**MatLang Translator**

**Usage**

//TODO To compile use -lm flag.

**Description**

* This module converts an expression or script in MatLang language to C language and creates an executable out.c.
* Module takes a file with .mat extension and written in MatLang language as input. Examines the file by performing syntax checks and type checks. If there is a syntax error in MatLang language, it will raise an error, otherwise it will output the C language equivalent of this program.

**Project Outline**

This module was written in C language and consists of five .c files, one .h file.

main.c: Reads the MatLang language from the input file and writes the corresponding c equivalence to the out file.

eval.c: Evaluates and translates the line and also performs type checking operations.

matlang\_functions.c: Declaration of necessary MatLang operations and functions like adding, multiplying.

assignment\_declaration.c: Declaration of scalar and matrix assignment and declaration functions.

globals.c: Declaration and assignment of global variables used in program.

structs.h: Definitions of scalar and matrix structs used in the project.

**Journey of a MatLang Code**

* First of all, the .mat file and the output out.c file are opened.
* At the beginning of the out.c file, function definitions such as choose, transpose that can be used throughout the program are written.
* The given input is examined line by line and is separated into the tokens.
* By looking at these tokens, we decide what the expressions are.
* After understanding what the expressions are, the translation operations are performed according to the expressions using the functions we have defined before.
* //TODO If there are statements that need to be parsed //TODO
* Statements whose translations have been completed to C language are simultaneously written to the out.c file.

**Used Technologies and Structures**

Since this is a group project, we used git to be able to work on the same files asynchronously and have the opportunity to review and comment on each other's code while staying up-to-date about changelog. As a C libraries; <stdio.h>, <stdlib.h>, <string.h>, <ctype.h> are used. Naturally, pointers and lists were used extensively throughout the program. In addition, the struct structure was used in order to be able to process more easily and to access the properties of the matrices easily.