

Differential Equations – Major/Grand Assignment

|  |  |
| --- | --- |
| **Group Members** | **Roll no** |
| -Muhammad Abdur Rafey | 21i-0705 |
| -Ayra Alamdar | 21i-2968 |
| -Naima Zafar | 21i-0642 |

**Objectives and Introduction of the problem**

**Harvesting a renewable resource**

The population of yy of a certain species of fish in a given area of the ocean is described by the logistic equation that is:

**dy/dx = r(1-y/K)y**

As we analyze that population is subjected to harvesting at a rate HH(yy,tt) members per unit time, then the harvested population is modeled by the differential equation:

**dy/dx = r(1-y/K)-H(y,t)**

* Now we see that the fish is desirable as the food source than it is intuitively clear that if too many fish are caught.
* Then the fish population may be reduced below a useful level and possibly even driven to extinction.
* The following problems explore some of the questions involved in formulating a rational strategy for managing the fishery.

***Constant Effort Harvesting***

We have been given that the rate at which the fish is caught depends on the population “y”. The rate at which the fish are caught is given by:

**H(y,t) = Ey**

The Schaefer model is given as:

**dy/dt = r(1-y/K)y-Ey**

***Constant Yield Harvesting***

The fish are caught at a constant rate “h” which is independent of the fish population, so the harvesting rate is:  
**H(y,t) = h**

We have also been given that

**dy/dt = r(1-y/K)y-h = f(y)**

**Analytical solution / Step-by-step solution by hand**

Graphical user interface

Description automatically generated

**Matlab Code**

***Problem 1: Constant effort harvest***

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

***Problem 2: Constant yield harvesting***

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# Matlab Solution and results

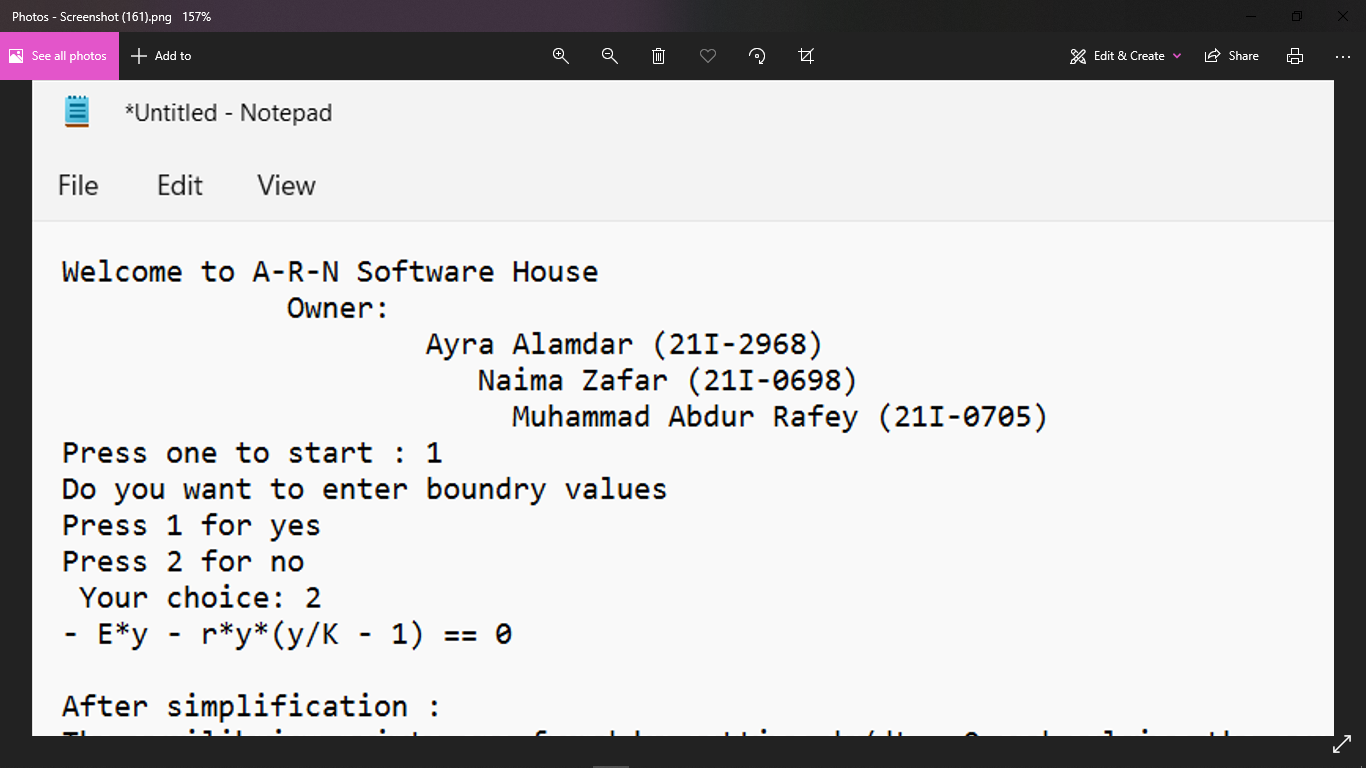
***Problem 1:***

On execution of the code, the user will be welcomed at a screen showing contributors and the software house name and will have to follow the following steps to proceed:

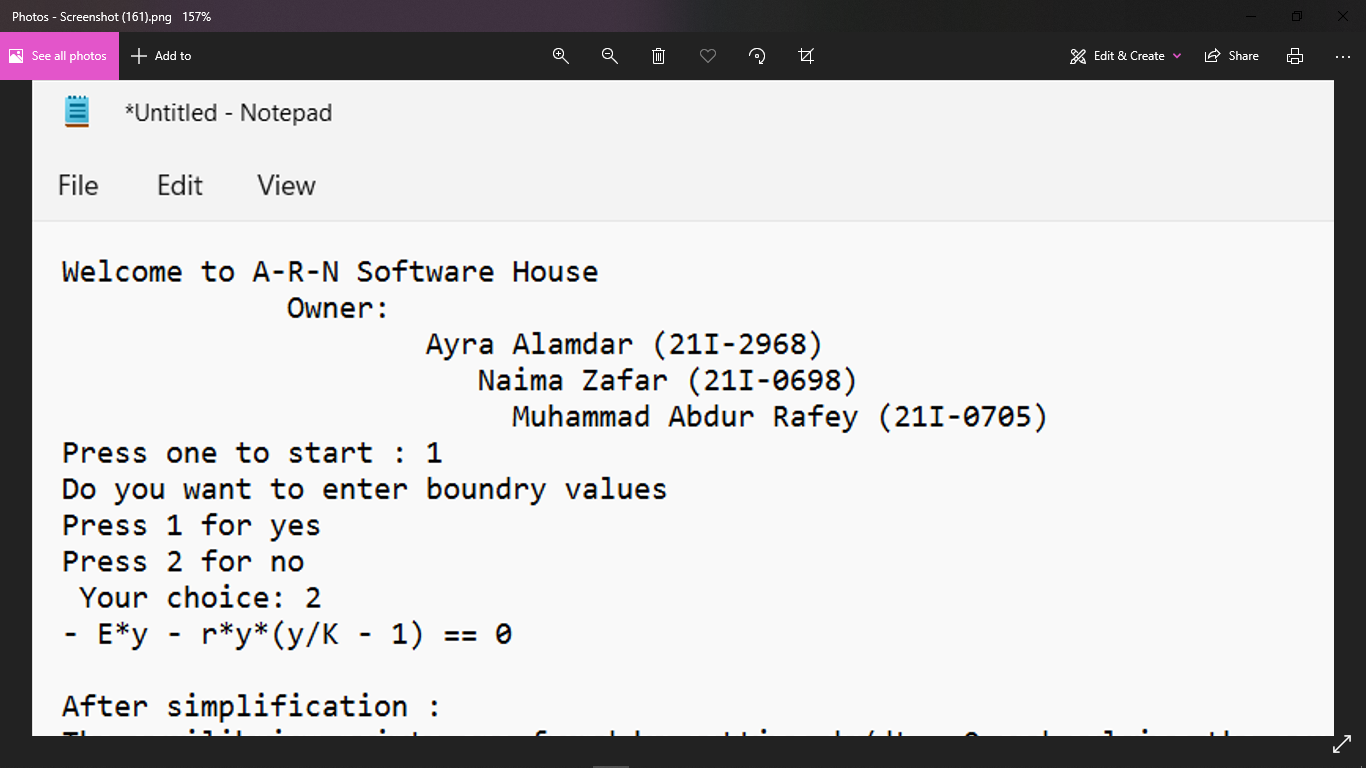
Text

Description automatically generated

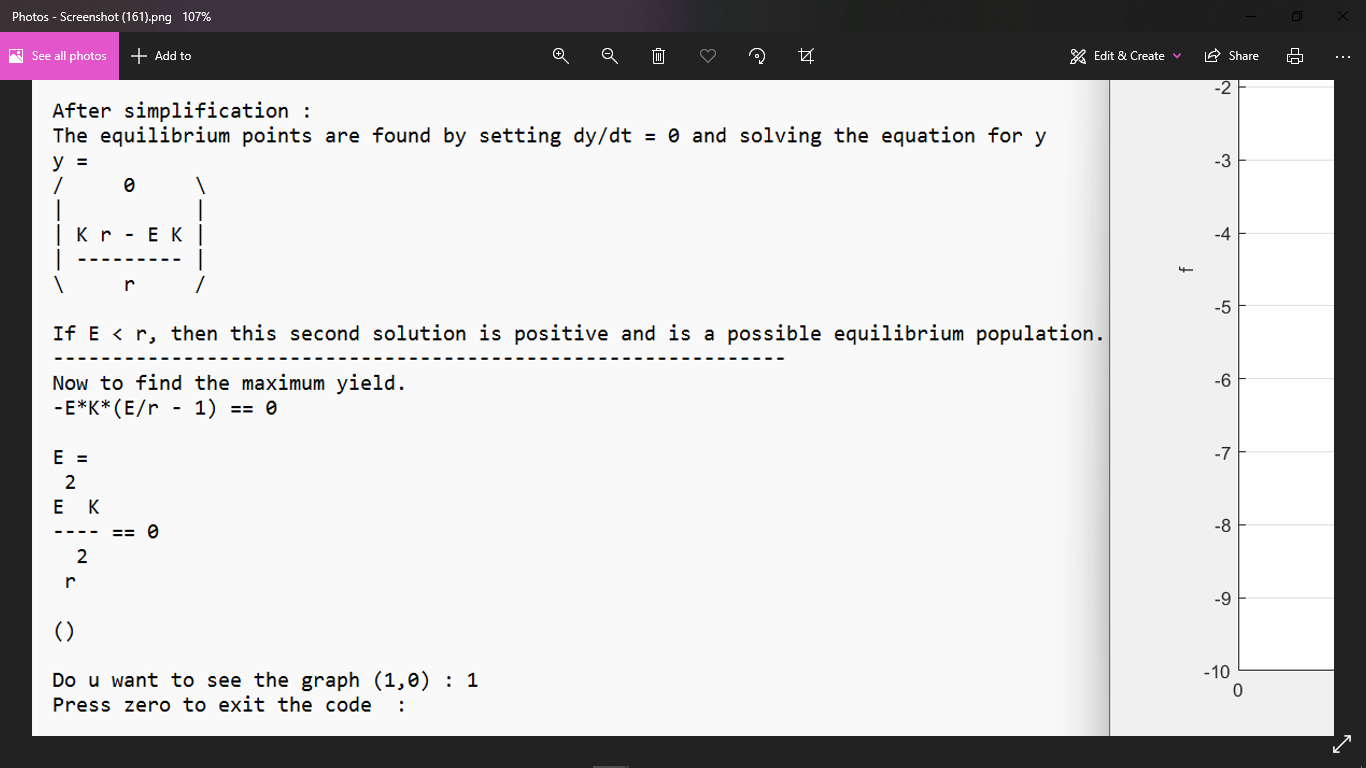
* The user will be asked to press a key, “1” to start and then the user is asked if he/she wants boundary values, “1” for yes and “2” for no:



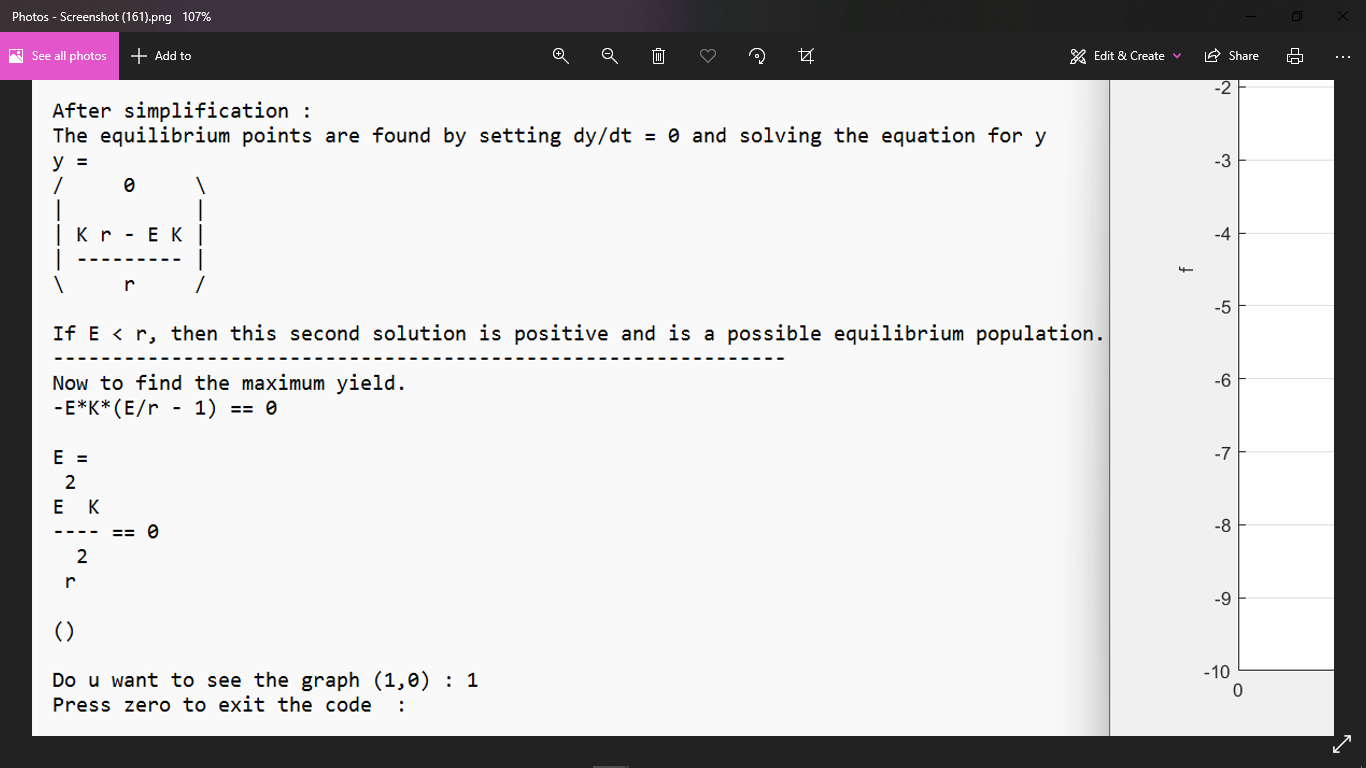
* Once a key is pressed to show the choice of the user, the workspace will display the user’s choice and the user will be redirected to the equation for the solution.



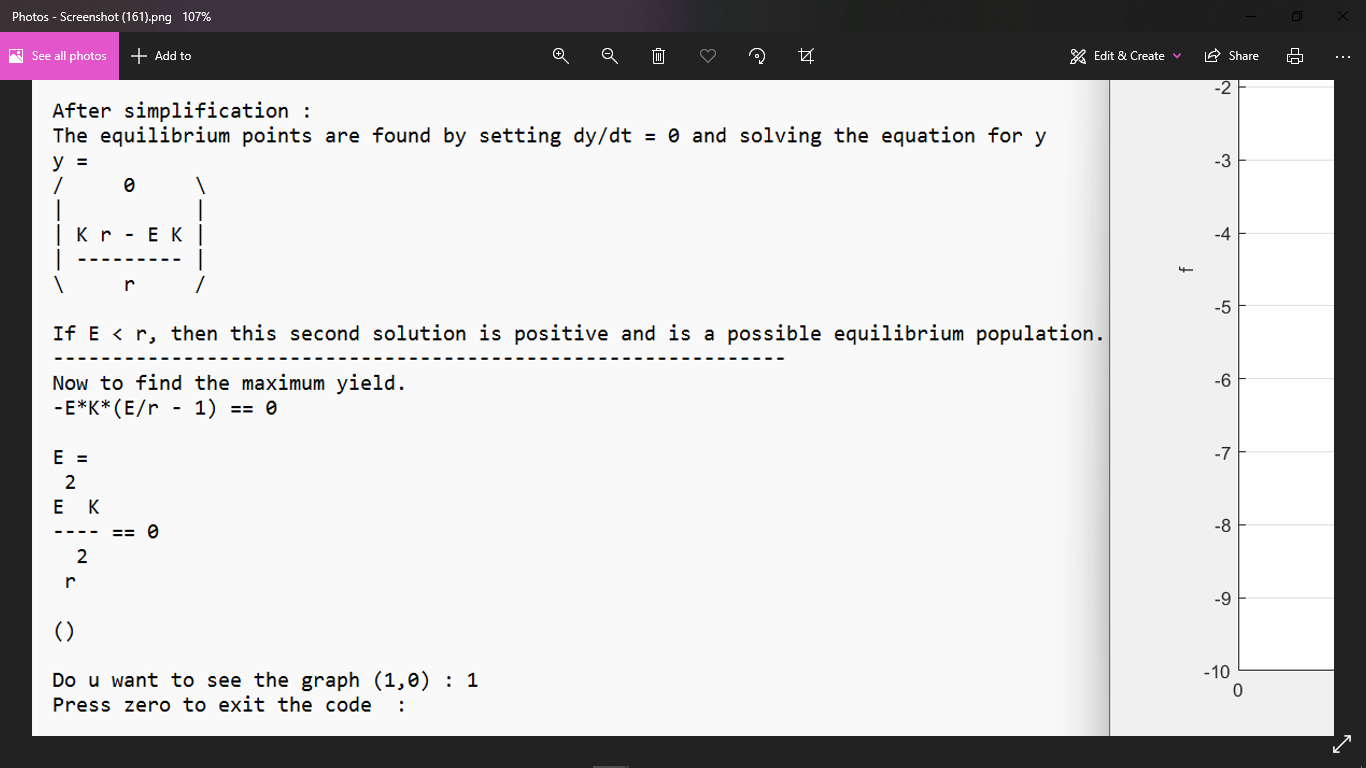
* After the user has chosen their pick, the program proceeds and simplification is done. It is displayed onto the terminal as well and the equilibrium points are found by setting dy/dx = 0. Simultaneously the equation for “y” is also solved :



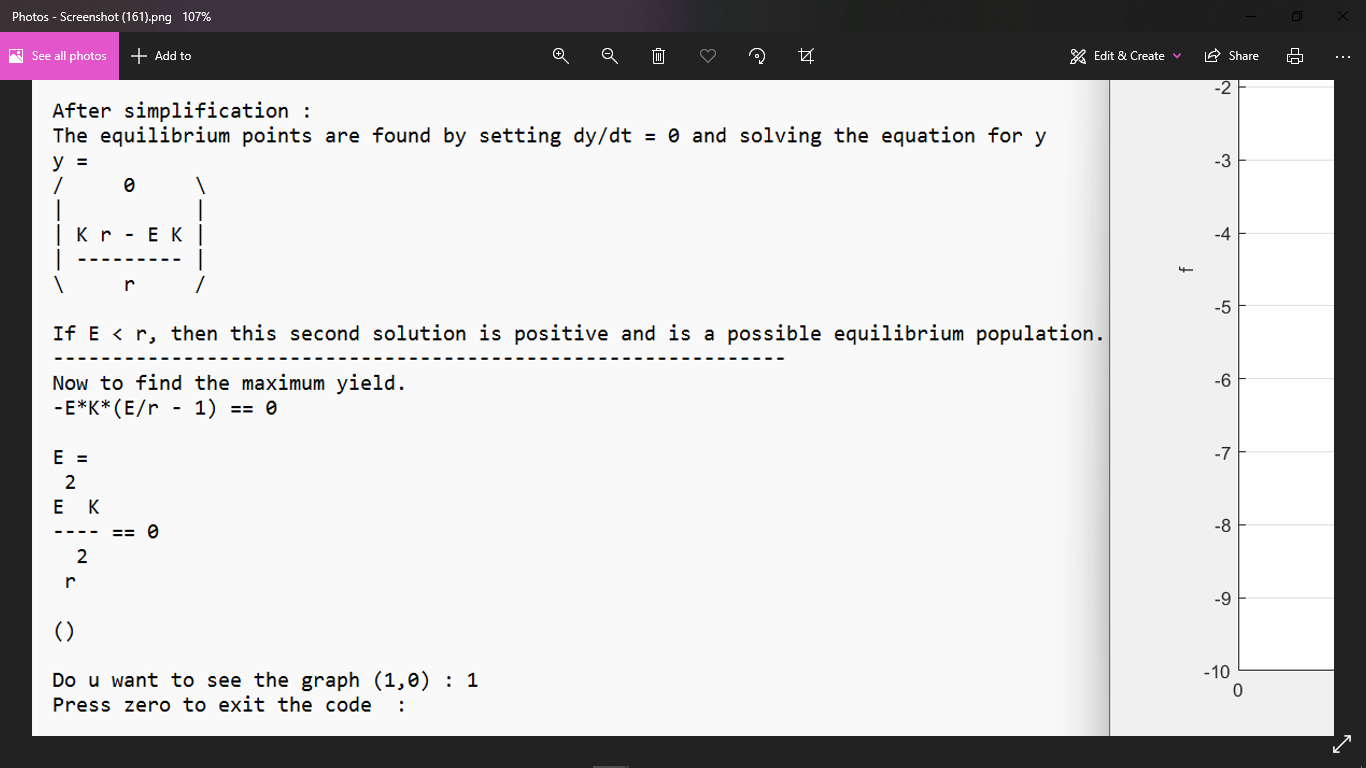
* The statement indicating whether the solution is positive and is a possible equilibrium population is displayed on the terminal to show the user the output of the value they inputted.



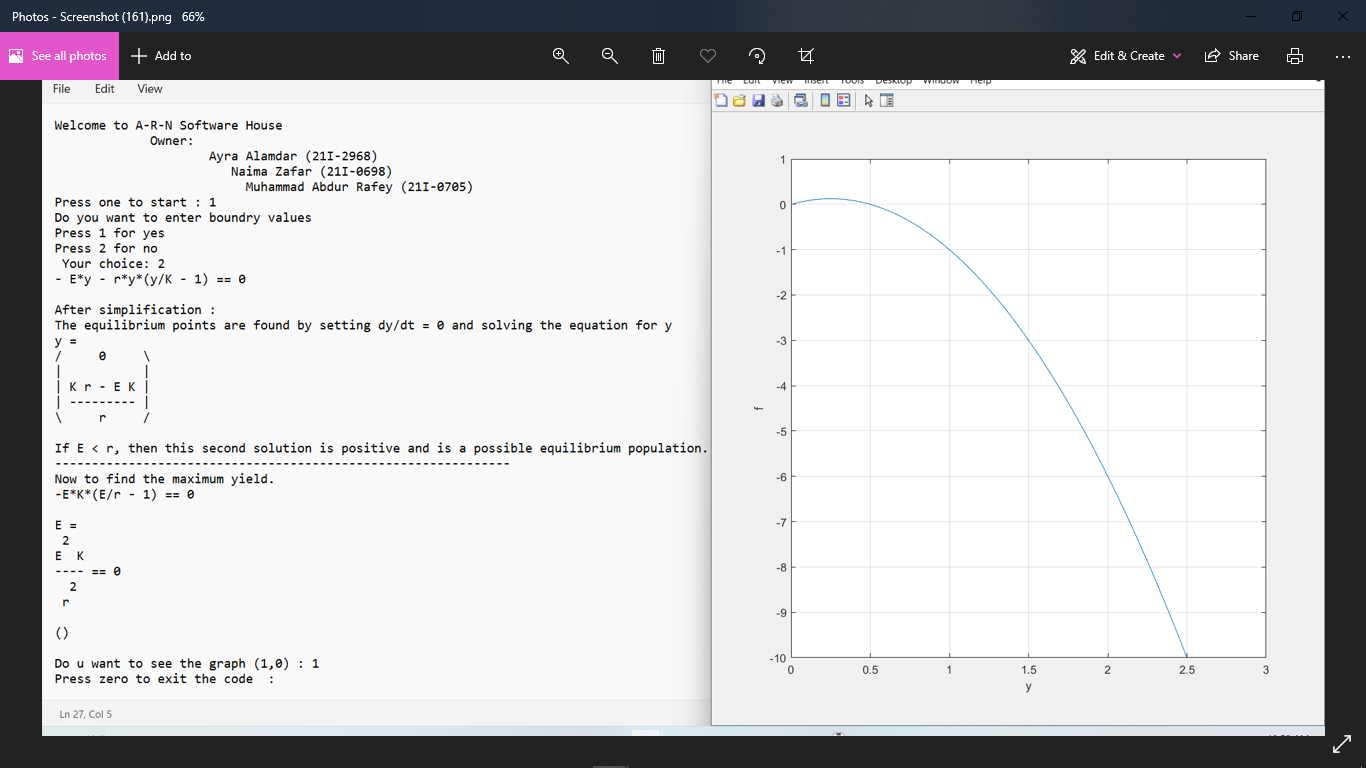
* The formula to determine the maximum yield, after figuring out whether the solution is positive and if it is a possible equilibrium population, is displayed onto the terminal to show what is happening in the code and to help make the understanding of the code easier for the user.



* Now the user is asked if he/she wants to see the graph for the maximum yield, the user is asked to press “1” for yes and “0” for no. After the user’s input on whether he/she wants to see the graph, they are asked to press “0” to exit the program.



* If the user chooses to see the graph and presses “1”, then the graph that will be displayed is as follows:



***Problem 2:***

When the code is executed, a screen will be shown to the user showing the contributors and the software house’s name. The user will be given a set of steps to follow as shown:

Text

Description automatically generated

* The user will be asked to press a key, “1” to start and then the user is asked if he/she wants boundary values, “1” for yes and “2” for no, the user’s choice is then displayed as well:

Graphical user interface, application

Description automatically generated

* After the user picks his/her choice, then the terminal displays the equilibrium points which are found by dy/dt = 0 and resultant of the equation of y(dy/dx) which is as follows:

**Graphical user interface, application

Description automatically generated**

* The formula for calculating the quadratic equation to determine the value of “y” is displayed on the terminal and the value of “y” is found.

**Graphical user interface, application

Description automatically generated**

* The calculations are done and the equilibrium populations are shown on the terminal.

Graphical user interface, text

Description automatically generated

* The user is asked if he/she wants to see the graph, and then an instruction for exiting the code is provided. A complimentary statement is displayed to portray the end of the program.

**Graphical user interface, text

Description automatically generated**

* The graphs are as follows:

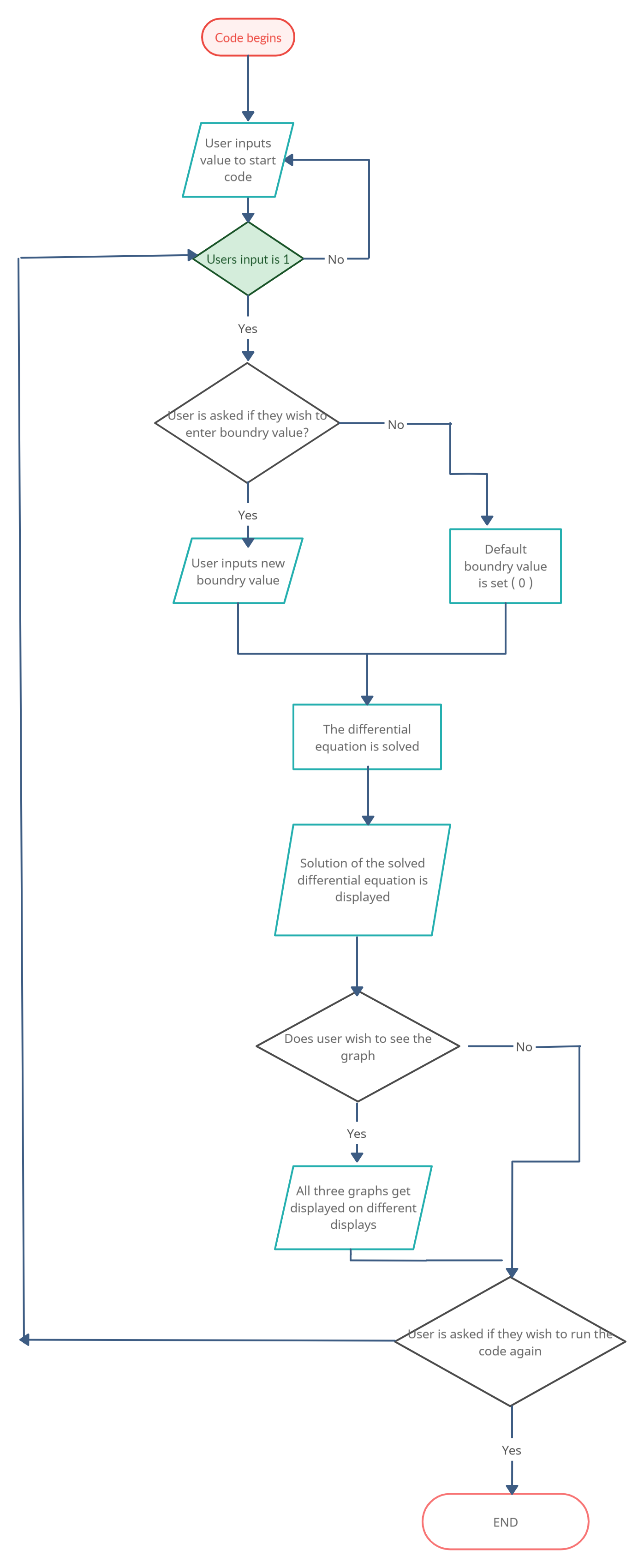
A screenshot of a computer

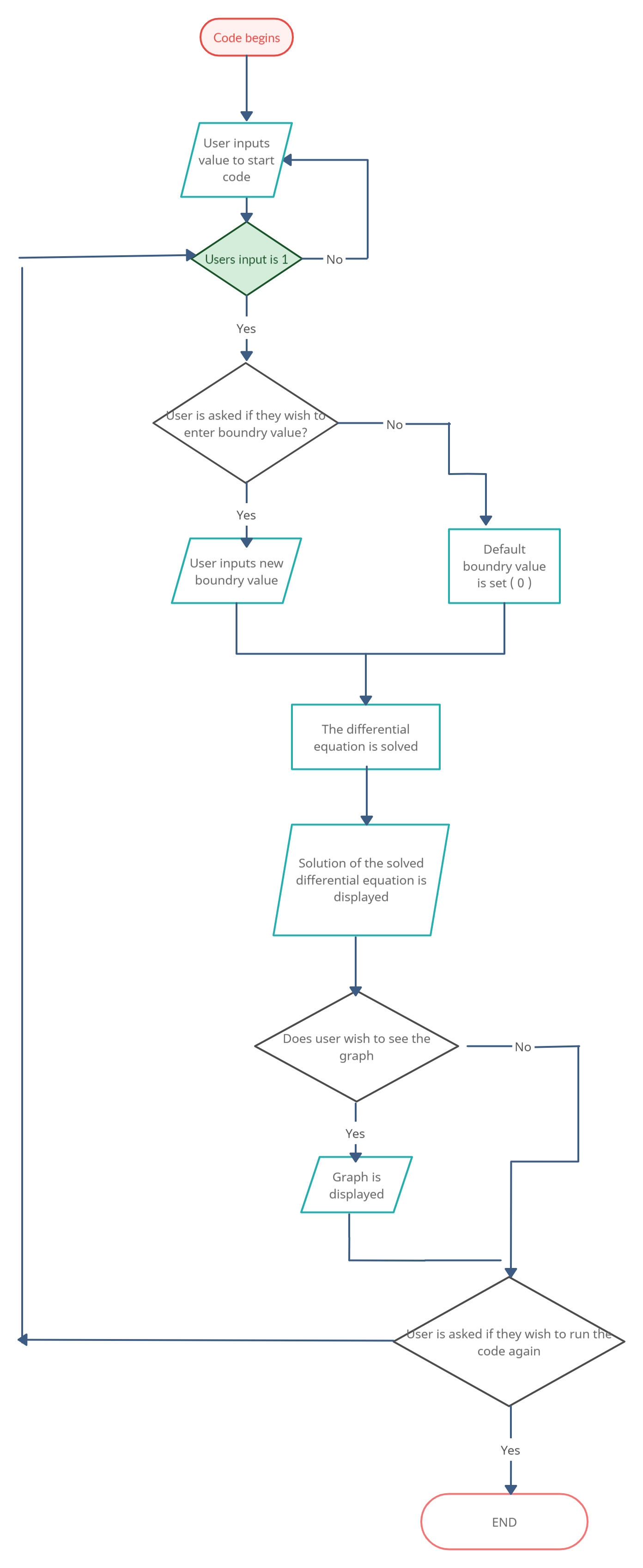
Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

# Flowcharts





**Conclusions**

At first the project looks pretty straight forward, We have been provided with a question that involves using differential equations and trying to figure out a solution for the question using Matlab, a software which we are currently using. When the code starts to run the user is asked to input in the values and conditions for how they would like the question to be run and what kind of problem it is. On running, the program asks the user to input the specific conditions for the differential equation otherwise it runs on default conditions by itself. The software then decodes the differential equation and asks the user if the user wants to see the graph of that very differential equation. At the end of the code, just before the program ends, the user is asked if he/she wants to run the program again or want to end it, if the user enters “yes” then the program runs again and takes values and asks the user to input in the initial conditions again otherwise the program will terminate and will eventually stop running.

Even though the project itself wasn’t that challenging, using the software Matlab proved to be quite a challenge, there were many things we were unaware of and many things we had not studied before such as:  
 1. We didn’t know how to input formulas in Matlab.

2. We didn’t know that input could be taken in Matlab.

3. We were not sure how to plot differential equation problems in Matlab.

We shared our difficulties together and tried and tested various ways to try to fix our code and overcome the difficulties we faced. We watched many videos and saw a couple manuals to fully understand and see how Matlab worked.

After all the hardships, we finally overcame all the difficulties and managed to finish this project.

# Contributions

**Muhammad Abdur Rafey:**

* Creation of Matlab Code.
* Creation of Flowchart.
* Creation of 3D Figures.
* Contributed in Analysis of the question.
* Worked in report.

# Ayra Alamdar:

* Analysis of the question.
* Compiled and created the report.
* Concluded the report.
* Creation of question summary.
* Named the company.

# Naima Zafar:

* First to Come up with a proper by hand solution method.
* Did research on the objectives and introduction of the questions.
* Contributed in research on how to create 3D figures in Matlab.
* Worked in report.
* Worked on understanding and analyzing the question.