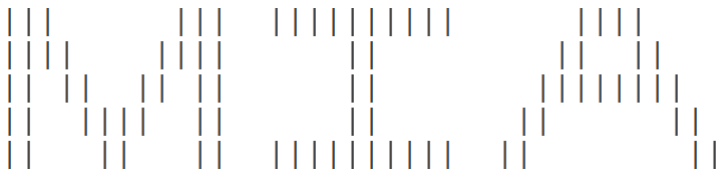

Multiple Integrated Applications (MIA)
Manual & Programming Documentation

Developer: Antonius Torode
Michigan State University
Department of Physics & Astronomy

Version – 0.025
Latest update: February 28, 2018

The Original Maintainer of this work is: Antonius Torode.

The Current Maintainer of this work is: Antonius Torode.

This document is designed for the sole purpose of providing documentation for Multiple Integrated Applications (MIA), a program written for and created for personal use. Throughout this document, "MIA" will be used as a reference to "Multiple Integrated Applications." This acronym is solely designed for use in conjunction with the program and was originally created by the creator of this document. MIA is designed to be used for multiple purposes based on the continual addition of functionality. It can be adapted to work in other situations and for alternate purposes.

This document is continuously under development.
Most Current Revision Date: February 28, 2018

Torode, A.
MIA Manual.
Michigan State University –
Department of Physics & Astronomy.
2017, Student.
Includes Source Code and References

Contents

This page intentionally left blank.
(Yes, this is a contradiction)

Chapter 1

Multiple Integrated Applications (MIA)

MIA is designed to be a collection of scripts, tools, programs, and commands that have been created in the past and may be useful in the future. It's original idea was a place for the original author to combine all of his previous applications and codes into one location that can be compiled cross platform. MIA is written in C++ but will contain codes that were originally designed in C#, Java, Python, and others. MIA is created for the authors personal use but may be used by others if a need or desire arises under the terms of Antonius' General Purpose License (AGPL).

The MIA acronym was created by the original author for the sole purpose of this application. The design of MIA is a terminal prompt that accepts commands. There are no plans to convert MIA into a GUI application as there is currently no need; however, some elements may be programmed in that produce a GUI window for certain uses such as graphs. The MIA manual is designed to be an explanation of what MIA contains as well as a guide of how to utilize the MIA program to it's fullest.

As MIA is continually under development, this document is also. Due to this, it may fall behind and become slightly outdated as I implement and test new features into MIA. I will attempt to keep this document up to date with all of the features MIA contains but I can only do so if time permits.

Chapter 2

Commands and Syntax

2.1 Valid Syntax

MIA is designed to be used similar to a terminal or command prompt. One enters commands and the uses the 'Enter' key to perform the commands. MIA commands are NOT case sensitive. By default, all commands are changed to lower case before executing through the MIA program. If a command is not typed exactly how it is intended (including spaces and newline characters) it may not execute.

2.2 Complete List of Valid Commands (CLVC)

```
help
```

Displays a valid lists of commands and a brief description to go along with each.

```
add
```

Adds two positive integers of any length. This adds two strings together using a similar algorithm one would when adding large numbers by hand. It is possible to get results by entering non-number entries but will serve no significance due to the way MIA internally converts strings to integers by shifting the ASCII values.

```
button spam
```

Spams a specified button (key press). This function asks for a key input as well as asks for a number of times the user would like the button spammed. Not all keys are programmed in and the time between button spamming is currently fixed (this will be updated at a later time). This function currently only works on Windows OS.

```
button spam -t
```

Does the same as the "button spam" command but also simulates the tab key in between each key press.

```
collatz
```

Produces a collatz sequence based on a specified starting integer. This method uses the logn data type which means if a number of the sequence extends the storage of a long, the results will become untrustworthy.

```
crypt -d0s1
```

Encrypts a string using the d0s1 algorithm. This is explained more in chapter 3.

```
crypt -d0s2
```

Encrypts a string using the d0s2 algorithm. This is explained more in chapter 3.

```
decrypt -d0s1
```

De-crypts a string using the d0s1 algorithm. This is explained more in chapter 3.

```
decrypt -d0s2
```

De-crypts a string using the d0s2 algorithm. This is explained more in chapter 3.

```
digitsum
```

Returns the sum of the digits within an integer of any size. Similar to the add command, this converts a string to an array of integers using ASCII shifting and then sums the values together. Due to this, you can also find values for entering non-numerical strings.

```
eyedropper
```

Returns the RGB value of the pixel located at the cursor.

```
factors
```

Returns the number of factors within an integer. The integer must be smaller than C++'s internal storage for the long data type.

```
find mouse
```

This function will locate the position of the usere mouse pointer after 5 seconds and print the coordinates it is located at.

```
lattice
```

Returns total lattice paths to the bottom right corner of an n x m grid. This function is only valid for situations in which the answer will not exceed the internal storage of a long data type.

```
mc dig
```

Simulates key strokes for continuous Minecraft diggigg. This function will press and hold the w key and the left mouse button for forward momentum whilst digging in Minecraft. This function currently only works on Windows OS.

```
mc explore
```

Explores a Minecraft map using /tp. This is for server exploration given someone with /tp power and creative mode. You can enter a range of coordinates and MIA will emulate the keystrokes to /tp over a large area in order to generate the map. This is handy when using mc server plugins such as dynmap which will display explored map areas via a web browser. This function also asks for the user to specify a time between each /tp so that it can be adapted for use on both fast and slow servers/computers. This function currently only works on Windows OS.

```
multiply
```

Multiplies two integers of any length. Similar to add, this multiplies two strings together using a similar algorithm one would when multiplying large numbers by hand. It is possible to get results by entering non-number entries but will serve no significance due to the way MIA internally converts strings to integers by shifting the ASCII values.

```
palindrome
```

Determines if a positive integer is palindrome. The integer must be smaller than C++'s internal storage for the long data type.

```
prime
```

Determines if a positive integer is prime or not. The integer must be smaller than C++'s internal storage for the long data type.

```
prime -help
```

Displays help defailts for prime functions.

```
prime -f
```

Determines all of the prime factors of a positive integer. The integer must be smaller than C++'s internal storage for the long data type.

```
prime -n
```

Calculates the n'th prime number up to a maximum number of 2147483647.

```
prime -n -p
```

Creates a file of all prime numbers up to a maximum number of 2147483647.

```
prime -n -c
```

Clears the file created by 'prime -n -p'.

`quadratic form`

Calculates the solution to an equation of the form $ax^2+bx+c=0$. This function accounts for imaginary answers.

`subtract`

Finds the difference between two integers of any length. Similar to add, this subtracts two strings together using a similar algorithm one would when subtracting large numbers by hand. It is possible to get results by entering non-number entries but will serve no significance due to the way MIA internally converts strings to integers by shifting the ASCII values.

`randomFromFile`

Prints a number of random lines from a text file. This will read in each line from a text file and print a user specified number of the lines by choosing them randomly.

`triangle`

Determines if a number is a triangle number or not. The integer must be smaller than C++'s internal storage for the long data type.

`wow dup letter`

Duplicates a letter in WoW a specified number of times. This will simulate entering a recipient, subject, and then pasting a message into the body followed by hitting the send button through the in game mailbox on World of Warcraft. The user specifies needed information and number of letters to send. This function is useful for RP scenarios.

`exit`

Quits MIA.

Chapter 3

D0sag3 Command (D3C) Integration

3.1 D3C Introduction and Overview

3.2 d0s1 Encryption

3.3 d0s2 Encryption

References

[1]