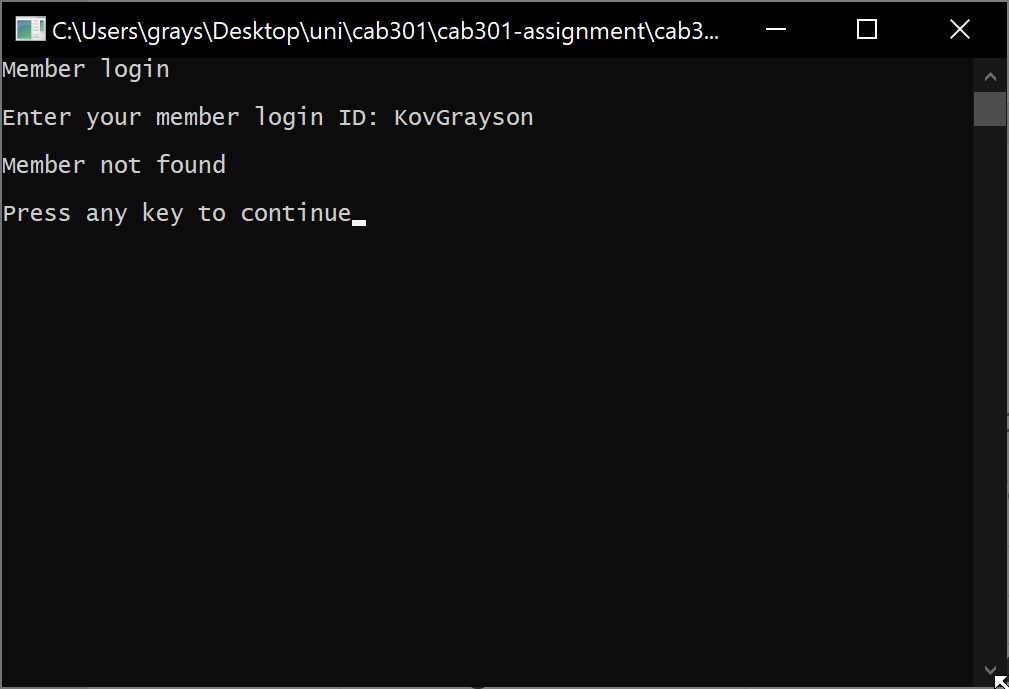
### Staff failed login



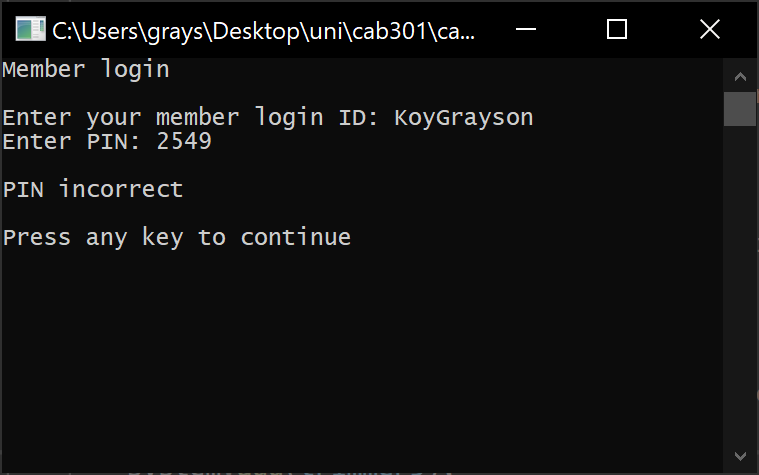
### Member login



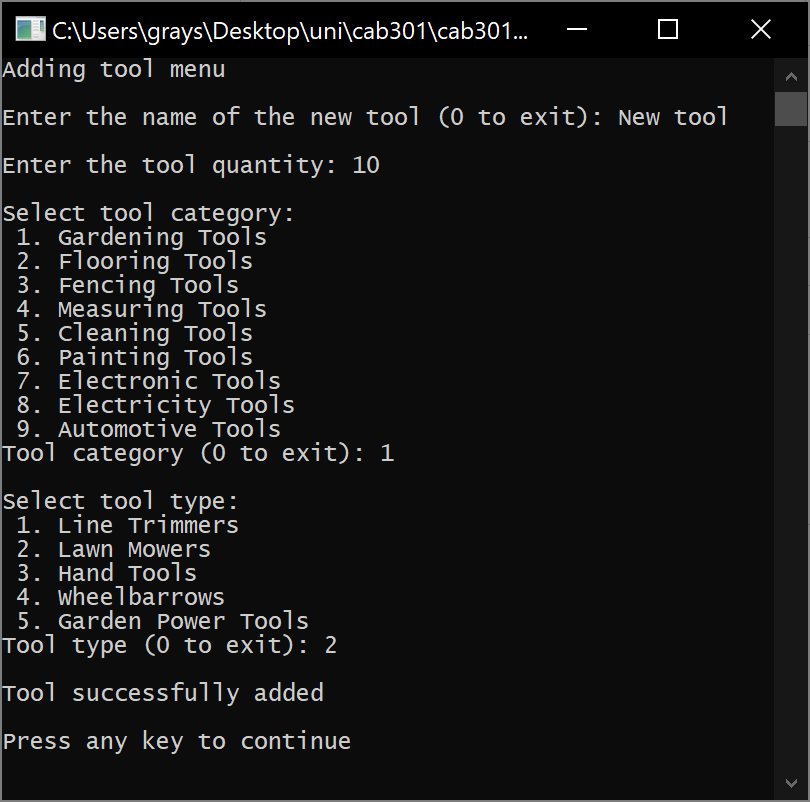
### Member login wrong username



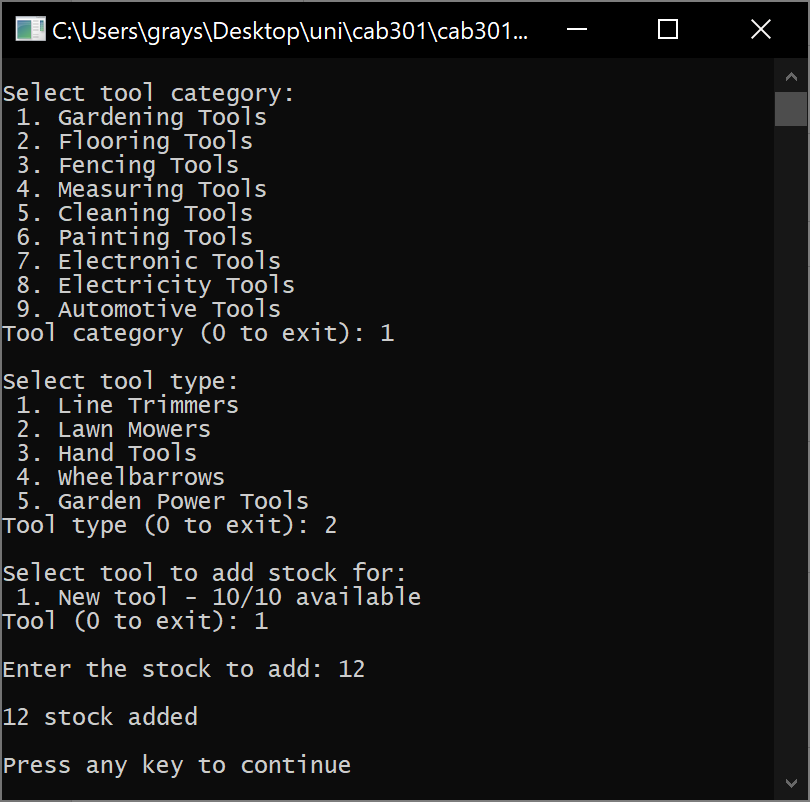
### Member login wrong PIN



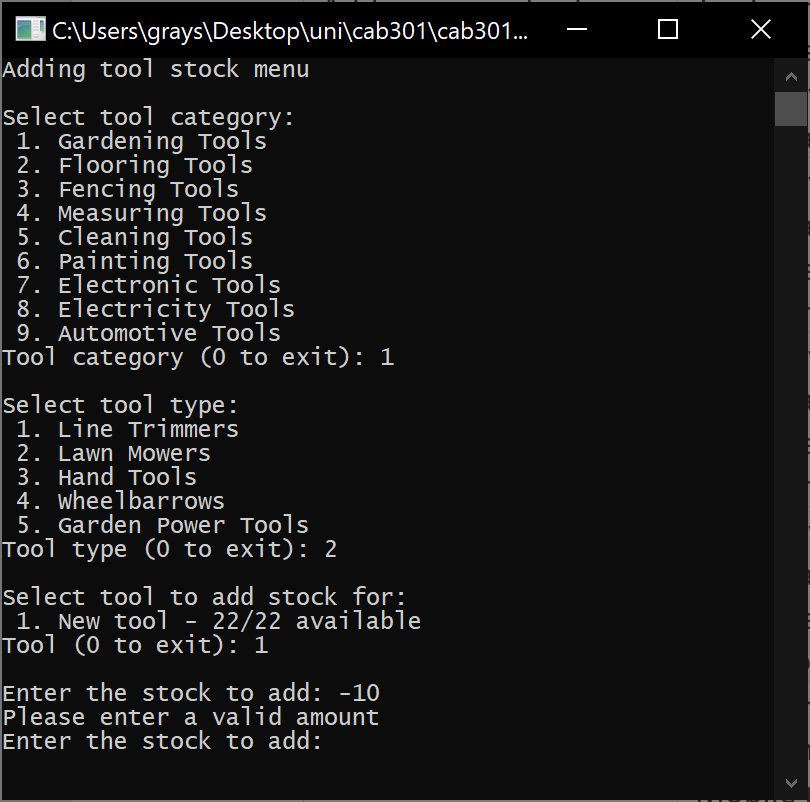
### Adding a new tool



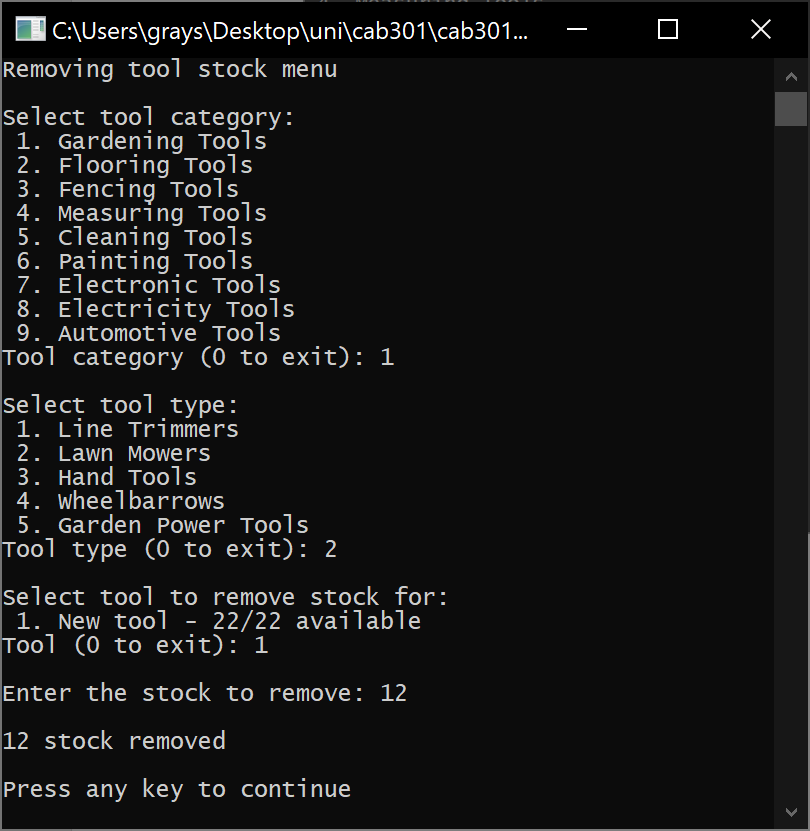
### Adding tool stock



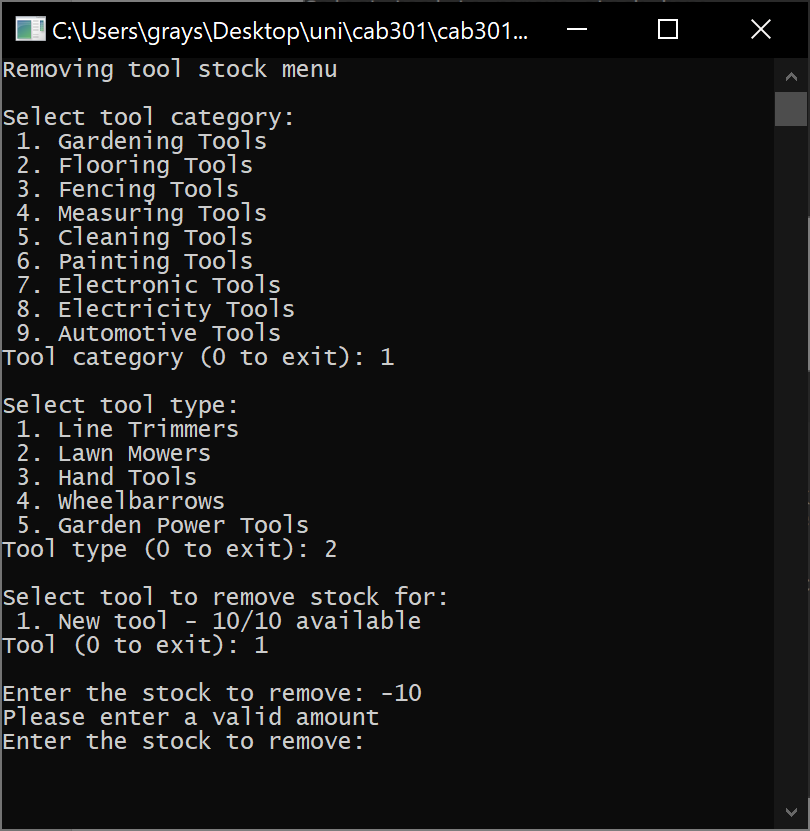
### Adding a negative amount of tool stock



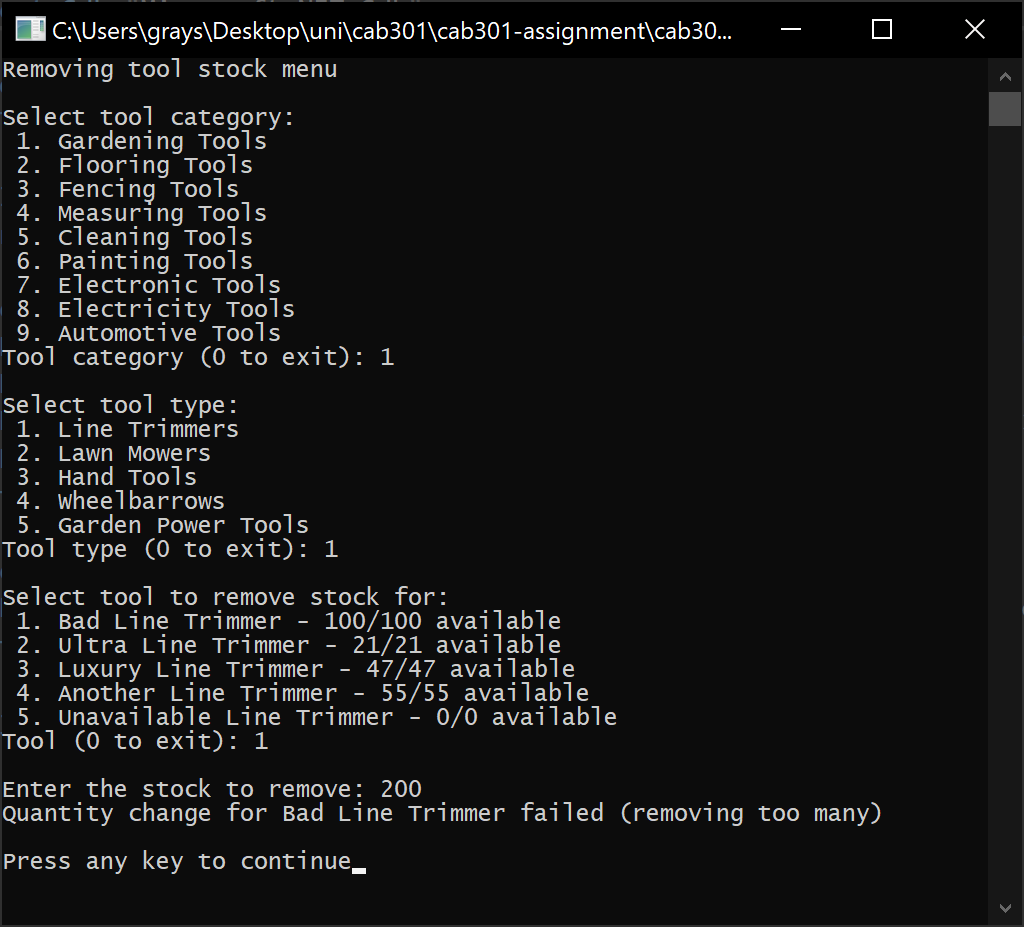
### Removing tool stock



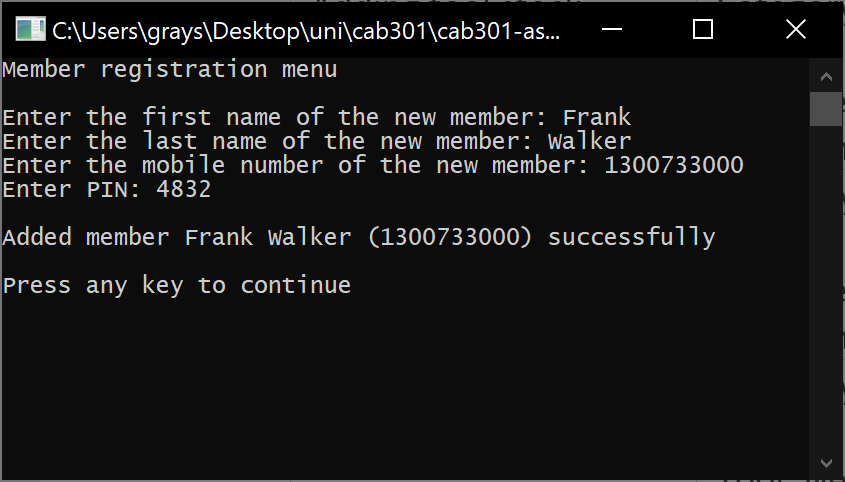
### Removing a negative amount of tool stock



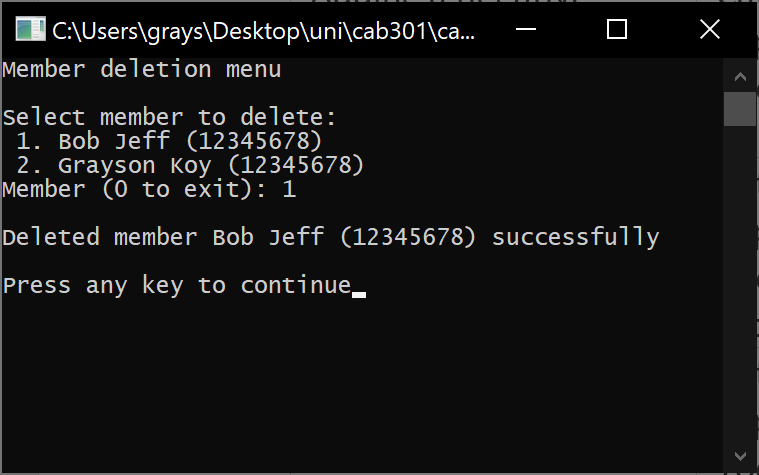
### Removing too much tool stock



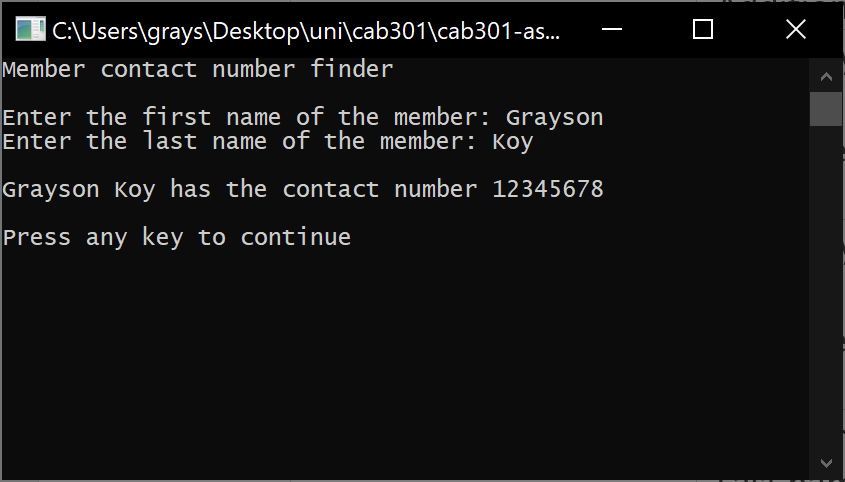
### Member registration



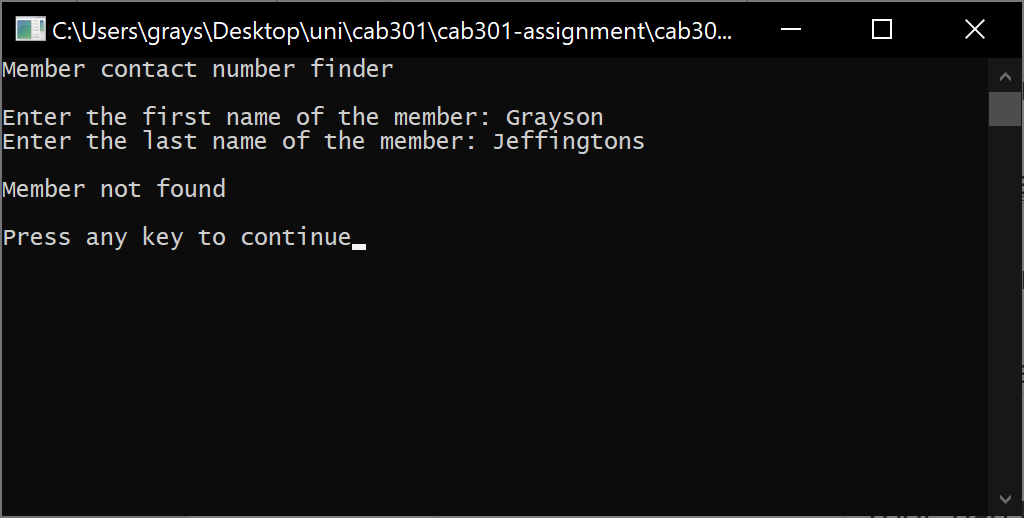
### Member removal



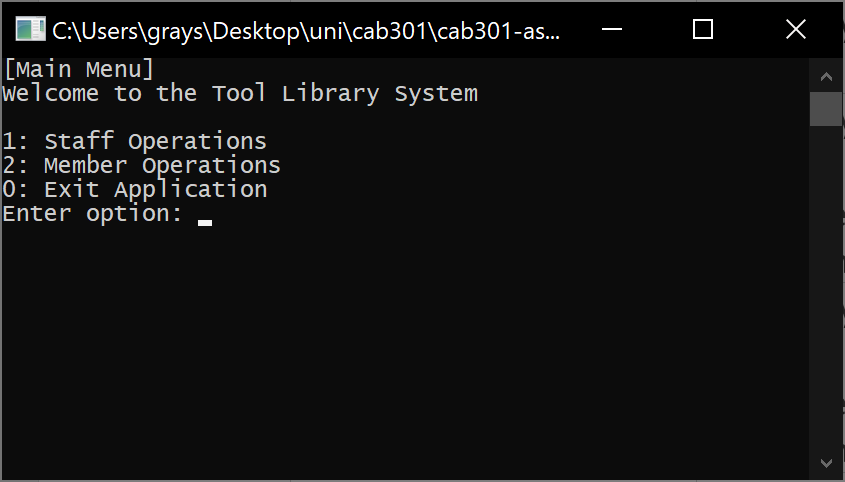
### Finding member contact phone number



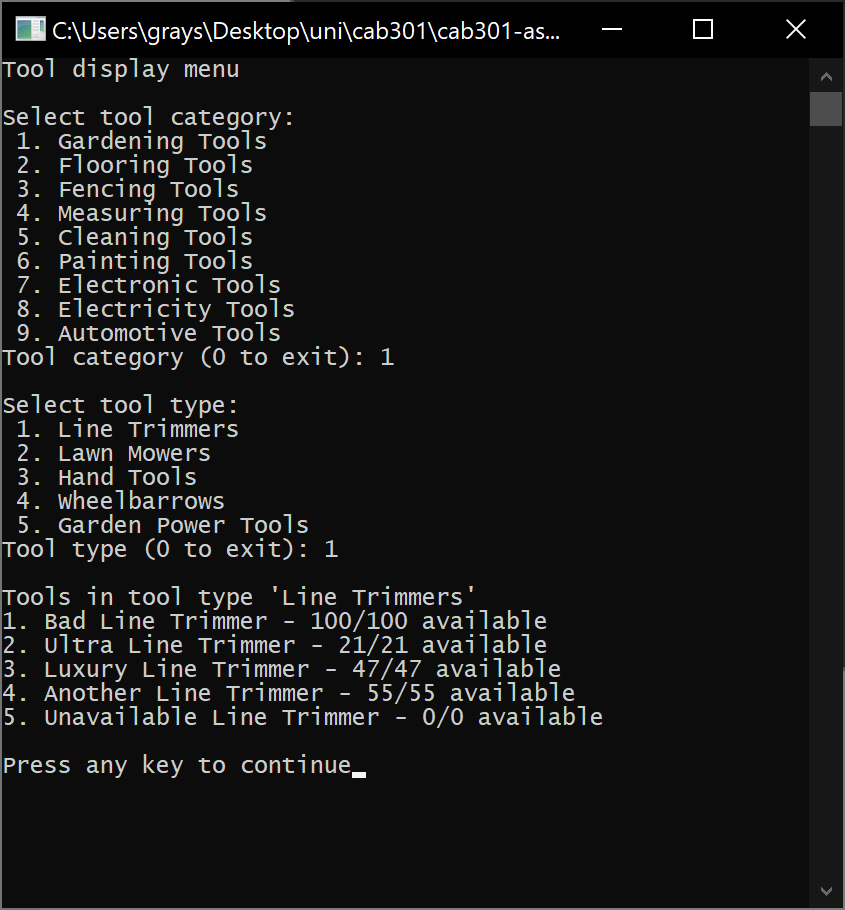
### Finding member contact number with non-existent name



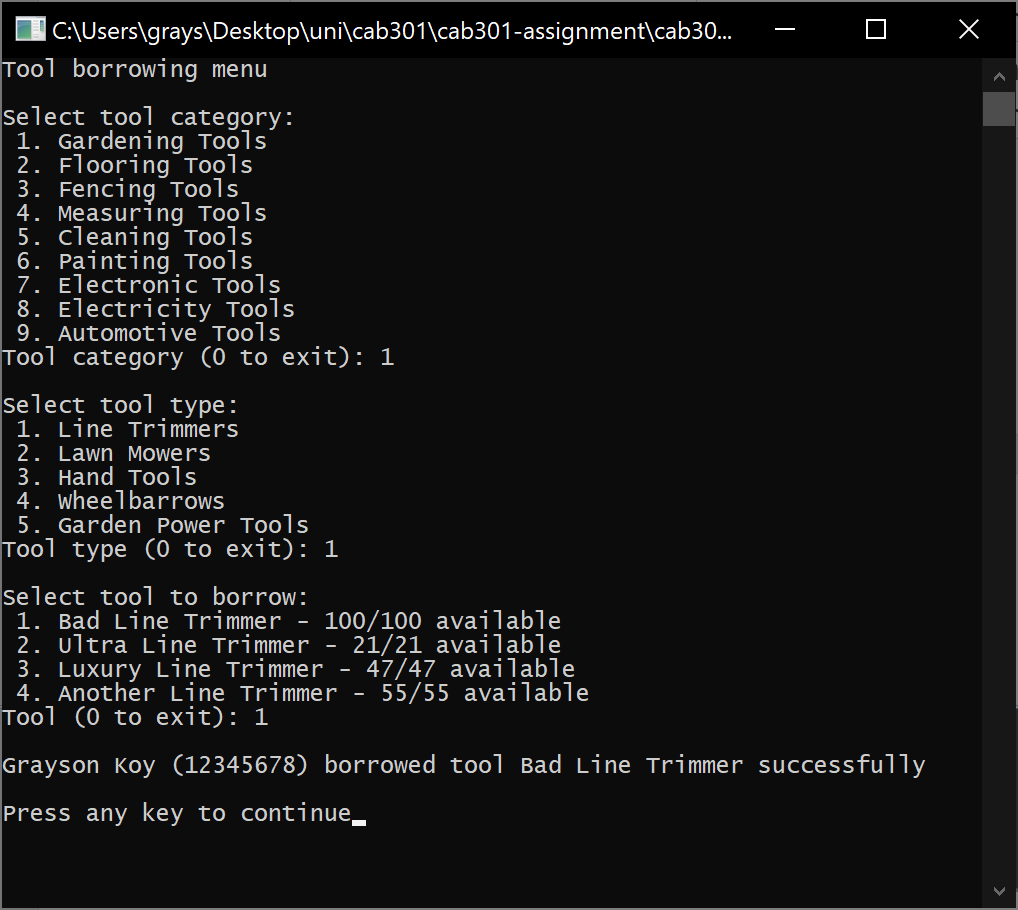
### Going back to main menu



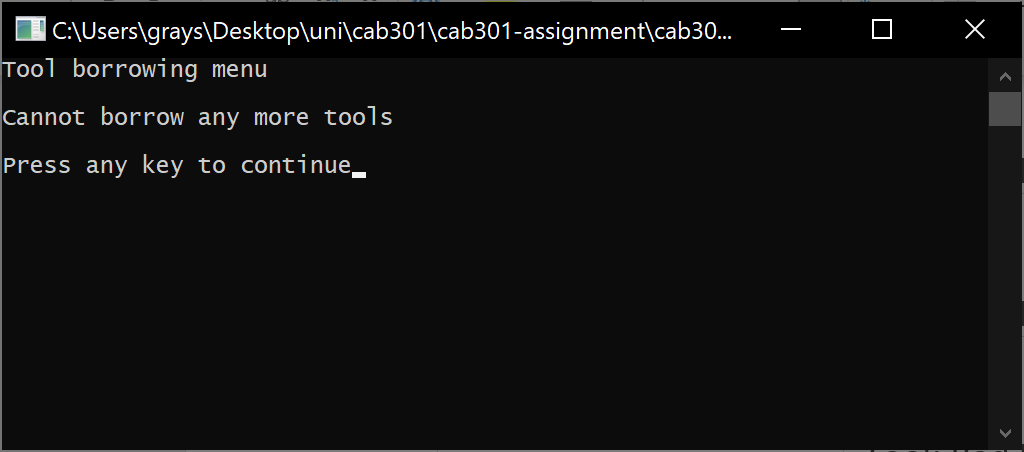
### Display tools by category



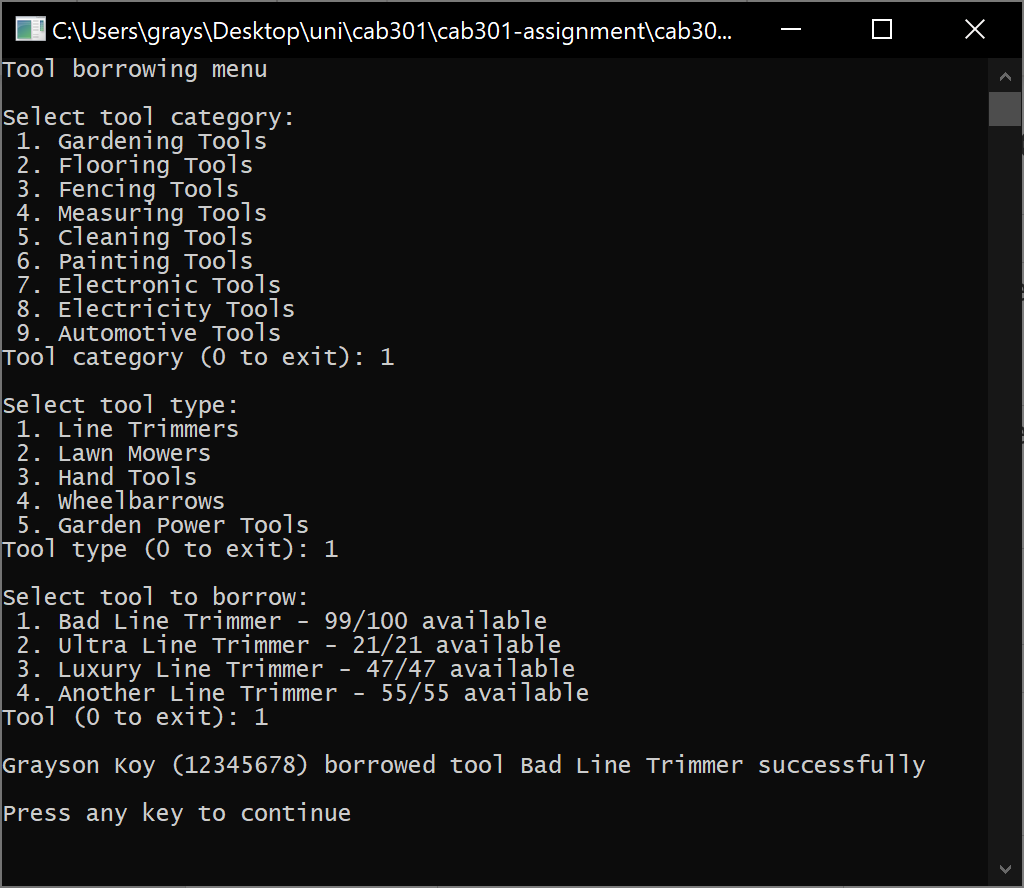
### Borrow tool from library



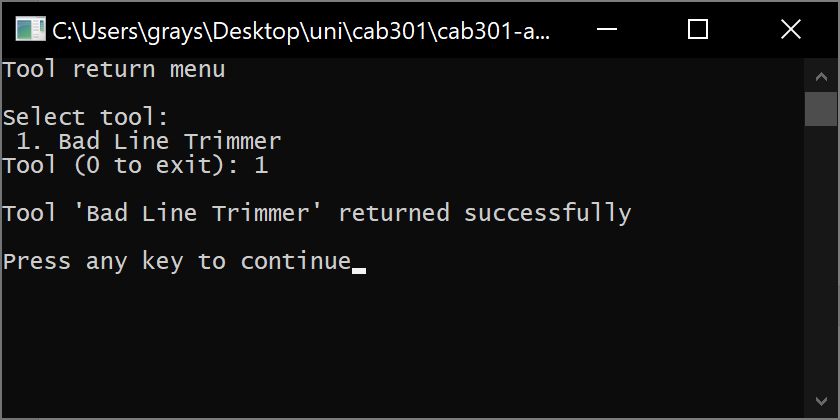
### Trying to borrow a tool when 3 are already borrowed



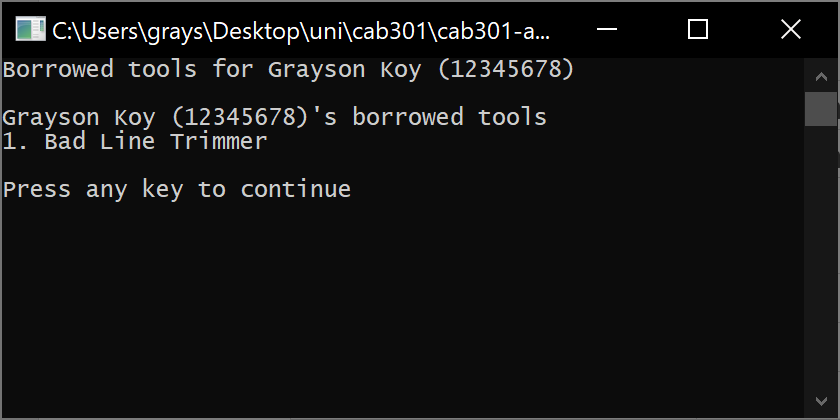
### Borrowing the same tool more than once



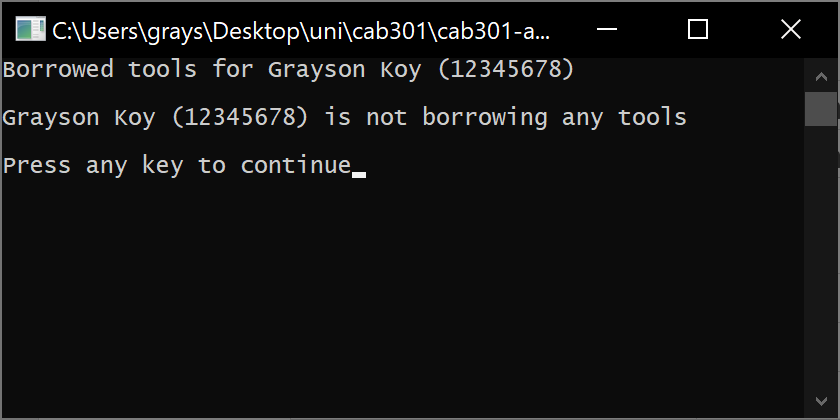
### Return tool to library



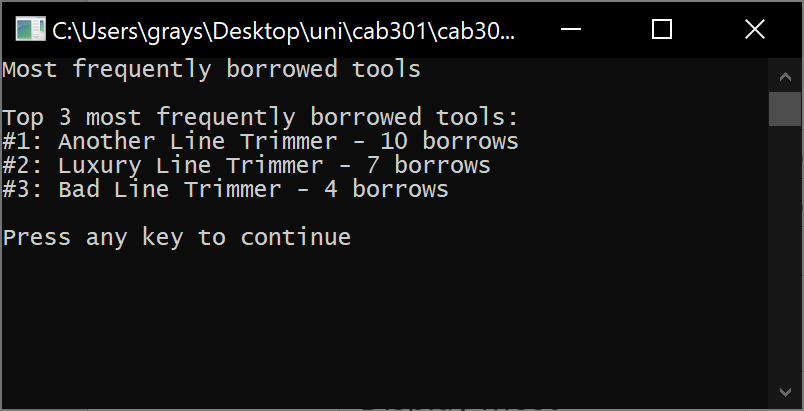
### List borrowed tools



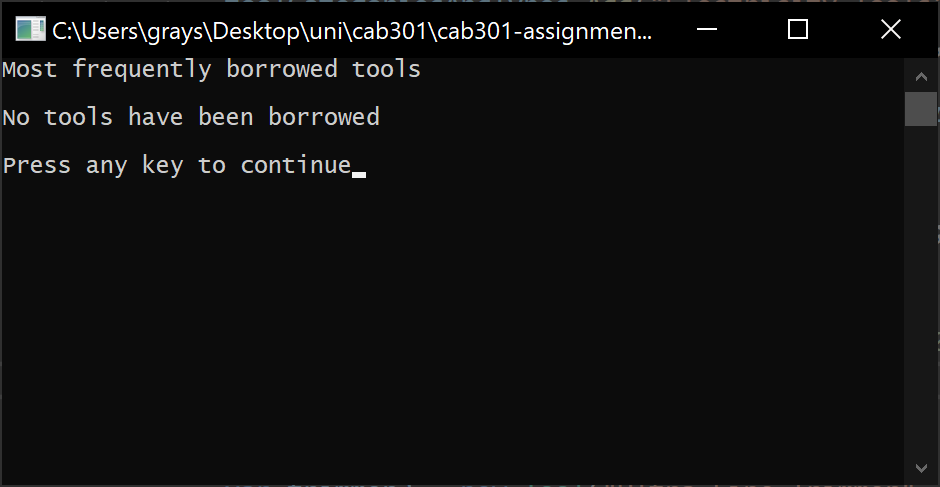
### List borrowed tools with no borrowed tools



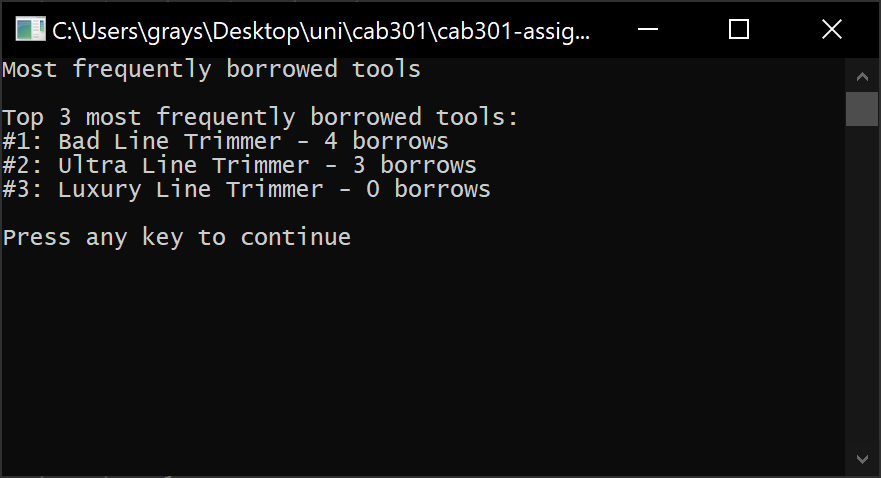
### Display most frequently borrowed tools



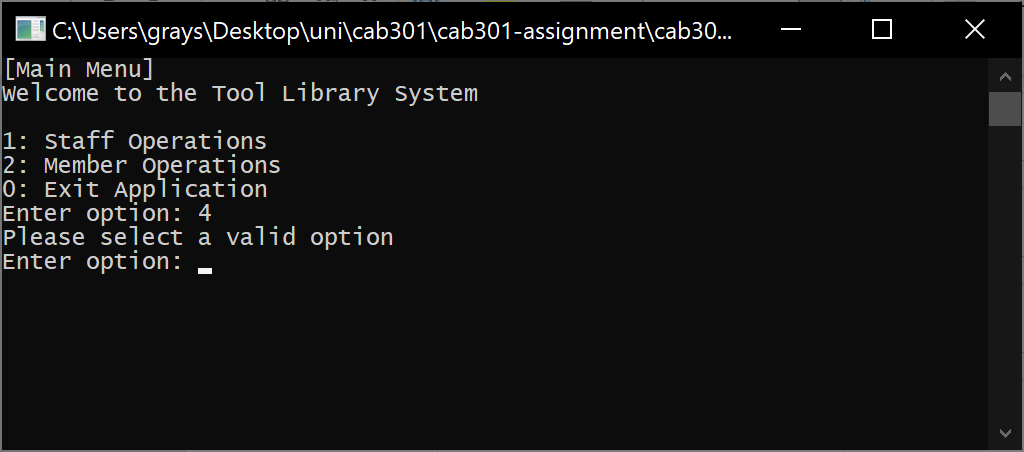
### Display most frequently borrowed tools when no tools have been borrowed



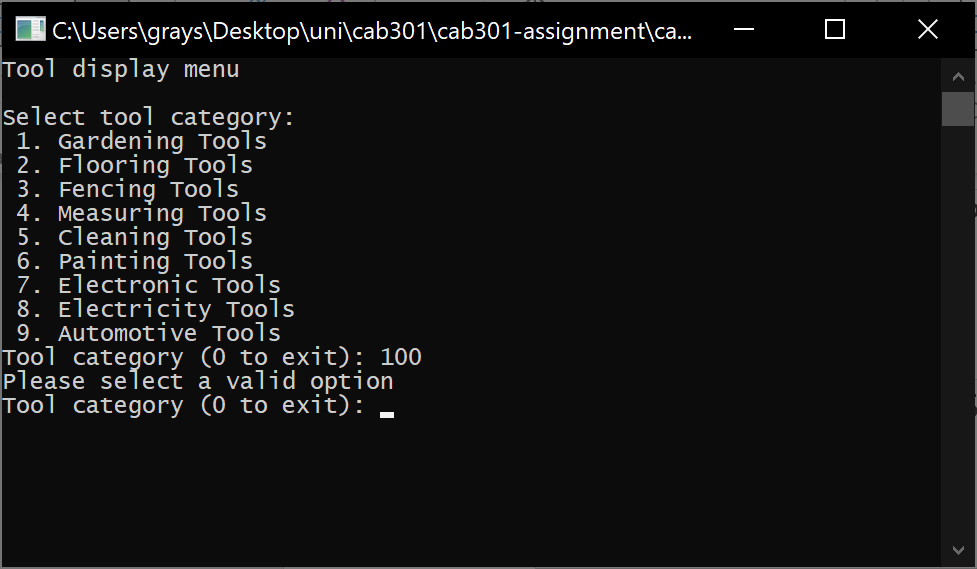
### Display most frequently borrowed tools when only 2 tools have been borrowed



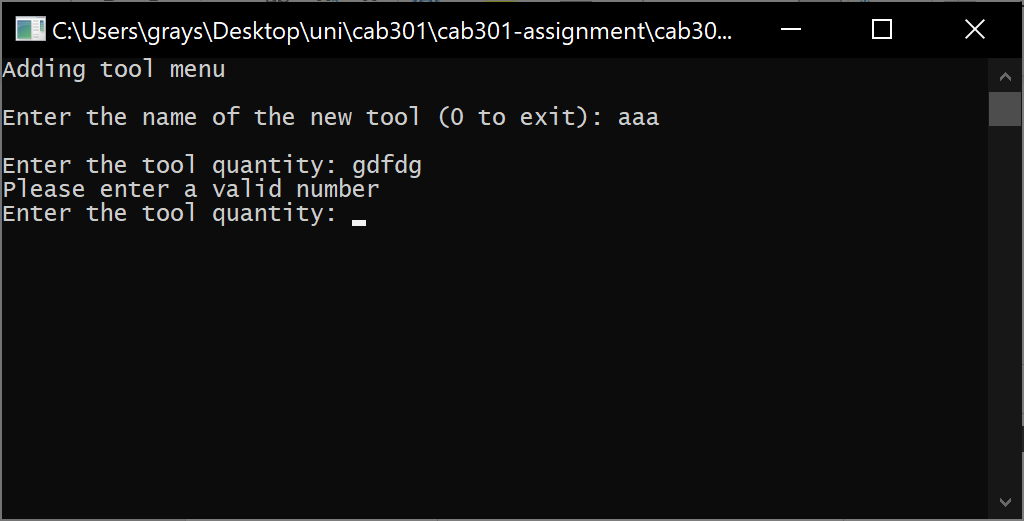
### Incorrect page is selected in menu



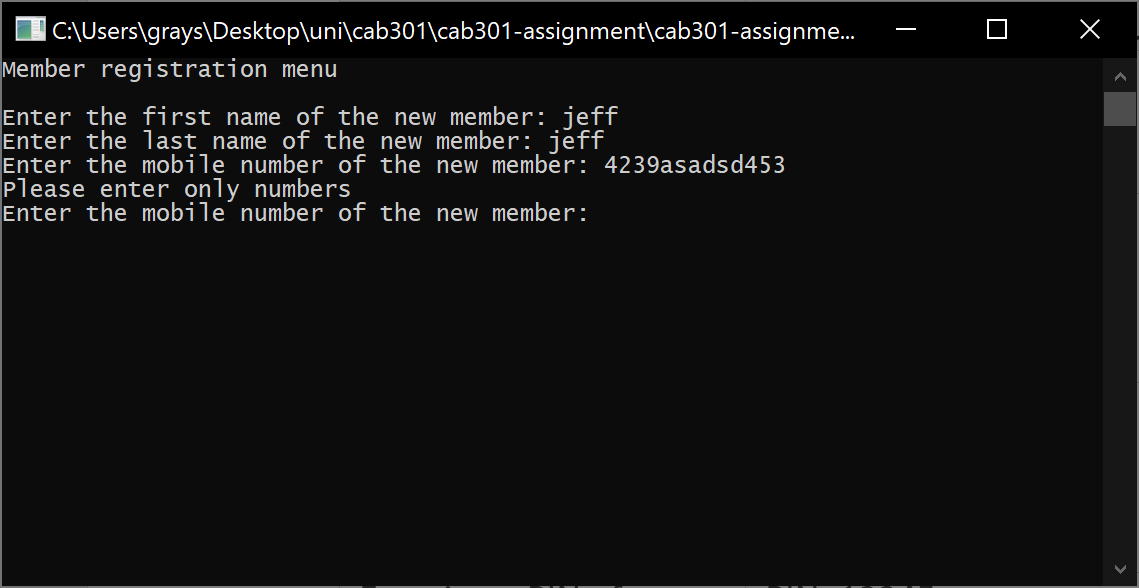
### Incorrect entry in list selector is selected



### Entering a string into an int input



### Entering a mobile number with letters when registering



### Entering a PIN of incorrect length when registering

