# Arrays and Functions in C

Due: October 29, 2019 on myCourses at 23:55

Labs E and F will provide some background help for this mini assignment.

## QUESTION 1: reverse.c

Write a C program called: `./reverse WORD1 WORD2`

Where:

- WORD1 is a series of characters without spaces, forming a single word.
- WORD2 is a series of characters without spaces, forming a single word.

The source file is called: `reverse.c`

The executable file is called: `reverse`

The program takes as input from the command line two words and checks to see if WORD2 is the reverse of WORD1. For example: abcd is the reverse of dcba.

Your program must do the following in order in `main()`:

1. Return an error message and halt the program when the user does not provide exactly 2 arguments at the command line. Display the following message: "Wrong number of arguments. Please input: ./reverse WORD1 WORD2." Program terminates with the 1 code.
2. Check to see if WORD2 is the reverse of WORD1.
3. Display "WORD1=%s WORD2=%s – REVERSE" if is is the reverse, or display "WORD1=%s WORD2=%s – NOT REVERSE" if it is not the reverse.
4. Program terminates with the zero code

## QUESTION 2: matrix.c

| | |
|---|---|
| Write a C program called | : `./matrix` |
| The source file is called | : `matrix.c` |
| The executable file is called | : `matrix` |

A matrix in algebra is an array of numbers arranged in rows and columns. In C, it is common to implement Matrices using two dimensional arrays. We will be focusing on manipulating a fixed size 5x5 square matrix (5 rows and 5 columns).

Create a C program called `matrix.c` that implements multiple matrix related functions.

The functions that you must implement are the following:

1. Write a function that will fill a 5x5 matrix by randomly generated numbers. You can use the rand() and/or srand() functions provided by <stdlib.h> library to generate random numbers between 1 and 100. The signature of the function you will create is:

   ```
   void fillMatrix(int matrix[ROWS][COLS]);
   ```

   ROWS and COLS are global constants defined as follow at the beginning of your c file:

   ```
   #define ROWS 5
   #define COLS 5
   ```

2. Write a function that given a 5x5 matrix as argument, will print each row of the matrix to the screen line by line. The signature of the function you will create is:

   ```
   void printMatrix(int matrix[rows][cols]);
   ```
   Use **array notation** to format your output.

3. In algebra, the transpose of a matrix is a matrix whose rows are the columns of the original. Write a function that will take a 5x5 matrix as parameter and then transpose it in-place (which means without the use of an extra or a temporary matrix while transposing). The signature of the function you will create is:

   ```
   void transposeMatrix(int matrix[rows][cols]);
   ```
   Use **pointer notation** in your implementation. Do not use array index.

4. Write a function that given three matrices as parameters (two as input matrices and one as output), will calculate the product of the first 2 matrices and place the result in the third one. The signature of the function you will create is:

   ```
   void multiplyMatrix(int m1[2][cols],
                       int m2[rows][cols],
                       int resultMatrix[rows][cols])
   ```

   Note that the first matrix isn't 5x5. Extra space in resultMatrix should be filled with zero. resultMatrix will hold the result of the multiplication of m1 X m2.

   Use **regular array notation** in your implementation. You can use the array index.

The main() function that will be used to test your functions is provided as follow:

```
int main()
{
    int matrix[rows][cols];

    fillMatrix(matrix);
    printMatrix(matrix);

    transposeMatrix(matrix);
    printMatrix(matrix);

    int matrix2[2][cols]={0,1,2,3,4,5,6,7,8,9};
    int matrixResult[rows][cols];

    multiplyMatrix(matrix2, matrix, matrixResult);
    printMatrix(matrixResult);

    return 0;
}
```

## IMPORTANT

You must use **mimi.cs.mcgill.ca** to create the solutions to this assignment. You cannot use your Mac command-line, Windows command-line, nor a Linux distro installed locally on your laptop. You can `ssh` or `putty` from your laptop to **mimi**, or you can go to the third floor of Trottier and use any of those labs, to complete this assignment.

## WHAT TO HAND IN

Everything must be submitted to My Courses before the due date. Remember that you can hand in your assignment up to two days late but there will be a penalty of 5% each day. After that, your assignment will not be accepted. Please hand in the following:

- The reverse.c file
- The matrix.c file
- A bash compile.sh script that compiles the programs for the TA using gcc.
- **Do not hand in the executable**
- Zip all these files so that myCourses will receive them correctly.
- Make sure to add comments to your C program.
- **Add your name and student ID number as a comment.**

## HOW IT WILL BE GRADED

THE TESTING SCRIPT

The TA will run a testing script when evaluating your assignment. This test script is included in the assignment. You can use this testing script to verify that your program runs correctly. The script file is called `mini4testerReverse.sh`. and `Mini4TesterMatrix.sh`

## POINTS AWARDED

The assignment is worth a total of 20 points.

- 10 points for Question 1
  - 2 points – code looks professional (pretty)
  - 2 point – command line arguments
  - 1 point – errors message for wrong parameters
  - 4 point – validates reverse correctly
  - 1 point – displays correct final output
- 10 points for Question 2
  - 2 points – code looks professional (pretty)
  - 2 points – fillMatrix()
  - 1 points – printMatrix()
  - 2 point – transposeMatrix()
  - 3 point – multiplyMatrix()

## GRADING RULES

The following rules are followed by the TA when grading assignments:

- A program must run in order to get a grade (even if it does not run well). If it does not run (does not compile) it will receive a zero. (Make sure to run your programs from mimi.cs.mcgill.ca).
- The TA will grade using the mimi.cs.mcgill.ca server.
- All questions are graded proportionally. This means that if 40% of the question is correct, you will receive 40% of the grade.