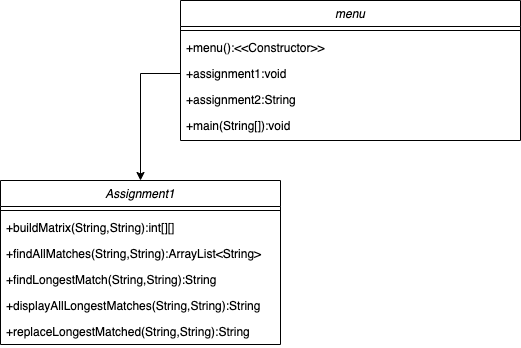
**CO3325 Data compression Report**

**Introduction:** In this assignment we will design and code 2 task, assignment1 and assignment from the requirement. This report includes design & code and run instruction. The report divided in 2 section for one is assignment 1 another assignment 2.

**Assignment 1:**

In this section we will show how we design algorithm and run our program.

****

If we look in the UML class diagram, we can see their have 5 methods/function in Assignment1 class and 4 method in the main/menu class. Menu is designed to make it easy to run the program for everyone. Lets drive deep into assignment 1 how we design and build the task.

In the assignment 1 most important methods are buildMatrix() and findAllMatches(), buildMatrix will take and 2 parameter and build an matrix of all matches, the create matrix is look like this. We can see we have found longest match in the matrix now we need another method to find all matches.

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1,

0, 1, 2, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 3,

0, 0, 0, 0, 3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 1, 0, 0, 0,

0, 1, 1, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 1, 1,

0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 9, 0, 0, 0,

0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0,

0, 1, 1, 1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 1, 1,

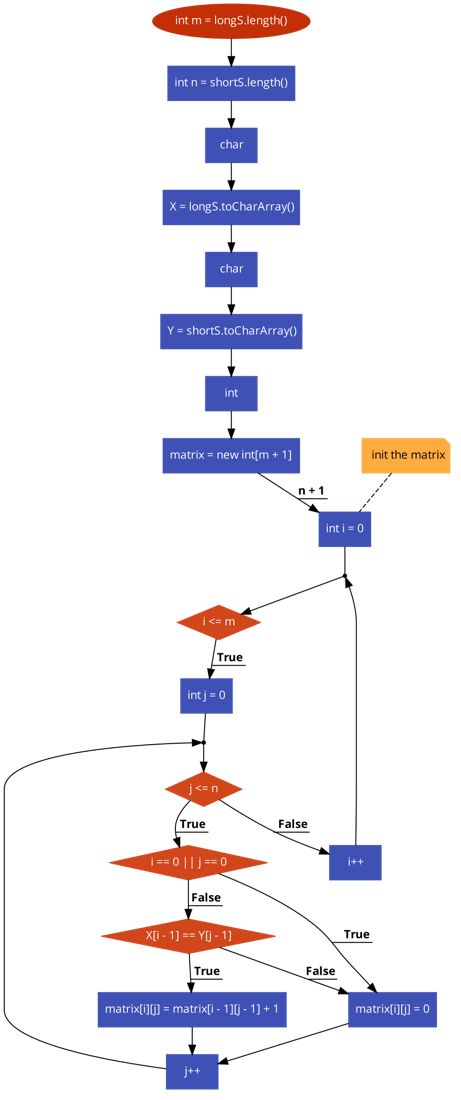
0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0,

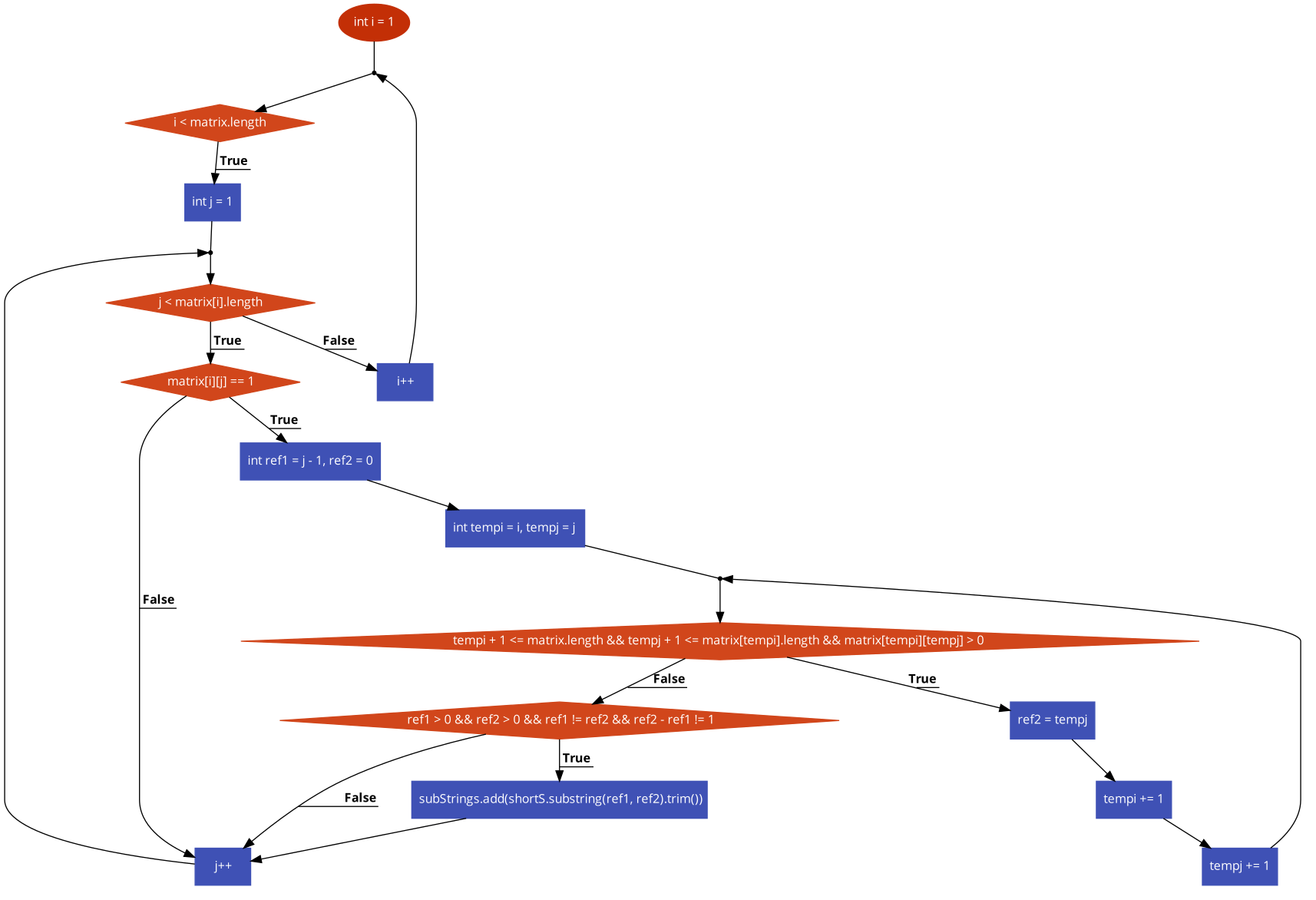
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 8, 0, 0, 0,



Here is the main flow for create the matrix. Now we need another method for find all match we did that in findAllMatches() method.



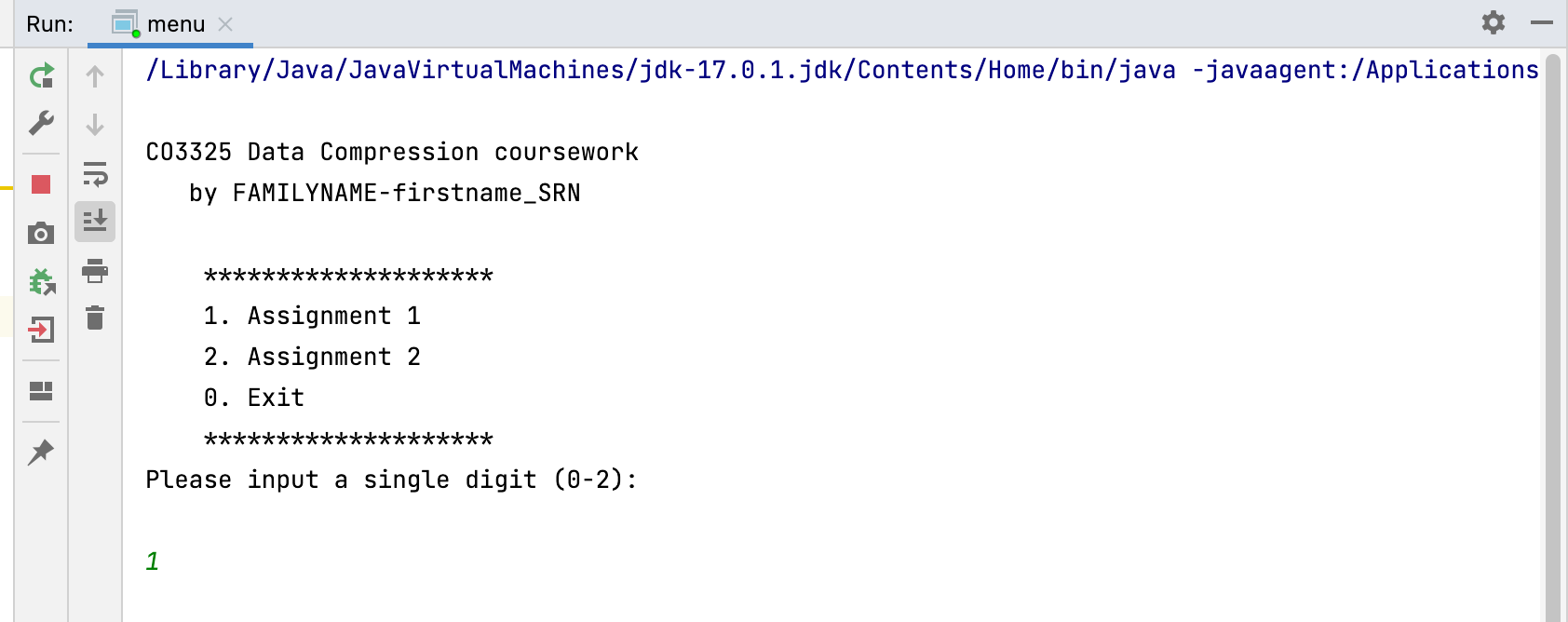
Here is the flow of the findAllMatches() method we have find all matches here and add in an a set so not duplicate match added to list.

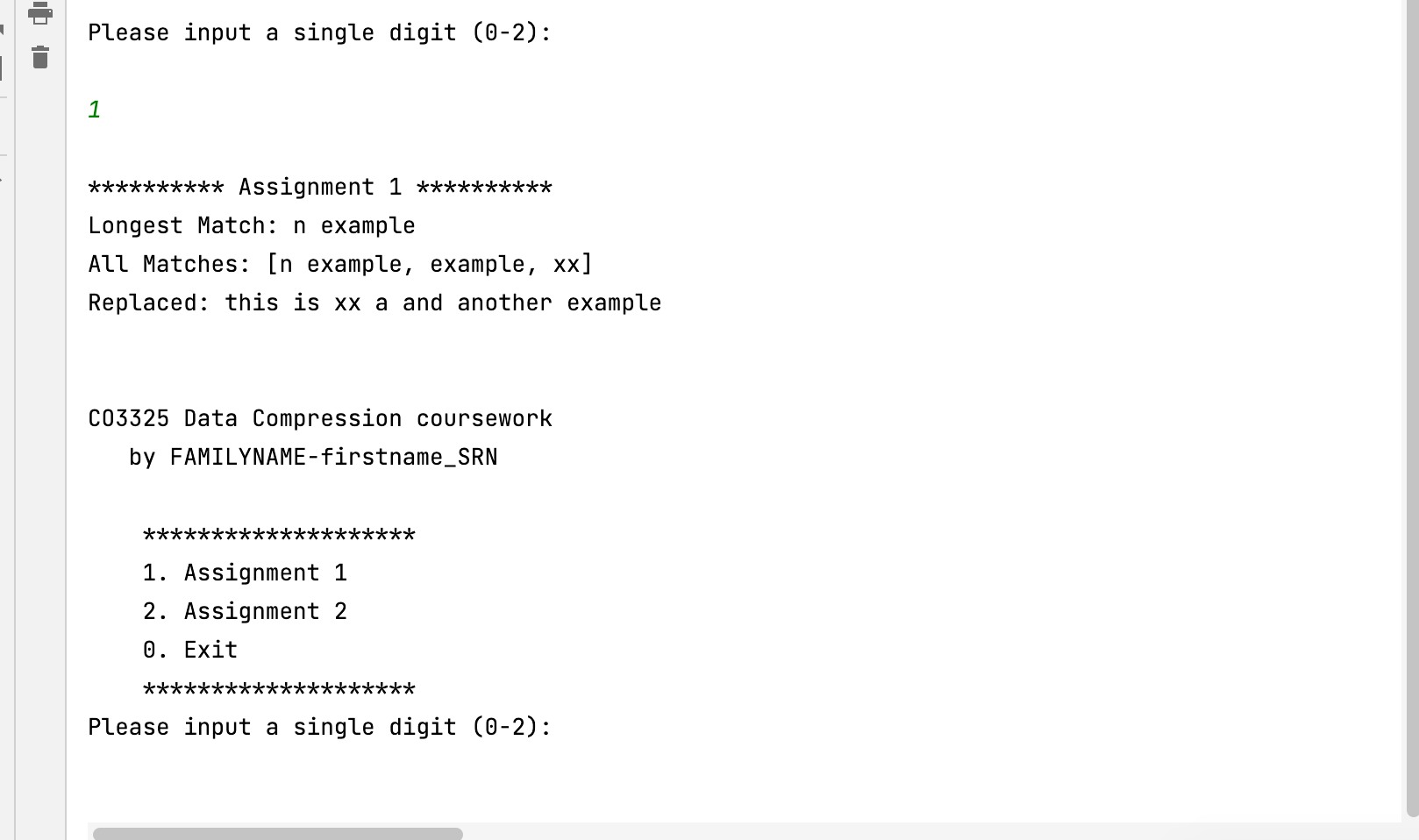
All other method just use this 2 method to produce all reset.

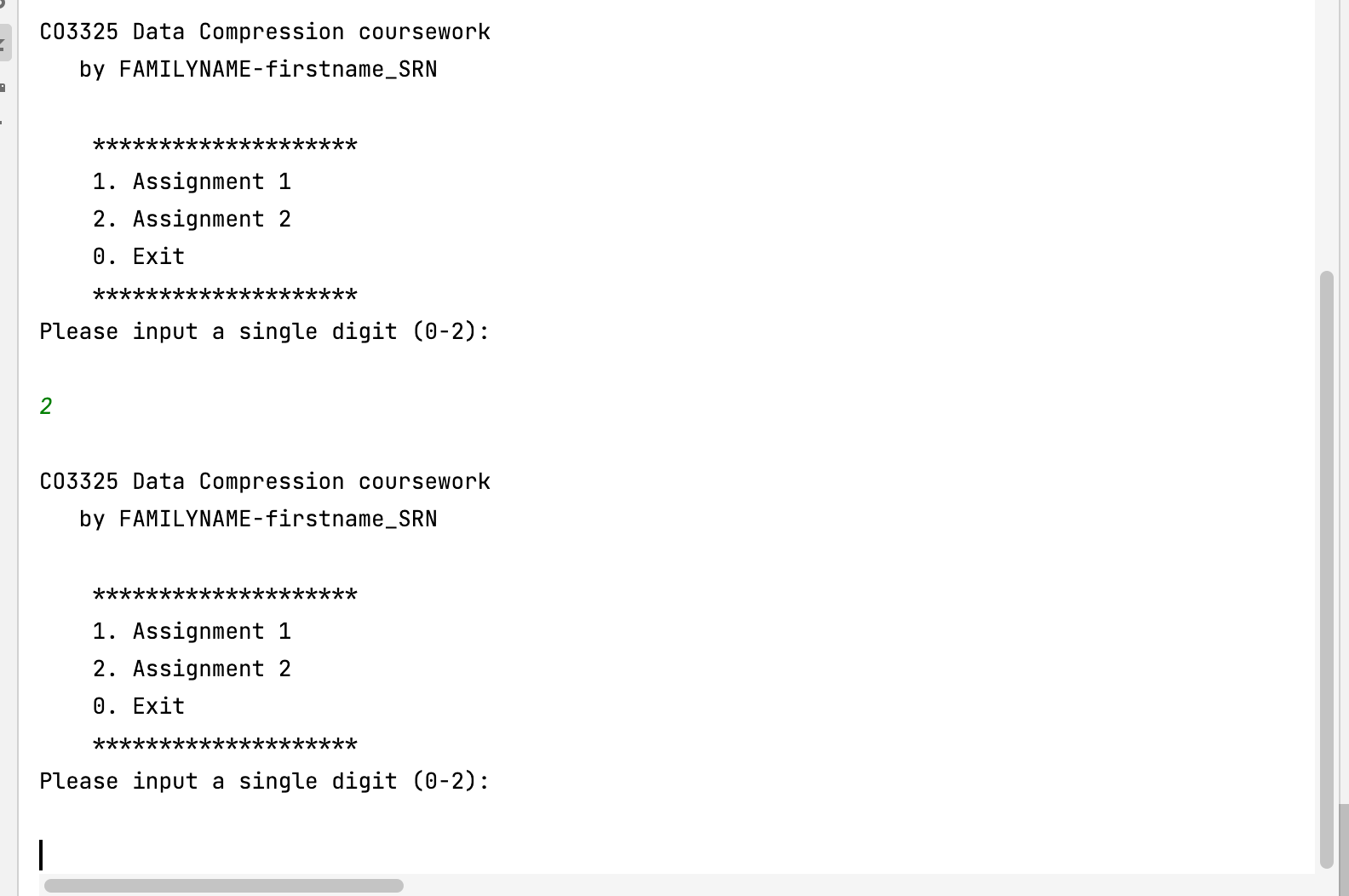
**Timing**:

|  |  |
| --- | --- |
| Total number of hours spent | 12 Hours |
| Hours spent for algorithm design | 3.5 Hours |
| Hours spent for programming | 3.5 Hours |
| Hours spent for writing report | 3 Hours |
| Hours spent for testing | 2 Hours |
| Note for the examiners (if any): |  |

**Run:**

****

****

****