CAB301 REPORT

Algorithm & Complexity

MAY 31, 2021

N10500511 - Aung Khant Kyaw

Table of Contents

1 Introduction

The tool library system is a software application which can be run on Windows desktop. The software is built using C# on Microsoft Visual Studio 2019. To implement this software application, I will need to use some of the data structures and algorithms to store, manage and manipulate data. The software is used in a community library to assist the community members with extra digital functions such as managing the information about the tools as well as managing the members in the library.

There are 9 categories in the tool library and each category consists of tools which are either from donations or funds raised by the community team. The members are needed to be registered in the system before they can borrow a tool. The registration of members is managed by the staff in the community.

The staff have several functionalities such as adding a new tool, adding quantities to the existing tool, remove some quantities of the existing tool, registering a new member to the system, removing a member from the system, finding the contact number of the member using member's first name and last name. These functions are done by the staff when successfully logged in.

The members also have different functions such as displaying the information about the tools from a different category, browsing the quantity of the given tool, borrowing a tool and returning it to the system, listing all the tools that a member is currently holding, displaying the top three tools which are rented frequently in the system.

There are three parts in this report consisting of algorithm design and analysis and the software test plan and test results. The algorithm needs to show how the design is made for showing the top three tools borrowed in the library. The analysis has to be done calculating the efficiency of time. These will be explained more in the sections. The test plan will be displayed as a table making it easy to read and check for the results.

2 Algorithm design

An algorithm is a sequence of unambiguous instructions for solving a computational problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time. Algorithms are independent from programming languages. An algorithm can be implemented in different programming languages. An implementation of an algorithm is a computer program.

A selection sort is used to design to sort the tools in the library based on the number of borrowings. The selection sort works by sorting elements in the array repeatedly. The algorithm finds the small element in the array and replacing the first position of the array with the smallest element. By continuing this way, the array is sorted using selection sort.

The pseudocode of selection sort is as follows:

Algorithm: Selection-sort pseudocode

```
Algorithm Sort (arr[0...n-1])

                    n = size of array
                    for i=0 to n-1
                      min_idx = i
                      for j=0 to n
                          if arr[j] < arr[min_idx] then
                              min_idx = j;
                          end if
                      end for
                      temp = arr[min_idx]
                      swap arr[i] and arr[min_idx]
                    end for
```

3 Algorithm analysis

The selection sort is efficient in sorting large objects with small keys. However, there is one worst case in selection algorithm where we want to sort the array in the ascending order when it is already in the descending order. The sorting algorithm has a minimum number of swaps where n -1 would be the worst case of all.

Selection sort is quadratic not only in the worst case but also in a normal case but the sorting algorithm requires no extra memory.

3.1.1 Complexity

$$(n-1)+(n-2)+\ldots+1=\sum_{i=1}^{n-1} i$$

By arithmetic progression,

$$\sum_{i=1}^{n-1} i=\frac{(n-1)+1}{2}(n-1)=\frac{1}{2}n(n-1)=\frac{1}{2}(n^2-n)$$

Figure 3.1.1 Sorting Algorithm Complexity

To measure the time complexity of the algorithm, a program written with the algorithm be needed to be executed. The measurement of time depends not only on the computer processor and the programming language, but also the system load and the type of operating system the machine is using. The time complexity of sorting algorithm is O(n2) where it is a number of comparisons. When sorting occurs, n-1 is required as one swap for the elements.

To compute the time in a mathematical function, a formula is used as follows:

T(n) = total time

c1 = one line of code statement

c2 = one line of code statement

ci = a number of lines of code statements

In our selection sort analysis,

```
Algorithm Sort (arr[0...n-1])

                    n = size of array                    c1
                    for i=0 to n-1
                           min_idx = I                   c2
                           for j=0 to n
```

```
                    if arr[j] < arr[min_idx] then  c3
                        min_idx = j;               c4
                    end if
                end for
                temp = arr[min_idx]                c5
                swap arr[i] and arr[min_idx]        c6
            end for
```

T(n) = c1 + (c2 + (c3 + c4)(n) + c5 + c6)(n-1)

For a particular machine, c1 = 3microseconds, c2 = 2 microseconds, c3 = 3 microseconds, c4 = 1 microseconds, c5 = 1 microseconds, c6 = 2 microseconds

T(n) = 3 + (2 + (3+1)(n) + 1 + 2)(n-1)

T(n) = 3 + (4n + 5)*(n-1)

When n = 10, T(10) = 3 + (4*10 + 5)*(10-1) = 3 + 45*9 = 3 + 405 = 408 microseconds

When n = 100, T(100) = 3 + (4*100 + 5)(100-1) = 3 + 395 * 99 = 3 + 39105 = 39108 microseconds

4 Software test plan and test results

4.1 Main Menu Functionalities

Test description	Input	Expected Result	Actual Result	Pass/fail	Appendix /Screenshots
Staff login	Staff username: staff Staff password: today123	Staff logged in and staff menu appeared	Staff logged in and staff menu appeared	Pass	Figure 1
Member Login	First name: a Last name: a PIN: a	Member logged in and member menu appeared	Member logged in and member menu appeared	Pass	Figure 2
Return to main menu	Menu option: 0	Main menu appeared	Main menu appeared	Pass	Figure 3

4.2 Staff Menu Functionalities

Test description	Input	Expected Result	Actual Result	Pass/fail	Appendix /Screenshots
Add a new tool	Tool name: Ozito 260w delta sander	New tool added with a default quantity of 1	New tool added to the system	Pass	Figure 4

Add new pieces of an existing tool	Choose tool type and add quantity: 4	Tool quantity to be increased by the input number	4 new pieces added to the tool	Pass	Figure 5
Remove some pieces of a tool	Choose tool types and remove quantity: 3	3 pieces removed and 2 left for the tool	3 pieces removed and 2 left for the tool	Pass	Figure 6
Register a new member	First name: aung Last name: kyaw Contact number: 0474268017 PIN: 1999	Member added to the system	Member successfully added to the system	Pass	Figure 7
Remove a member	Select aung as number: 2	Member removed from the system	Member successfully removed from the system	Pass	Figure 8
Find the contact number of a member	Using first name: aung Last name: kyaw	Reveal contact number of aung	Found contact number of input member	Pass	Figure 9
Return to main menu	Menu option: 0	Main menu appeared	Main menu appeared	Pass	Figure 3

4.3 Member Menu Functionalities

Test description	Input	Expected Result	Actual Result	Pass/fail	Appendix /Screenshots
Display all the tools type	Choose 6. Gardening Tools and then 1. Sanding tools	Tools in the category displayed on the console	Tools in the category displayed on the console	Pass	Figure 10
Borrow a tool	Choose 6. Gardening Tools and then 1. Sanding tools. Type the name of the tool: Irwin 125mm Orbital Sander	Tools borrowed by the member	Tools borrowed by the member	Pass	Figure 11
Return a tool	Borrowed tools displayed. Select: 1	Tool is returned to the library	Tools is returned to the library	Pass	Figure 12
List all the tools being rented	Choose menu option 4.	Show a list of tools being rented	Tool list shown	Pass	Figure 13
Display top three most frequently rented tools	Choose menu option 5.	Show a list of top three tools	Tool list shown	Pass	Figure 14
Return to main menu	Menu option: 0	Main menu appeared	Main menu appeared	Pass	Figure 3

5 Screenshots of tests

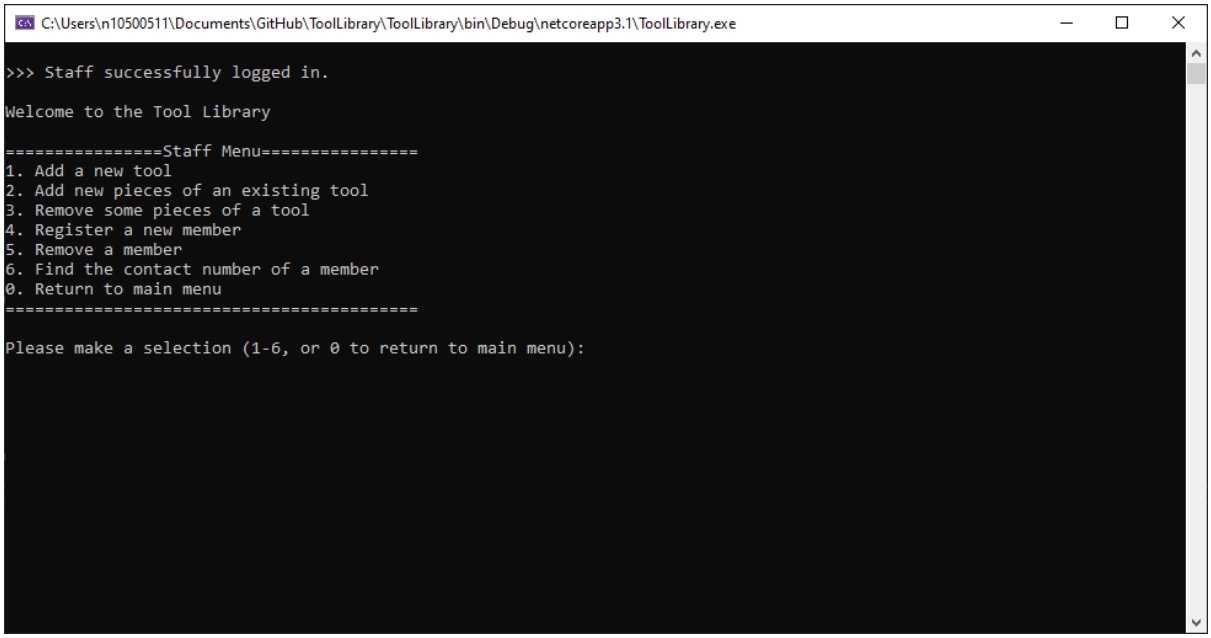

Figure 1 – Staff logged in

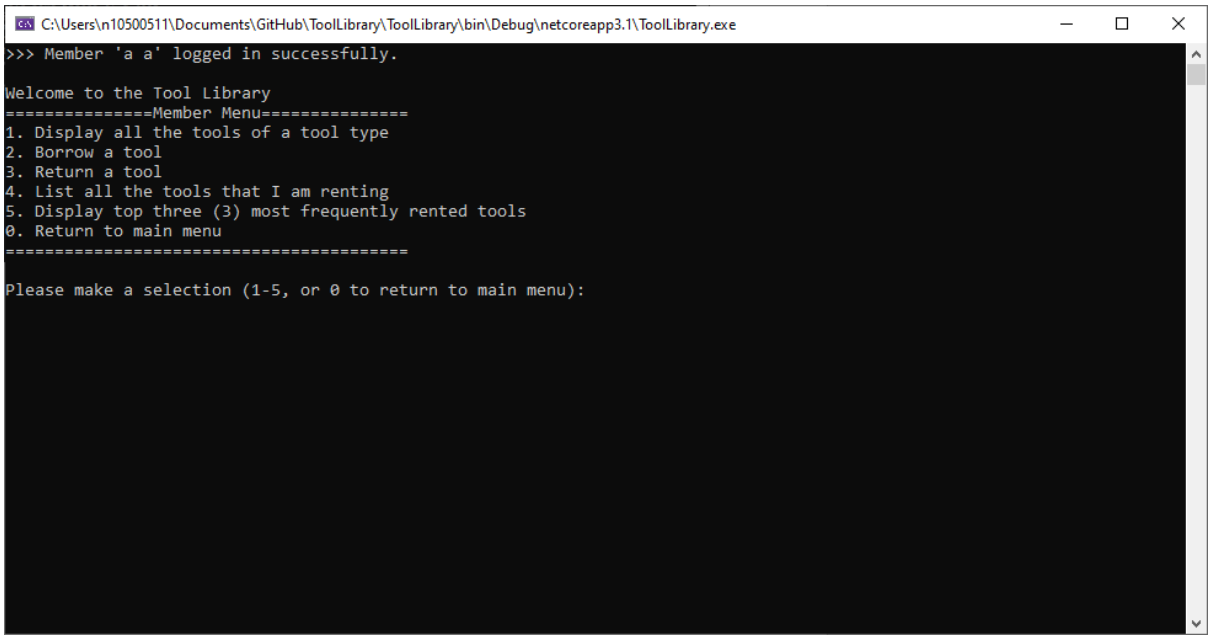

Figure 2 – member logged in

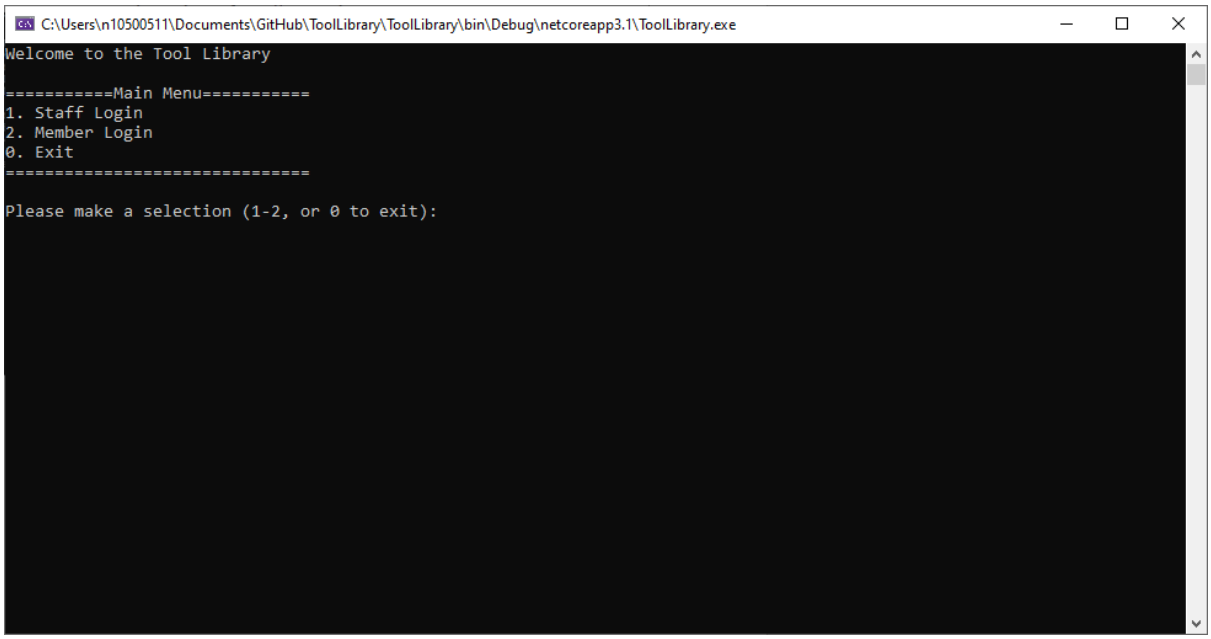

Figure 3 – main menu

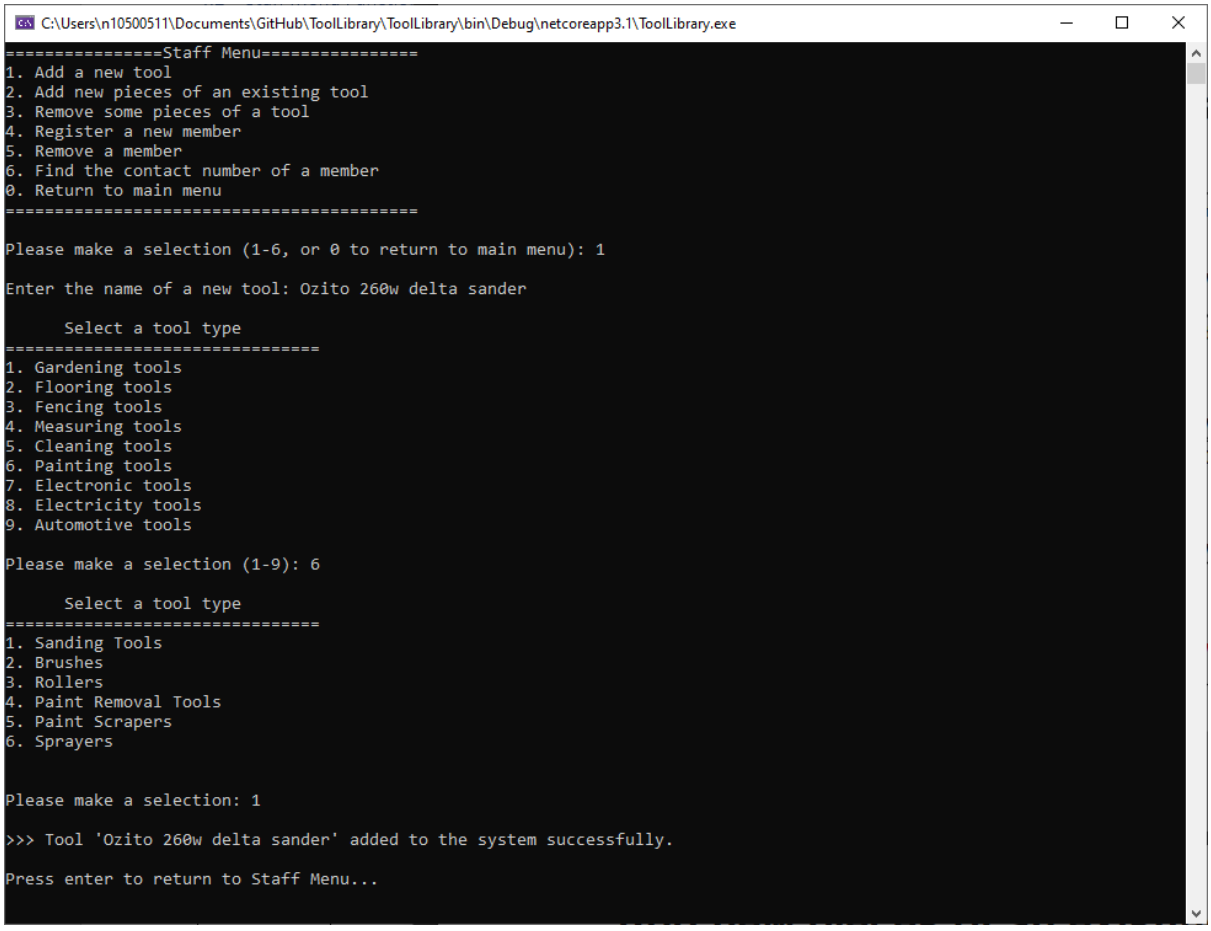

Figure 4 – Add new tool

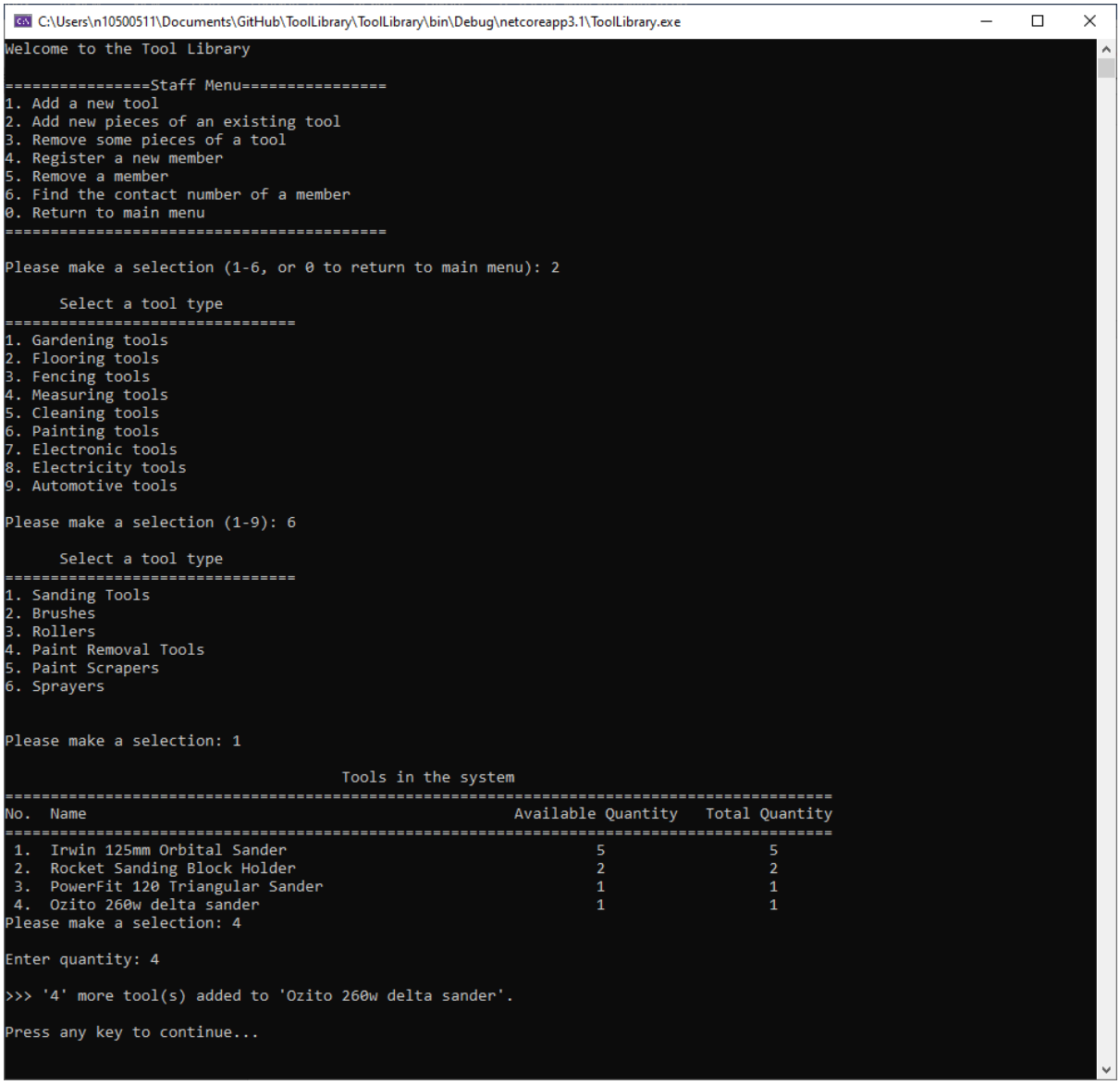

Figure 5 – add new pieces

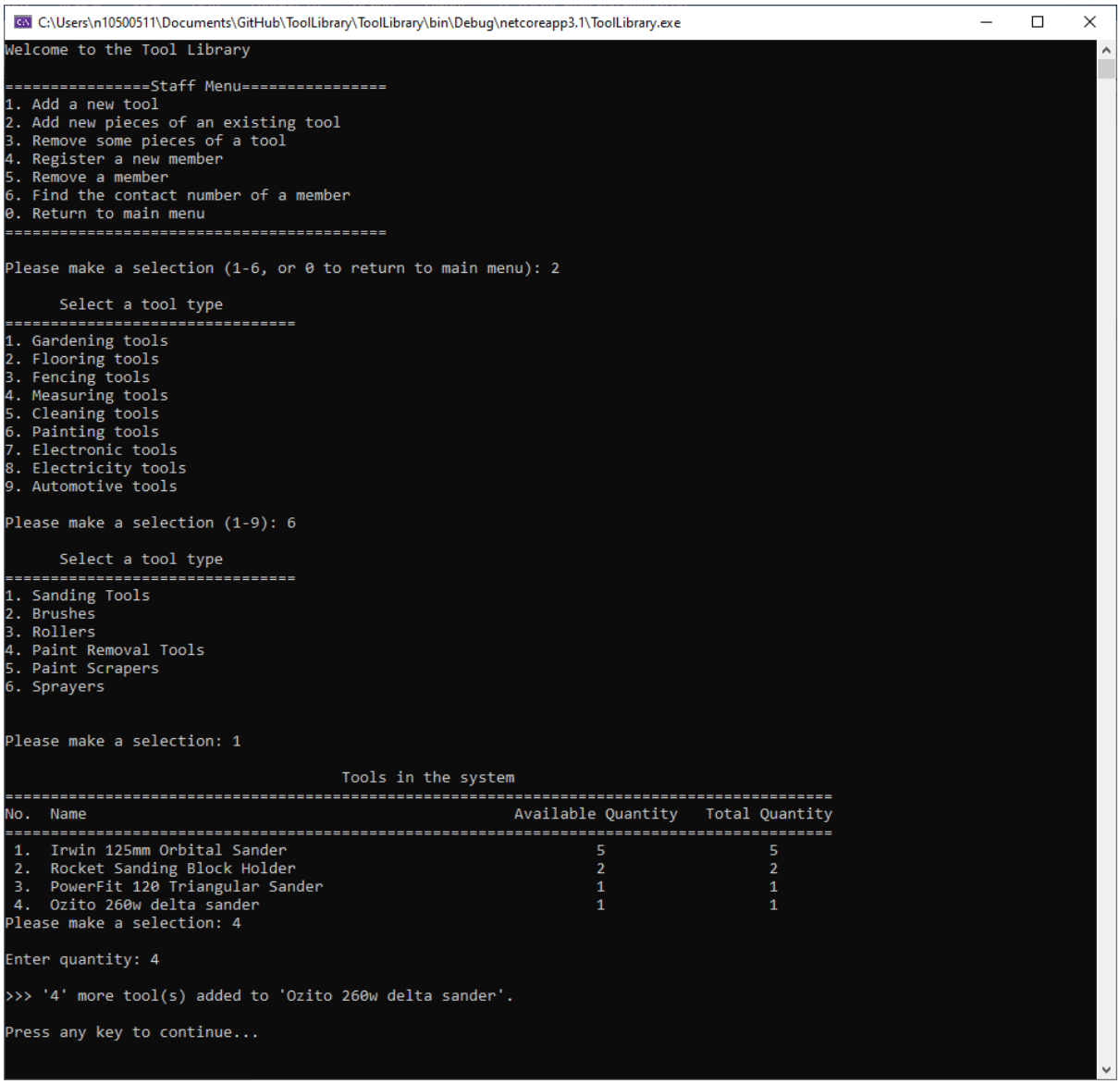

Figure 6 – remove some pieces

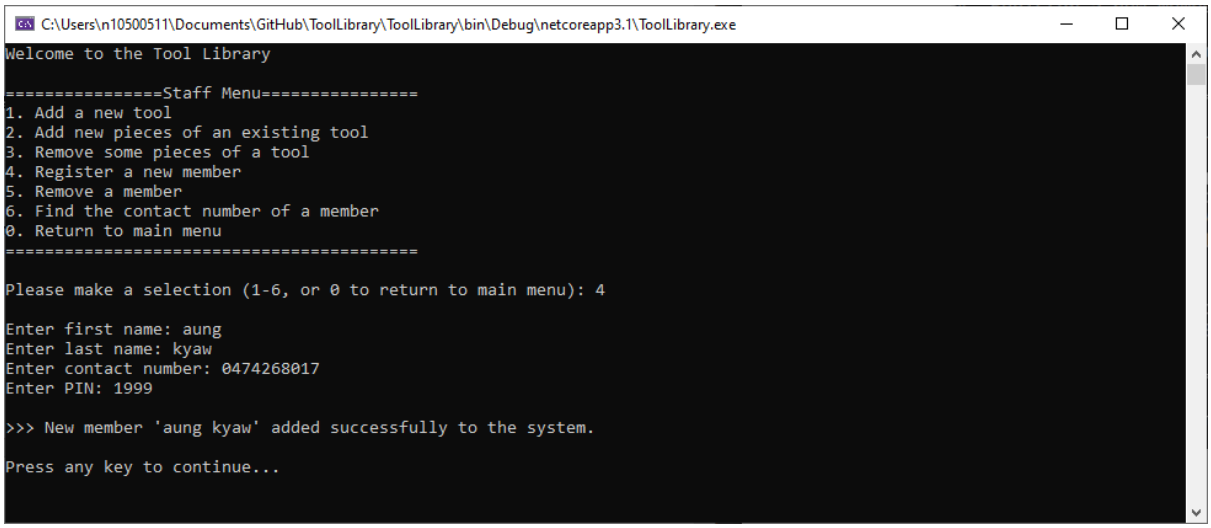

Figure 7 – register a new member

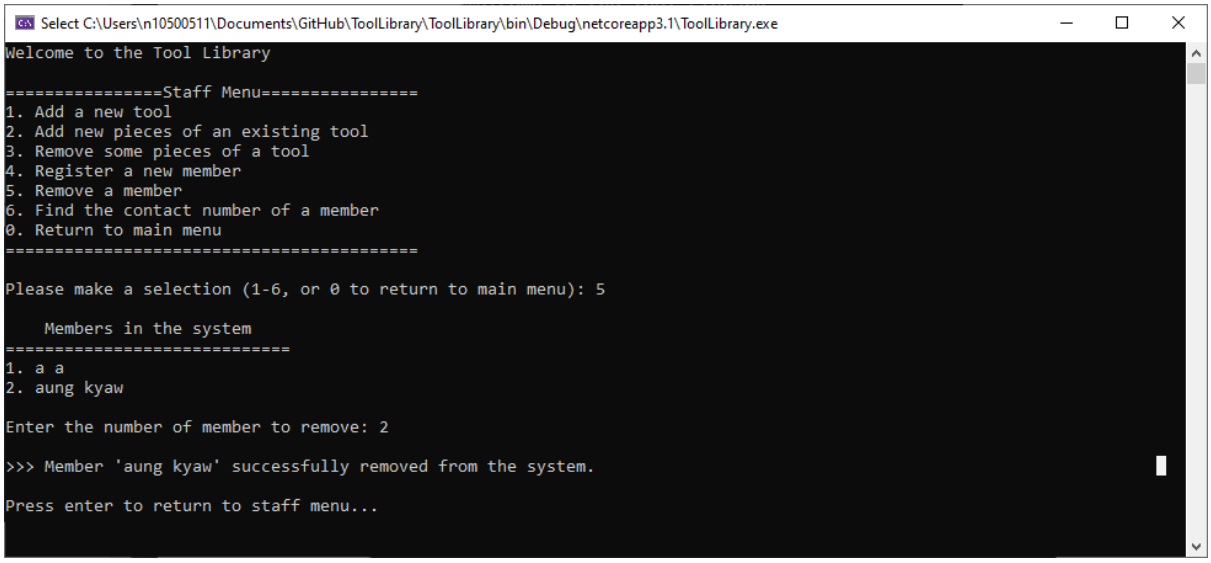

Figure 8 – remove a member

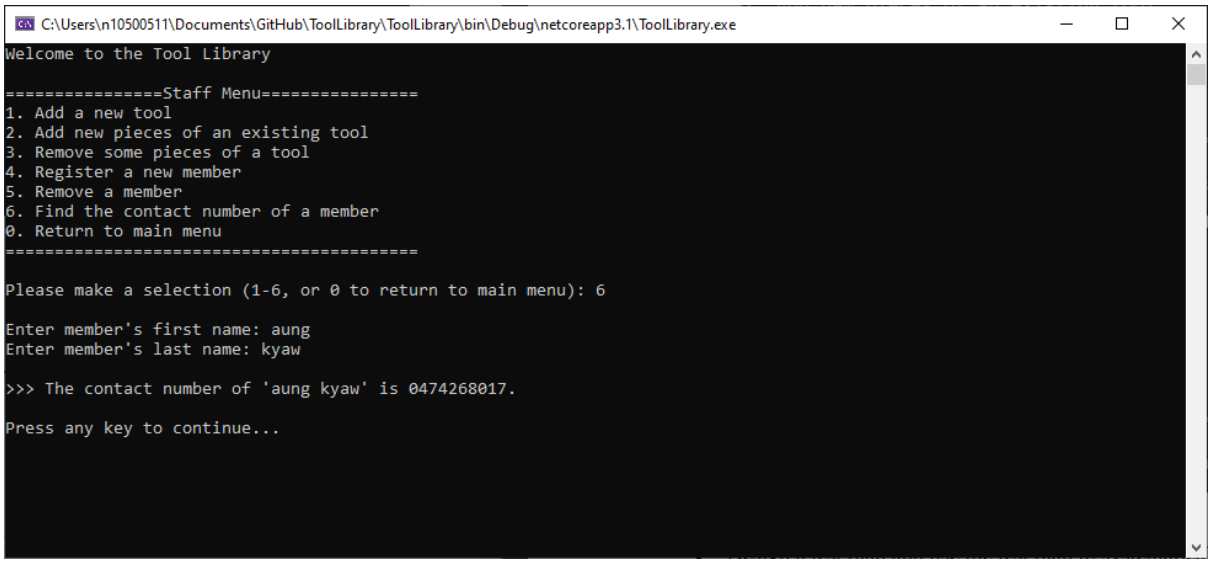

Figure 9 – find the contact number of a member

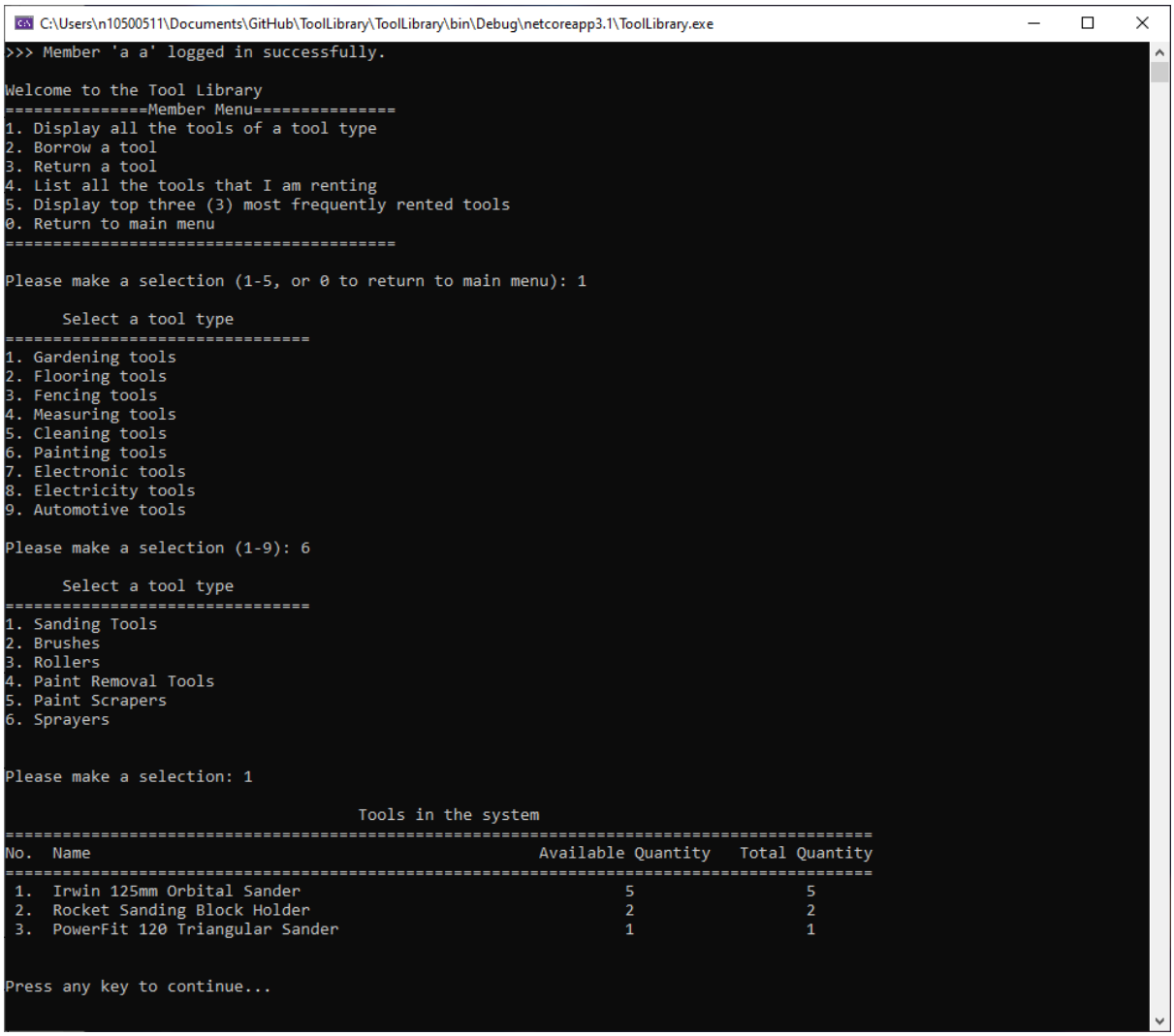

Figure 10 – Display all tool types

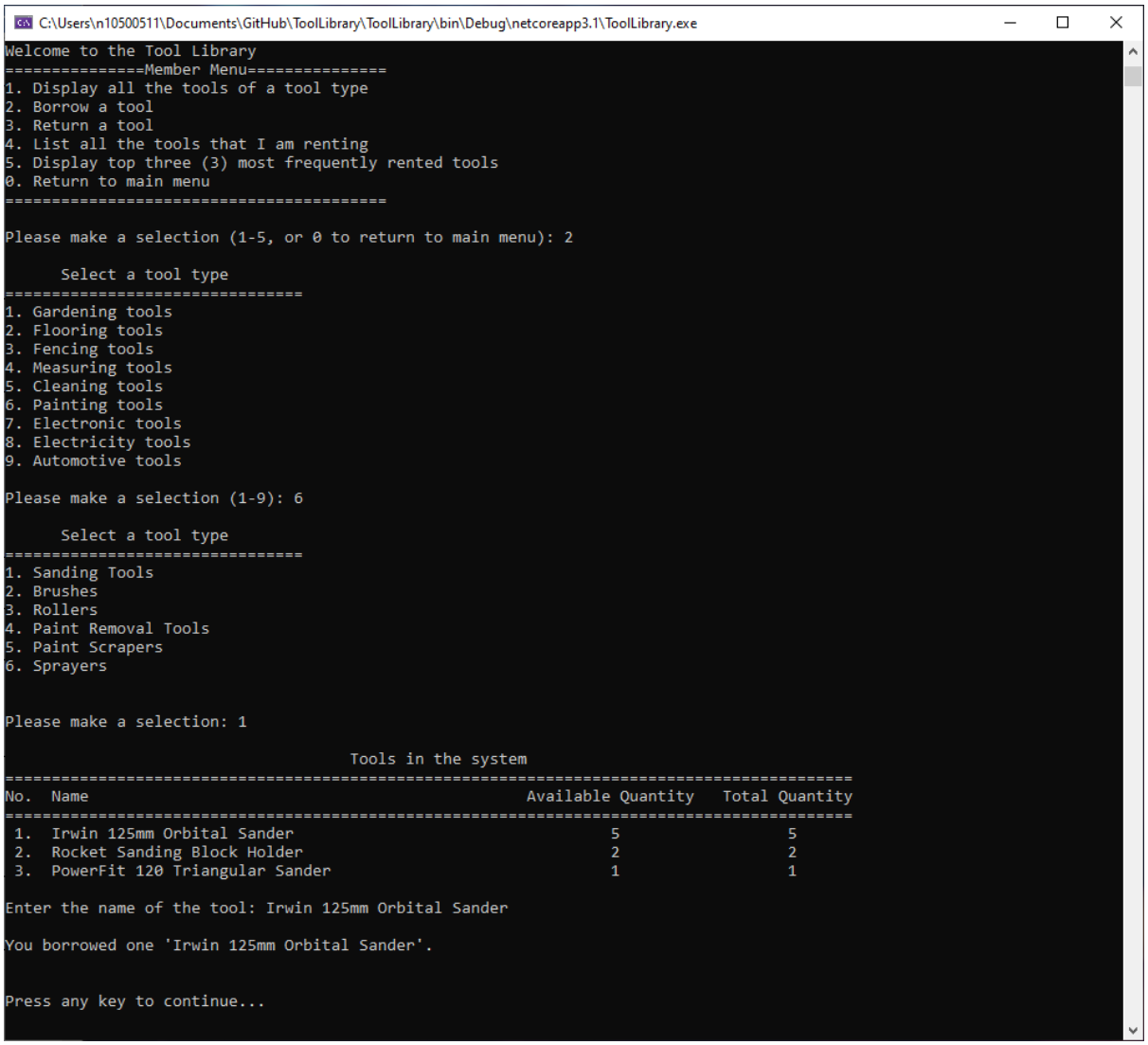

Figure 11 – Borrow a tool

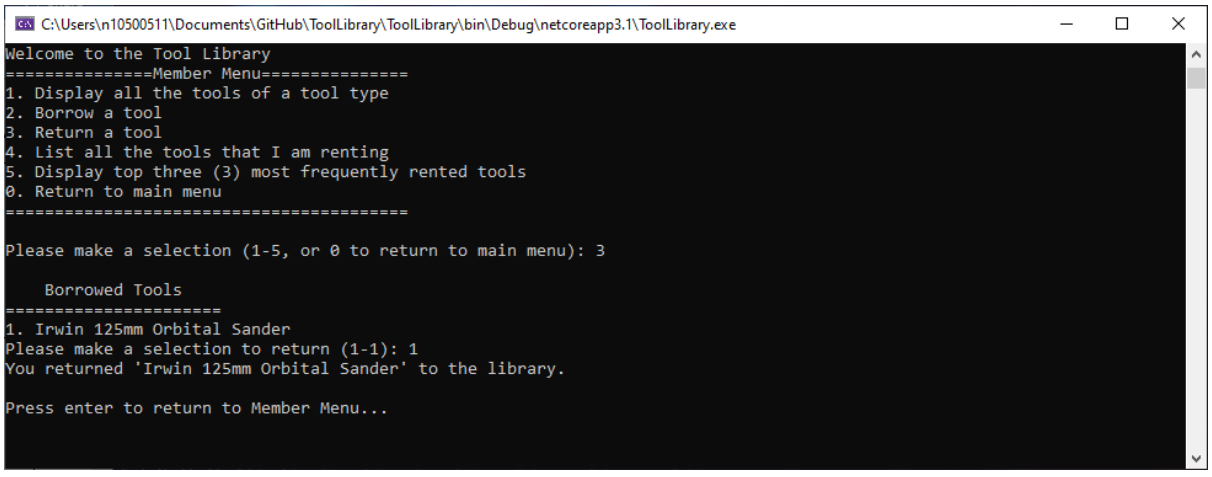

Figure 12 – Return a tool

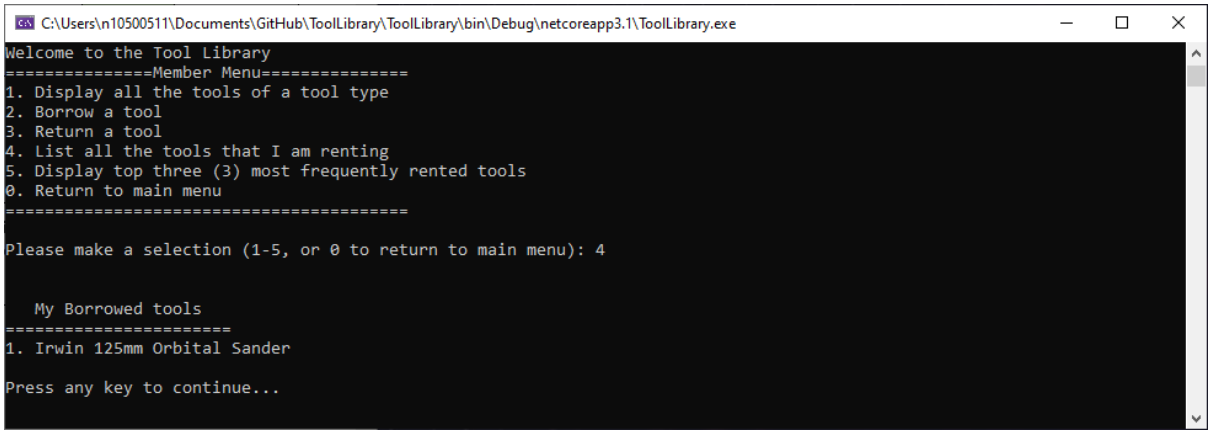

Figure 13 -List all tools being rented

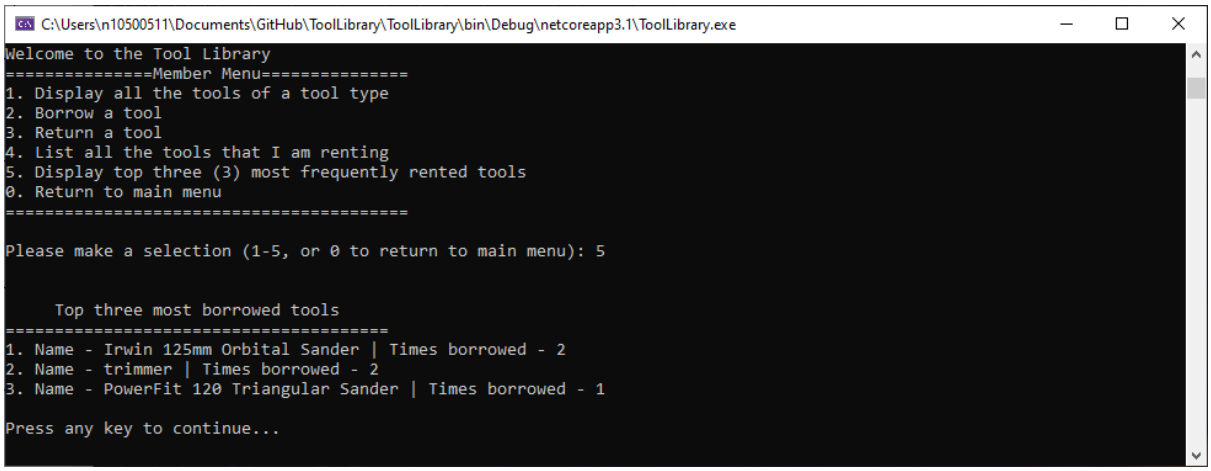

Figure 14 – Display top three tools

6 References

- tutorialpoints. 2021. “Design and Analysis Selection Sort.” Accessed May 31, 2021. https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_selection_sort.htm
- GeeksforGeeks. 2019. “Selection Sort.” Accessed May 31, 2021. https://www.geeksforgeeks.org/selection-sort/
- Wikipedia. 2021. “Selection Sort.” Accessed May 31, 2021. https://en.wikipedia.org/wiki/Selection_sort
- Maolin T. 2021. “CAB301 algorithms and complexity: Week 1 lecture notes.” Accessed May 31, 2021. https://blackboard.qut.edu.au/bbcswebdav/pid-9064363-dt-content-rid-37579885_1/courses/CAB301_21se1/CAB301-Lecture1-Part%202.pdf